

C2000 Piccolo 评估开发套件

用户手册

目录

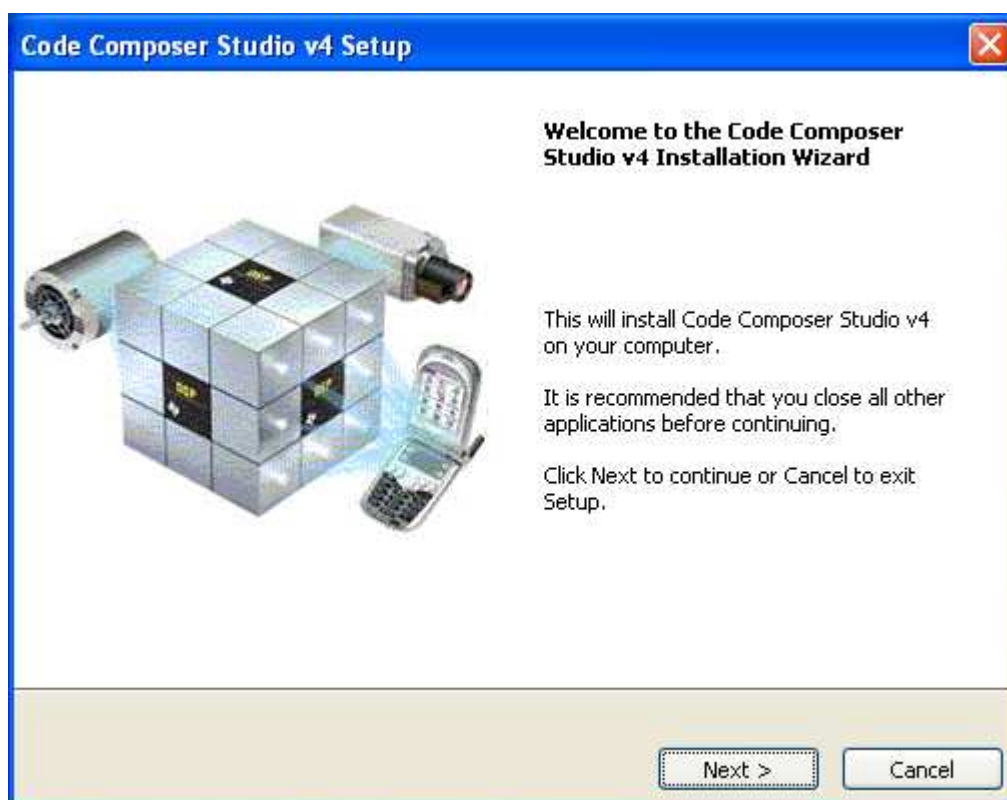
1、CCSv4 简介	2
2、安装 CCSv4	2
3、激活 CCSv4	6
4、在 CCSv4 中运行第一个 C2000 项目	7
4.1 选择工作区	7
4.2 运行 CCSv4 cpu_timer project	7
5、将 C2000 项目从 CCSv3.3 迁移到 CCSv4	14
5.1 将一个项目从 CCSv3.3 迁移到 CCSv4	14
5.2 将多个项目同时从 CCSv3.3 迁移到 CCSv4	18
6、C28x 闪存编程	22
7、C2000 GEL 文件迁移	23
8、Control Suite 软件说明	23
8.1 Control Suite 简介	23
8.2 Control Suite 特性	23
8.3 Control Suite 当前版本中的组件	24
8.4 Control Suite 支持的芯片	24
8.5 Control SUITE 下载及安装	25
8.6 Control SUITE 的桌面	29
8.7 芯片、开发套件、库	30
9、开发套件电路图	30
9.1 TMS320F28027DA-38pin 电路图	31
9.2 TMS320F28027PT-48pin 电路图	32
9.3 TMS320F28035PAG-64pin 电路图	33
9.4 TMS320F28035PN-80pin 电路图	34

1、CCSv4 简介

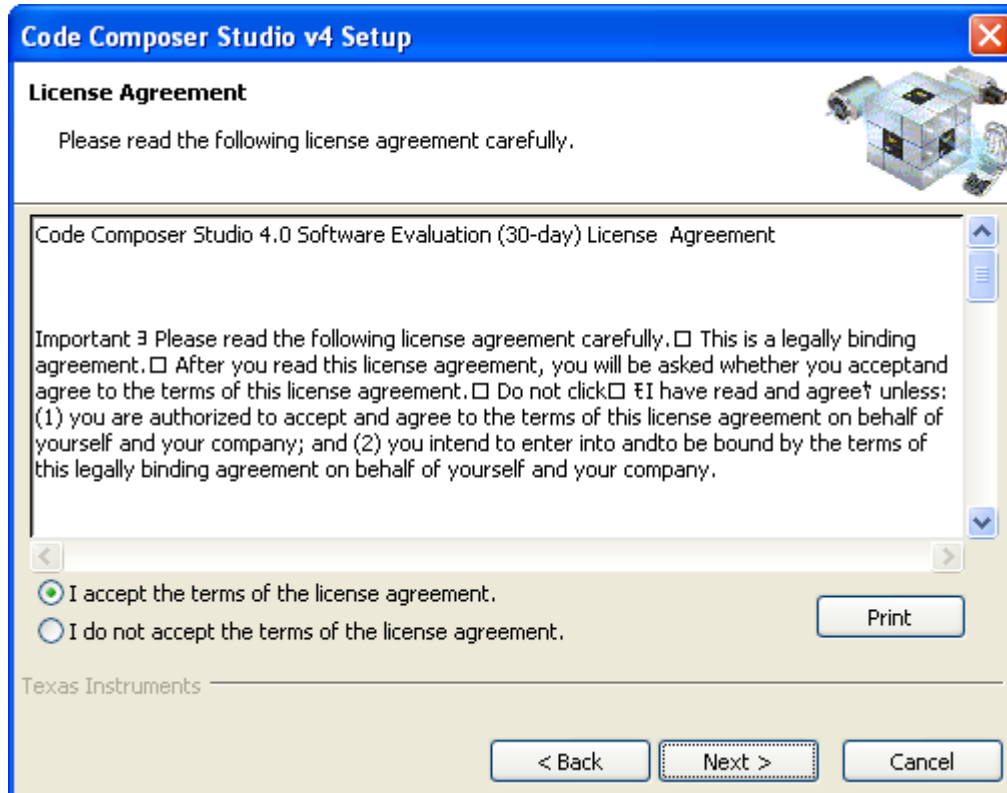
因为 Code Composer Studio v4.0 (CCSv4) 现在是基于 Eclipse 3.2 IDE 开源软件框架的新版本，所以与所有旧版 C2000 Collateral 所基于的 Code Composer Studio v3.x (CCSv3.x) 相比，两者之间存在多处差异。本手册详细介绍了如何安装 CCS，注册 CCS 许可证，以及如何将 C2000 Collateral 从 CCSv3.x 迁移到 CCSv4，并帮助用户开始建立第一个 CCSv4.0 程序。

2、安装 CCSv4

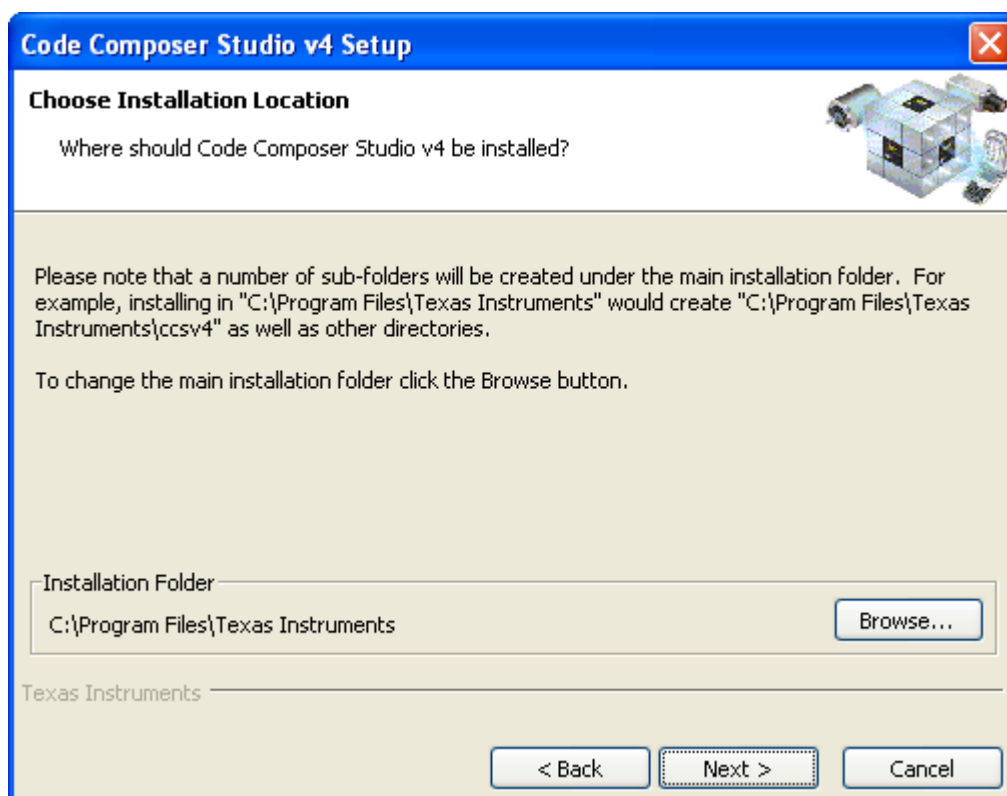
1. 将 DVD 光盘插入到 DVD 光驱中。
2. 运行光盘中的“setup_CCS_4.1.0.02000.exe”。
3. 出现安装画面，点击 Next



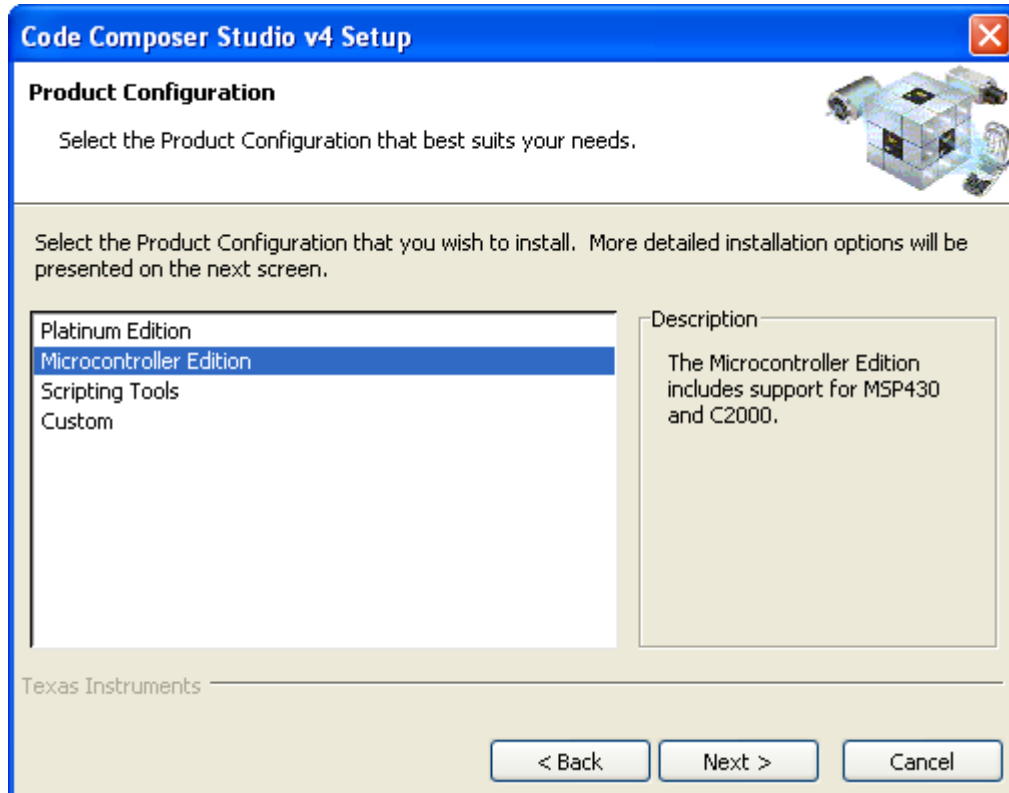
4. 点击 Next



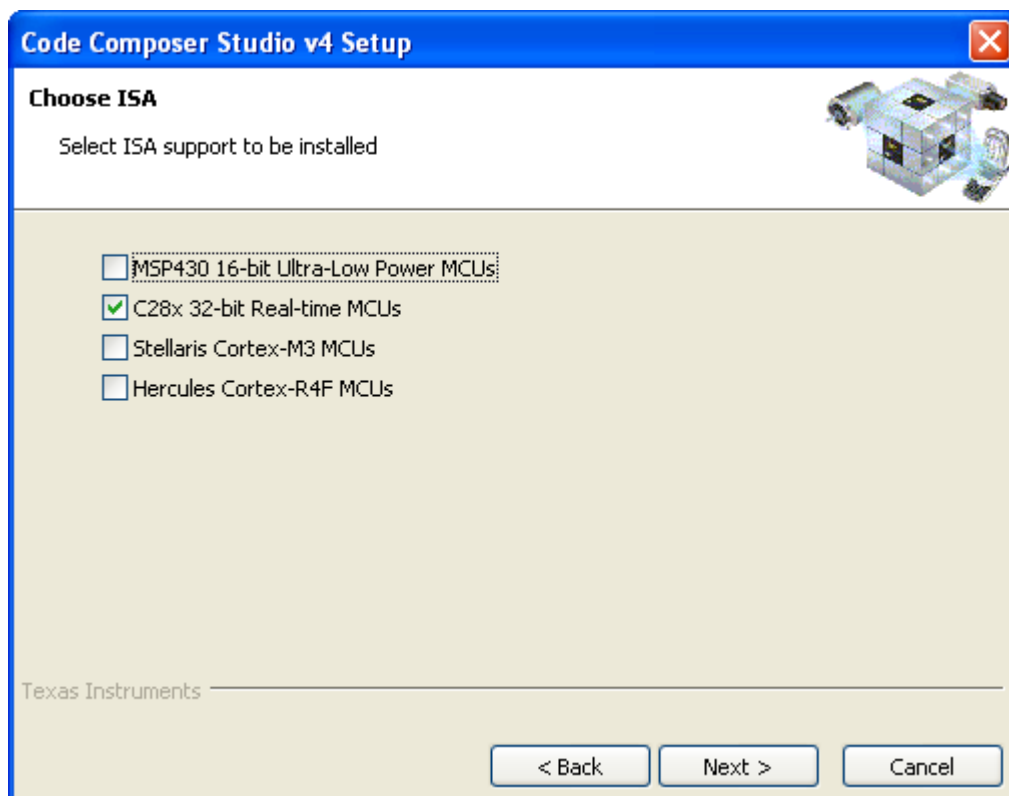
5. 选择 I accept the terms of the license agreement, 点击 Next



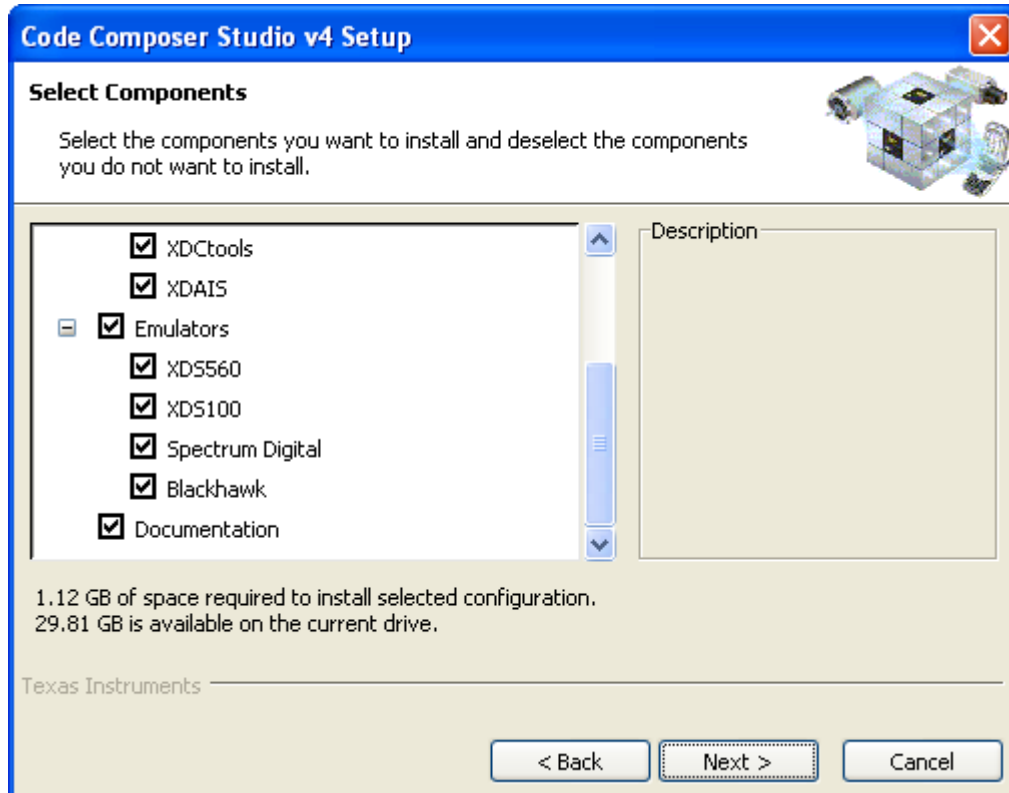
6. 根据您所使用的 TI 的平台, 选择合适的版本, 针对 C2000, 请选择 Microcontroller Edition, 点击 Next



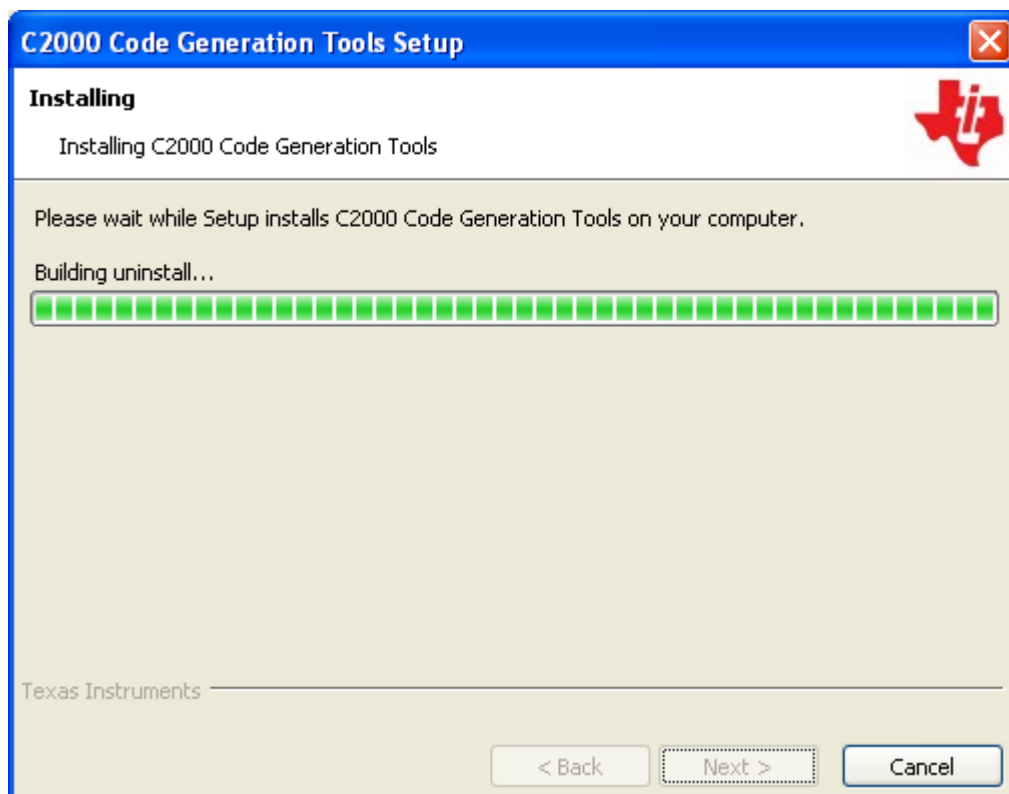
7. 针对 C2000，请选择 C28x 32-bit Real-time MCUs，点击 Next



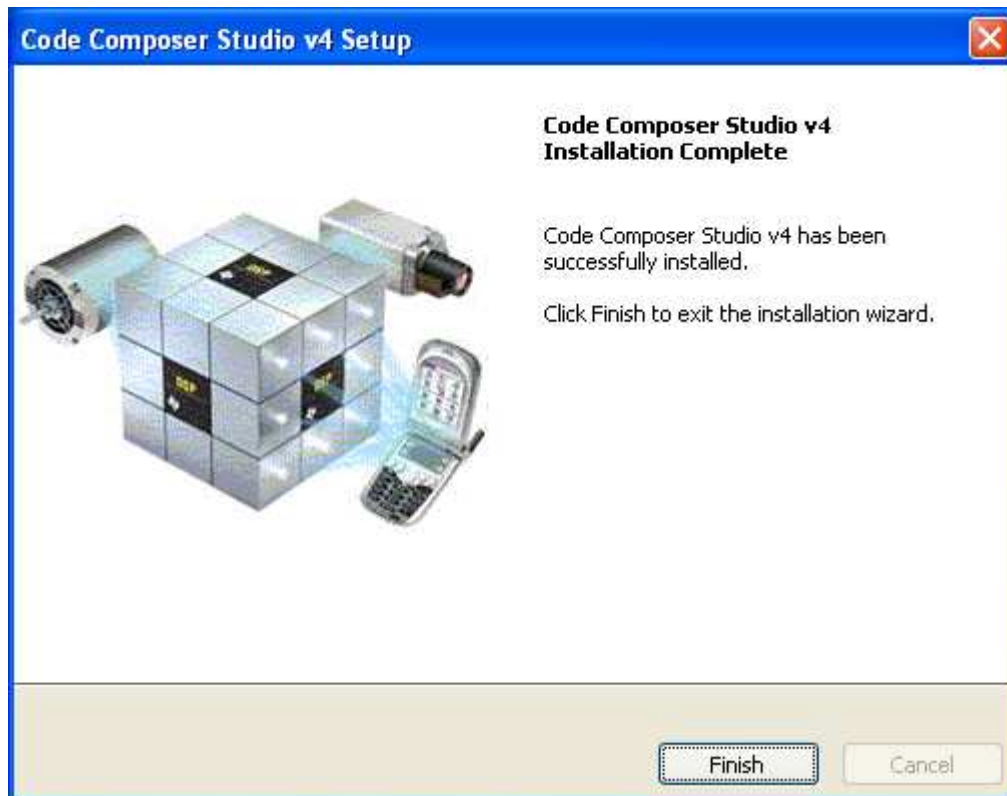
8. 按照默认的设置，点击 Next，默认情况下，已安装 XDS560、XDS100、Spectrum Digital、Blackhawk 仿真器的驱动。



9. 点击 Next，开始安装 CCSv4.



10. 点击 Finish，完成 CCSv4 的安装。



3、激活 CCSv4

CCSv4 可以在以下网址上下载:

<http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html>

	CCS-FREE	TMDSCCS-MCU	TMDSCCS-ALL
Name	FREE OPTIONS - Click Download to see all free options	MICROCONTROLLER - Supports MSP430, Stellaris, C28x and CortexR4F devices	PLATINUM - Supports all TI Embedded Processors
Status	ACTIVE	ACTIVE	ACTIVE
Price (US\$)	Free	\$445.00	\$1895.00
Current Version	v4.x	v4.x	v4.x
	Order Options Download	Order Options TI eStore TI Distributors	Order Options TI eStore TI Distributors

其中有两种免费的版本:

1. XDS100, 限制许可证。这个许可文件是指在利用 XDS100 进行板级在线仿真时是完全免费的, XDS100 可以针对所有 C2000 的芯片进行板上仿真。点击 CCS-FREE, 进行下载。
2. MCU 版本, 32K 代码限制。对于所有 C2000 芯片, 此版本只允许编译 32K 代码范围内的程序。注: 此版本不包含 DSP/BIOS。

3. 其他有关 CCSv4 注册许可的详解，请访问 Code Composer Studio Wiki 网站：
[CCSv4 Getting Started Guide](#) 或 [Code Composer Studio v4](#)

4、在 CCSv4 中运行第一个 C2000 项目

4.1 选择工作区

首次启动 CCSv4 时，将显示一个对话框，提示您选择工作区。该工作区是一个文件夹，用于保存特定个人计算机的所有 CCSv4 自定义设置。这些自定义设置中包括关闭 CCSv4 时的所有项目设置、宏和视图。例如，如果关闭 CCSv4 时您正同时处理多个项目，开着多个内存窗口和图形窗口，而且正在使用着多个监视变量，则当您重新打开 CCSV4 时，将显示与关闭前相同的项目和设置。



关闭 CCSv4 时，工作区将自动保存。如果不希望覆盖当前工作区的设置，则可以通过转至以下菜单选项打开或切换到新的工作区：“File -> Switch Workspace (文件 -> 切换工作区)”。

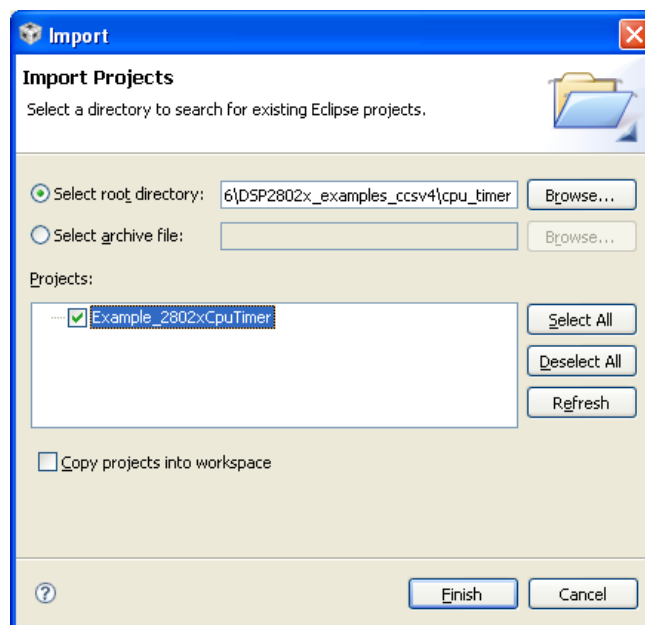
请注意，在 CCSv4 中，工作区不可移植，因为工作区中包含您个人计算机系统的特定数据。所以，您不能保存一个工作区，进行压缩，然后再发送给另一位用户。

4.2 运行 CCSv4 cpu_timer project

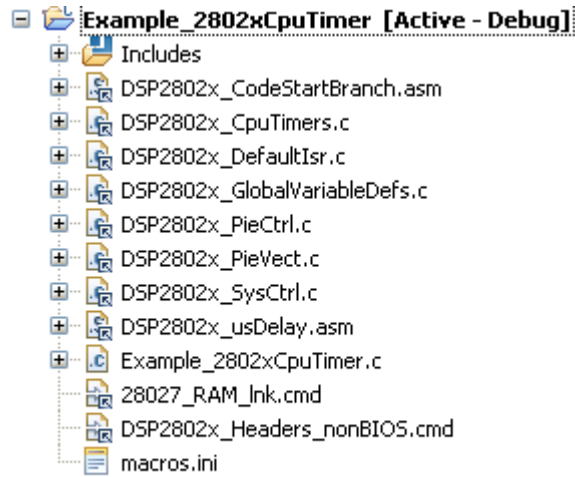
1. 打开“C2802x C/C++ Header Files and Peripheral Examples” (v1.26 或更高版本) 附带的 CCSv4 cpu_timer 项目
2. 转至“Project -> Import Existing CCS/CCE Eclipse Project (项目 -> 导入现有 CCS/CCE Eclipse 项目)”。



3. 单击“Select root directory: (选择根目录:)”旁边的“Browse (浏览)”按钮，浏览到“<header file base>/DSP2802x_examples_ccsv4/cpu_timer”目录文件夹，然后选择“OK (确定)”。请确保已在“Projects: (项目:)”文本框中选中“Example_2802xCpuTimer”。请勿选中“Copy projects into workspace (将项目复制到工作区)”复选框。单击“Finish (完成)”按钮。

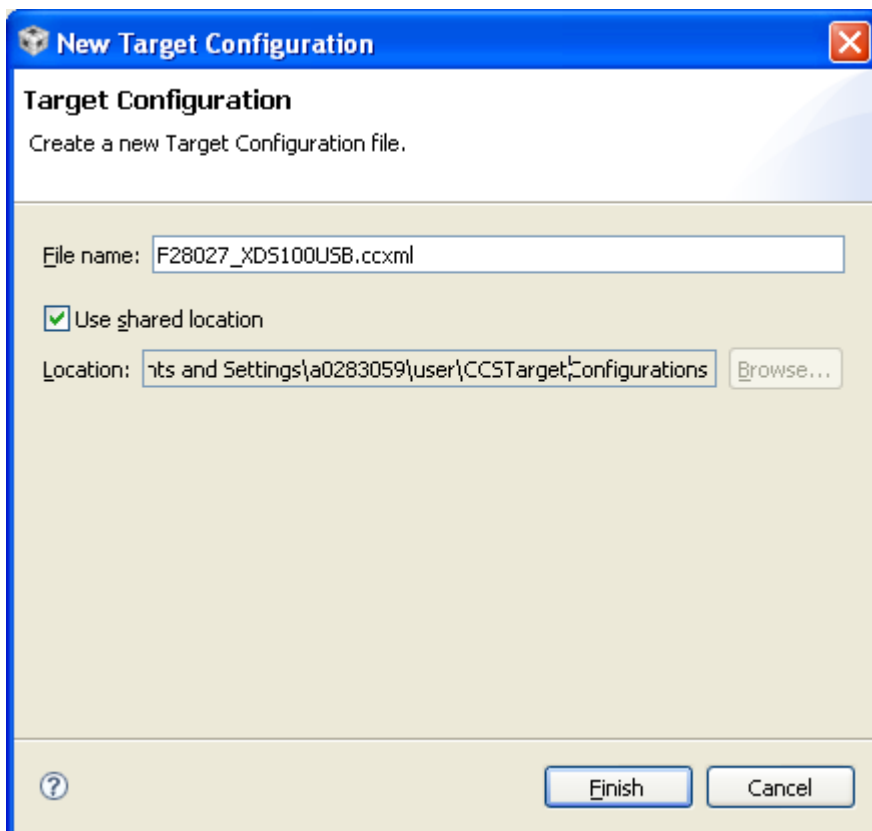


4. 您现在位于 C/C++ 透视视图中，项目文件将显示在“C/C++ Projects (C/C++ 项目)”窗口中。



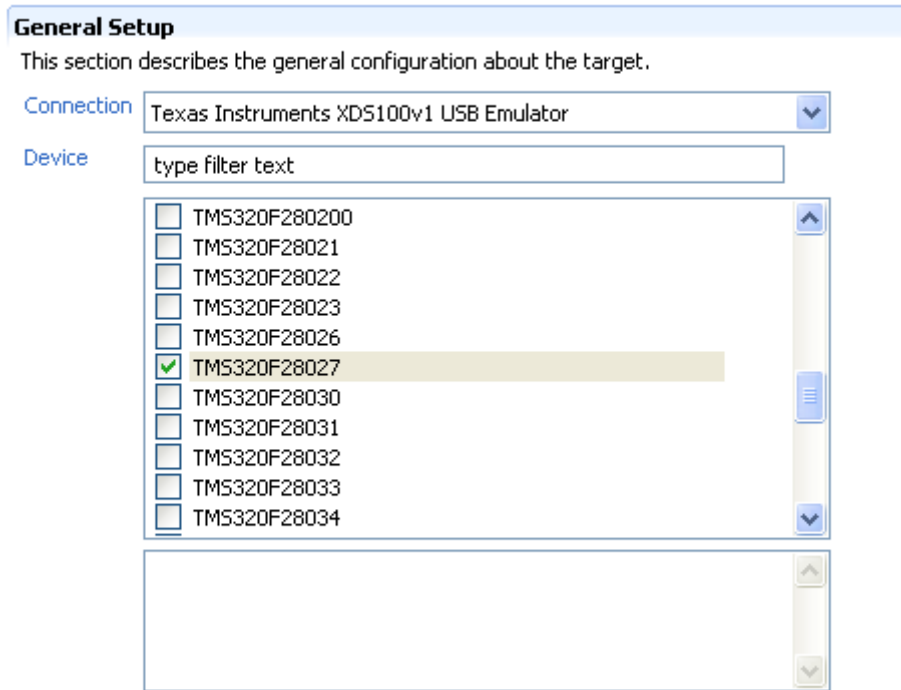
5. 为设备仿真设置目标配置选项

- 在当前示例中，要设置仿真器目标配置，请转至“Target -> New Target Configuration（目标 -> 新建目标配置）”。
- 请键入目标配置文件的名称，该文件的后缀为“.ccxml”。选中“Use default location（使用默认位置）”复选框，然后选择“Finish（完成）”。



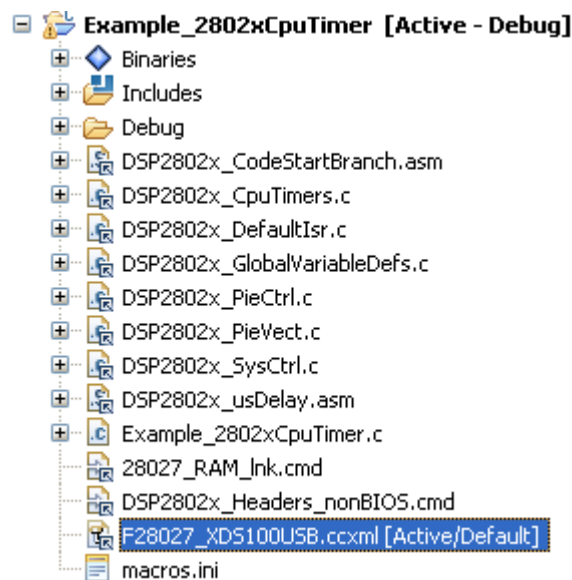
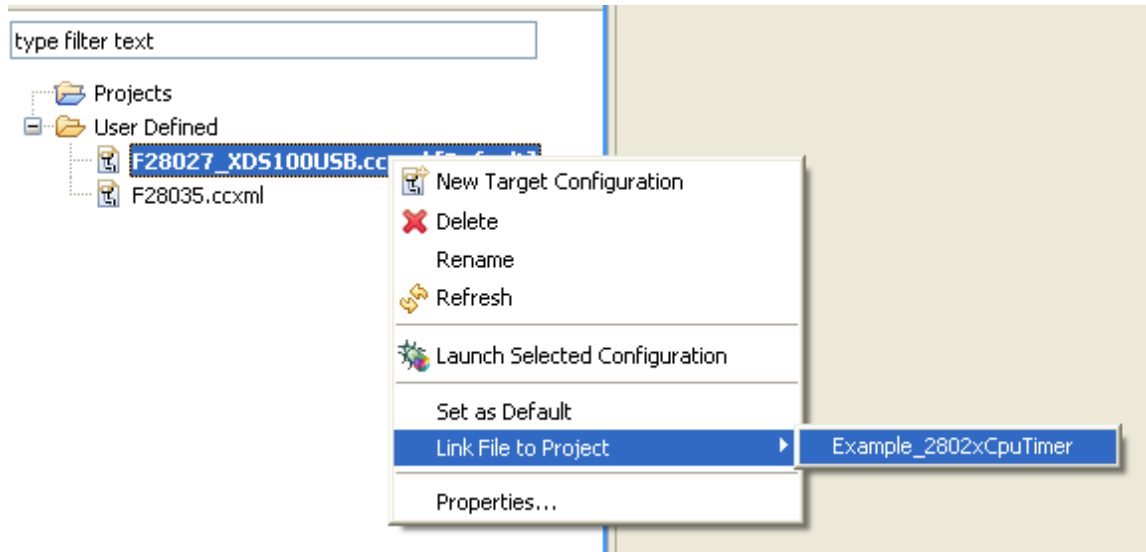
- 在显示的窗口中，通过“Connection（连接）”下拉列表选择仿真器，然后在“Device（设备）”列表中选择设备。在本示例中，已经选中“Texas Instruments XDS100v1 USB Emulator”和“TMS320F28027”设备复选框。

Basic



Note: Support for more devices may be available from the update manager.

- 然后，转至“File -> Save（文件 -> 保存）”。
- 现在，转至“View -> Target Configuration（查看 -> 目标配置）”。
- 在“User-Defined”下，将显示新的目标配置 .ccxml 文件，右键单击该文件。
 - 选择“Link File to Project -> Example_2802xCpuTimer（将文件链接到项目 -> Example_2802xCpuTimer）”，将在工作区中的特定项目始终使用该目标配置。
 - 或者，选择“Set as Default（设置为默认值）”（该默认设置适用于：希望以后在所有工作区中打开的未指定目标配置的项目都默认使用此仿真器和设备 GEL 文件）。



- 在 Binaries 下 - 建立项目后，项目的 .out 文件位于此处。
- 在 Includes 下 - 生成选项所引用的 Include 指令的所有目录都在此处列出。展开目录将列出这些目录中所有的 .h 和 .inc 文件，项目也可能不会使用这些文件。
- 在 Debug 下 - 所有调试文件（包括 .map 和 .obj 文件）都位于此处。
- 之后将列出所有源文件。如果有任意源文件包含函数、include 文件或结构，选择位于源文件旁边的“+”号，将列出该文件中包含的所有函数、include 文件和结构名称。双击函数名，将转到在源代码中定义函数、include 文件和结构的位置。

6. 生成和加载项目

- 将所有活动项目一次性全部加载到目标中

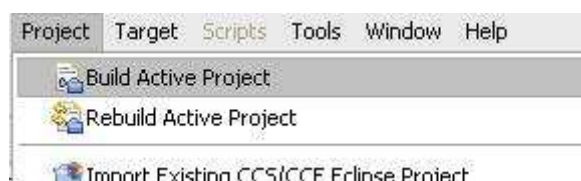
要构建项目，请首先连接到目标，将所有活动项目一次性全部加载到目标中，然后单击工具

栏上的绿色昆虫图样按钮，即“Debug（调试）”。接着跳到下方的“重置 CPU”步骤。

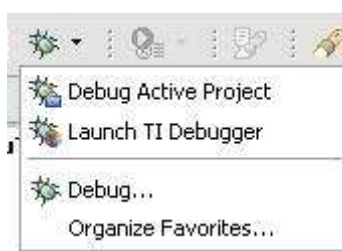


- 不自动连接目标

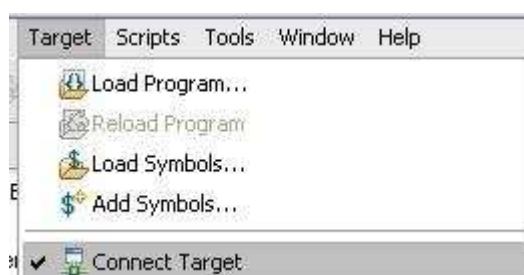
要构建项目，但不自动连接目标并将活动项目对象代码加载到设备，请转至“Project -> Build Active Project（项目 -> 建立活动项目）”。



然后，选择工具栏上绿色昆虫图样按钮旁边的向下箭头，然后选择“Launch TI Debugger（启动 TI 调试程序）”，此时将位于“Debug（调试）”透视视图中。



转至“Target -> Connect Target（目标 -> 连接目标）”，连接到目标设备。



转至“Target -> Load Program（目标 -> 加载程序）”，然后浏览到“Example_2802xCpuTimer.out”（位于“/cpu_timer/Debug”目录中）。

7. 重置 CPU

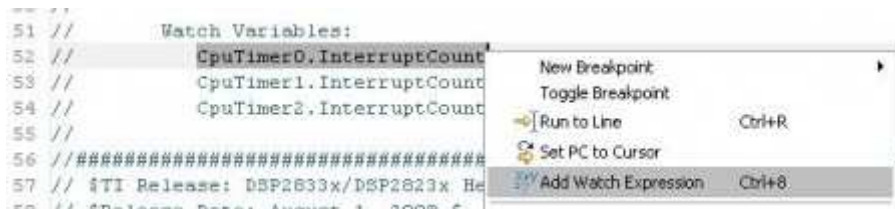
依次选择“Target -> Reset -> Reset CPU（目标 -> 重置 -> 重置 CPU）”，重置设备。要重新启动程序，请依次选择“Target -> Restart（目标 -> 重新启动）”。

8. 将监视变量添加到监视窗口

- 如果“Example_2802xCpuTimer.c”尚未打开，请在屏幕的右上角，单击 C/C++ 小图标，切换回“C/C++”透视视图，然后打开源文件。此时，单击屏幕右上角绿色昆虫图标旁边的“Debug（调试）”小图标，切换回“Debug（调试）”透视视图。

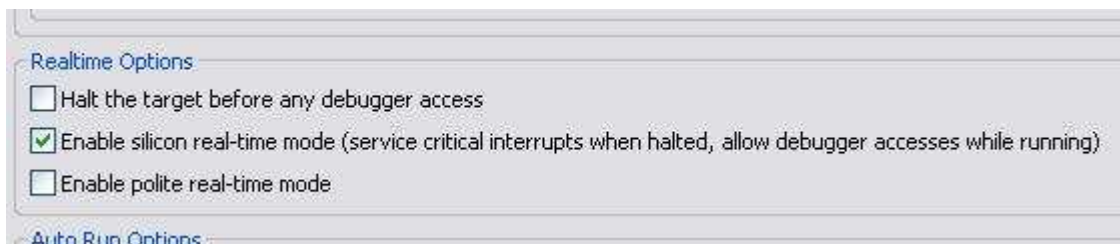


- 在“Example_280xxCpuTimer.c”中，在位于文件顶部的注释中向下滚动，直到到达“Watch Variables（监视变量）”。突出显示“CpuTimer0.InterruptCount”，单击右键，然后选择“Add Watch Expression（添加监视表达式）”。之后，请依次对“CpuTimer1.InterruptCount”和“CpuTimer2.InterruptCount”重复执行上述步骤。现在，这些变量将显示在监视窗口中。



9. 启用实时模式以在监视变量发生变化时进行监视

- 转至“Tools -> Debugger Options->Generic Debugger Options（工具 -> 调试程序选项）”。在打开的窗口中，向下滚动到“Realtime Options（实时选项）”，然后选择“Enable silicon real-time mode (service critical interrupts when halted, allow debugger accesses while running)（启用硅片实时模式(停止时关键服务中断，允许调试程序在运行时访问)）”。如果弹出一个对话框，询问是否允许实时切换，请选择“Yes（是）”。



- 在“Watch”监视窗口的右上角，单击白色向下箭头，然后选择“Customize Continuous Refresh Interval...（自定义持续刷新时间间隔...）”。将持续刷新时间间隔更改为“1”（秒），替换默认值“5”（秒）。



- 在监视窗口的右上角，单击暂停符号上以环形旋转的黄色箭头图标，启用持续刷新模式，让系统在程序运行时持续刷新监视变量。



9. 运行程序

- 选择工具栏上的绿色箭头图标，或者转至“Target -> Run（目标 -> 运行）”。



- 观察位于监视窗口中的变量在 CPU 定时器中断时每秒增加一次。

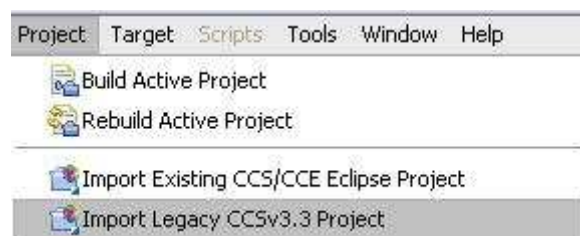
Name	Value	Address	Type
(x) CpuTimer0.InterruptCount	13	0x00008A12@Data	unsigned long
(x) CpuTimer1.InterruptCount	13	0x00008A02@Data	unsigned long
(x) CpuTimer2.InterruptCount	13	0x00008A0A@Data	unsigned long
<new>			

- 祝贺您！您刚刚在 CCSv4 中运行完第一个 C2000 项目！

5、将 C2000 项目从 CCSv3.3 迁移到 CCSv4

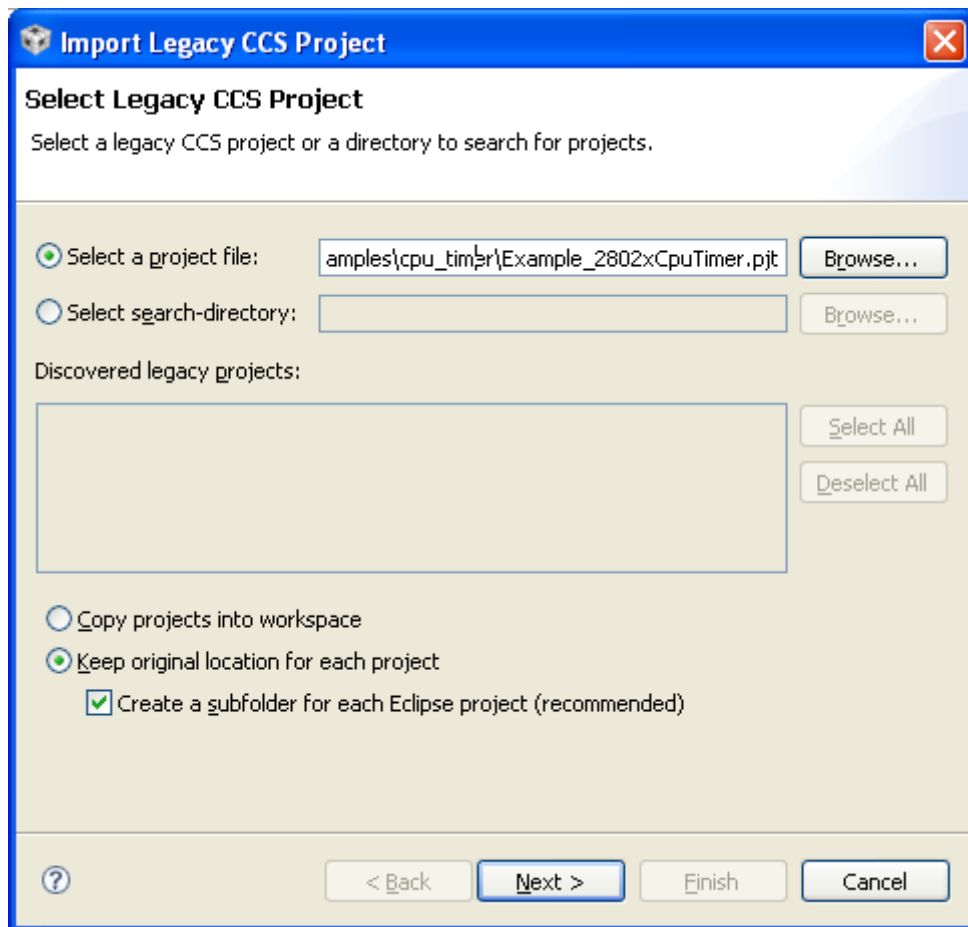
使用 Code Composer Studio v4，用户可以将项目从 CCSv3.3 迁移到 CCSv4，而且用户可以根据需要一次迁移一个或多个项目。通过这种方式，如果在一个目录中存在多个需要从 CCSv3.x 迁移到 CCSv4 的项目，则可以轻松地一次性迁移所有项目。

- 要开始迁移进程，请转至“Project -> Import Legacy CCSv3.3 Project（项目 -> 导入旧版 CCSv3.3 项目）”，打开“Project Migration Wizard（项目迁移向导）”。

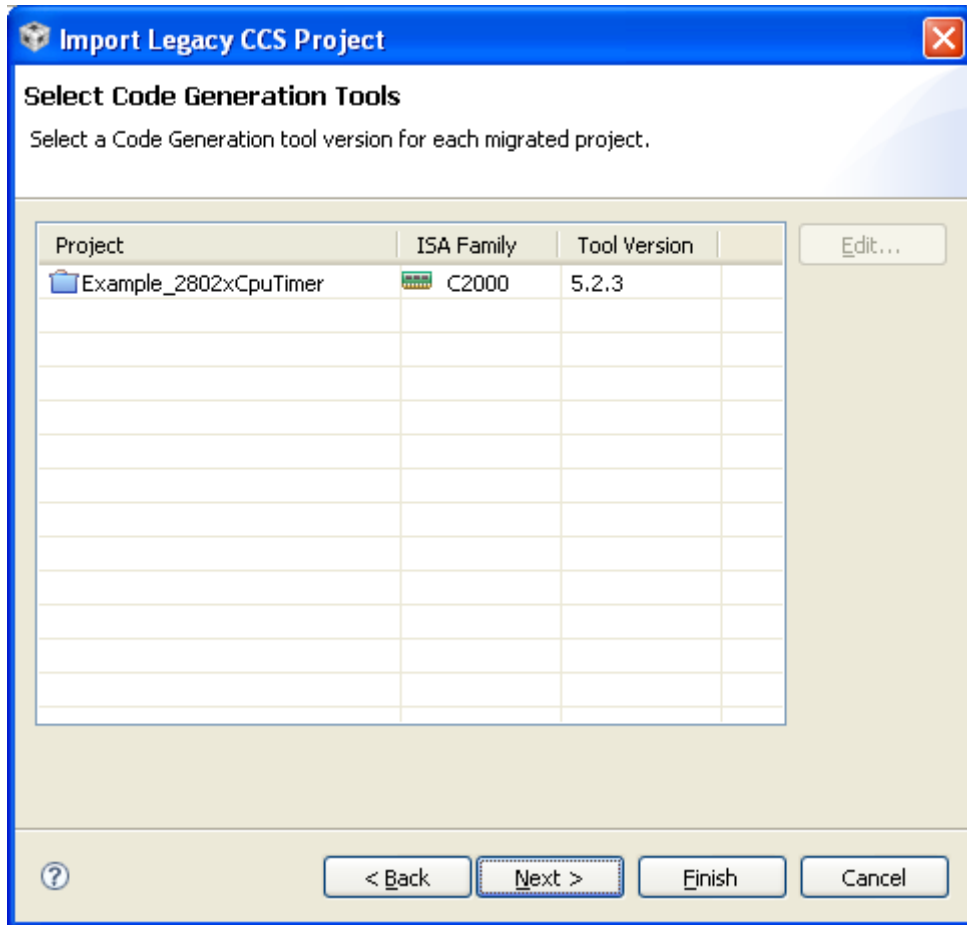


5.1 将一个项目从 CCSv3.3 迁移到 CCSv4

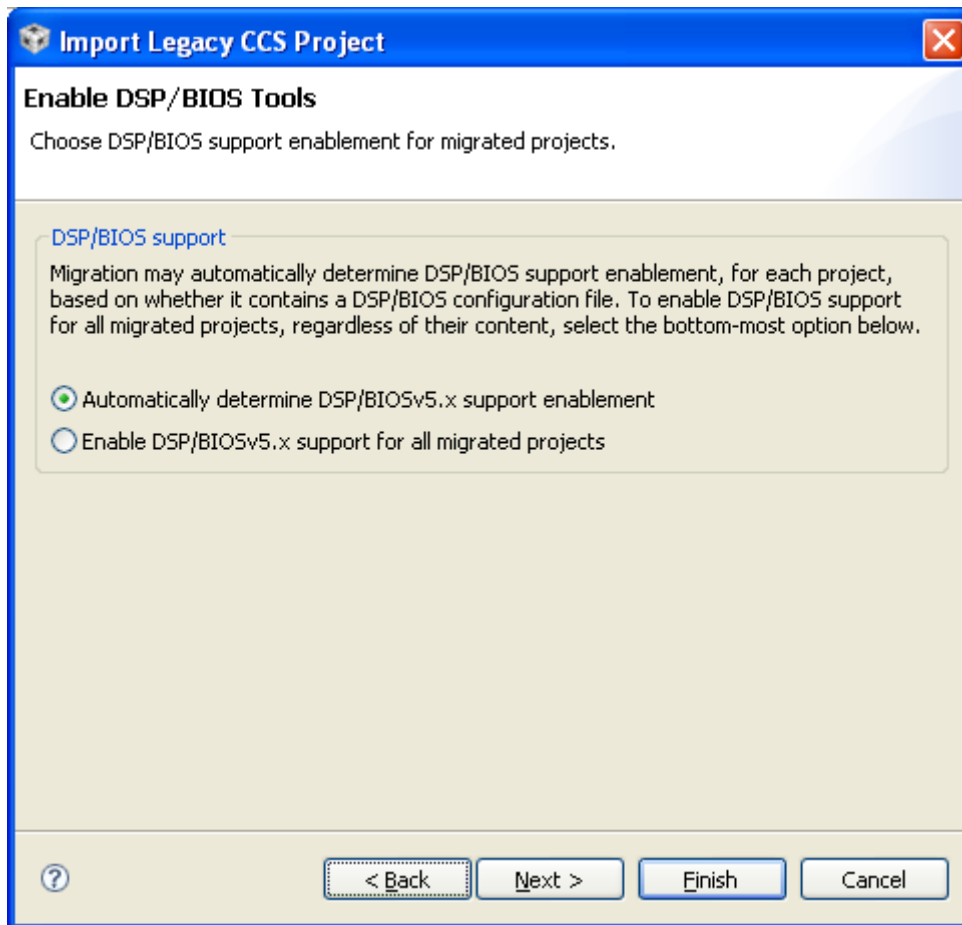
- 在“Project Wizard (项目向导)”中，选择“Select Project: (选择项目:)”单选按钮，然后浏览到要迁移的 CCSv3.3 .pjt 项目文件。在下图中，已选中 Example_2802xCpuTimer.pjt 文件。然后单击“Next (下一步)”按钮。



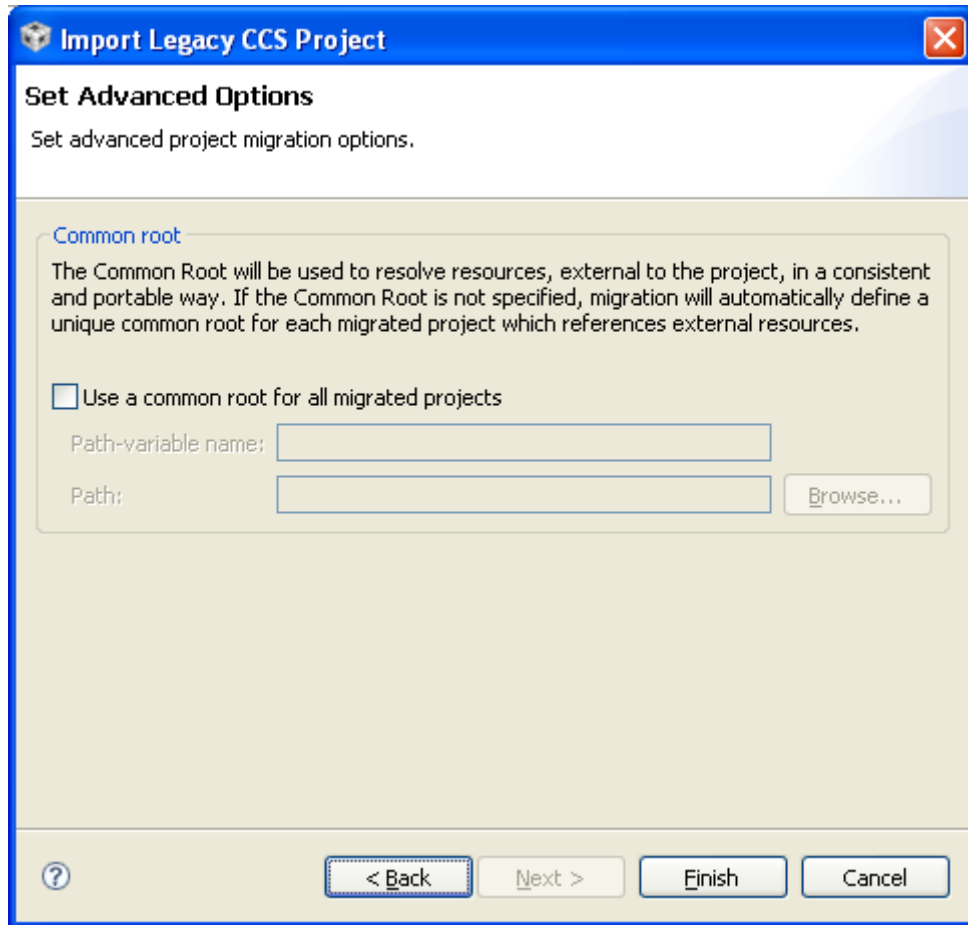
- 请选择构建所选项目所需的“Code Generation Tool (代码生成工具)”的版本。



- 如果旧版项目中包含对 DSP/BIOS 的引用，则请将其删除，即使项目并不使用 DSP/BIOS，也请执行此项操作（有关详细信息，请参阅 [C2000 Code Composer Studio v4 入门#应用 DSP BIOS 所引起的源文件缺失](#)）。如果您要使用 DSP/BIOS，则请选择适用于项目的 DSP/BIOS 工具版本。默认条件下，虽然 C2000 示例不使用 DSP/BIOS，但由于 .PJT 文件中包含一个对“DspBiosBuilder”的空引用，迁移向导将选择默认的 DSP/BIOS 工具版本。单击“Next（下一步）”按钮。



- 取消选中“Use common root for all migrated projects (对所有已迁移项目使用公共根目录)”，使 CCS 自动确定项目所引用的所有文件的唯一公共根目录，然后单击“Finish (完成)”。(有关为项目迁移和已生成的 macros.ini 文件定义“公用根目录”的详细信息，请参阅 [CCSv4 中针对 C2000 的可移植项目](#))。



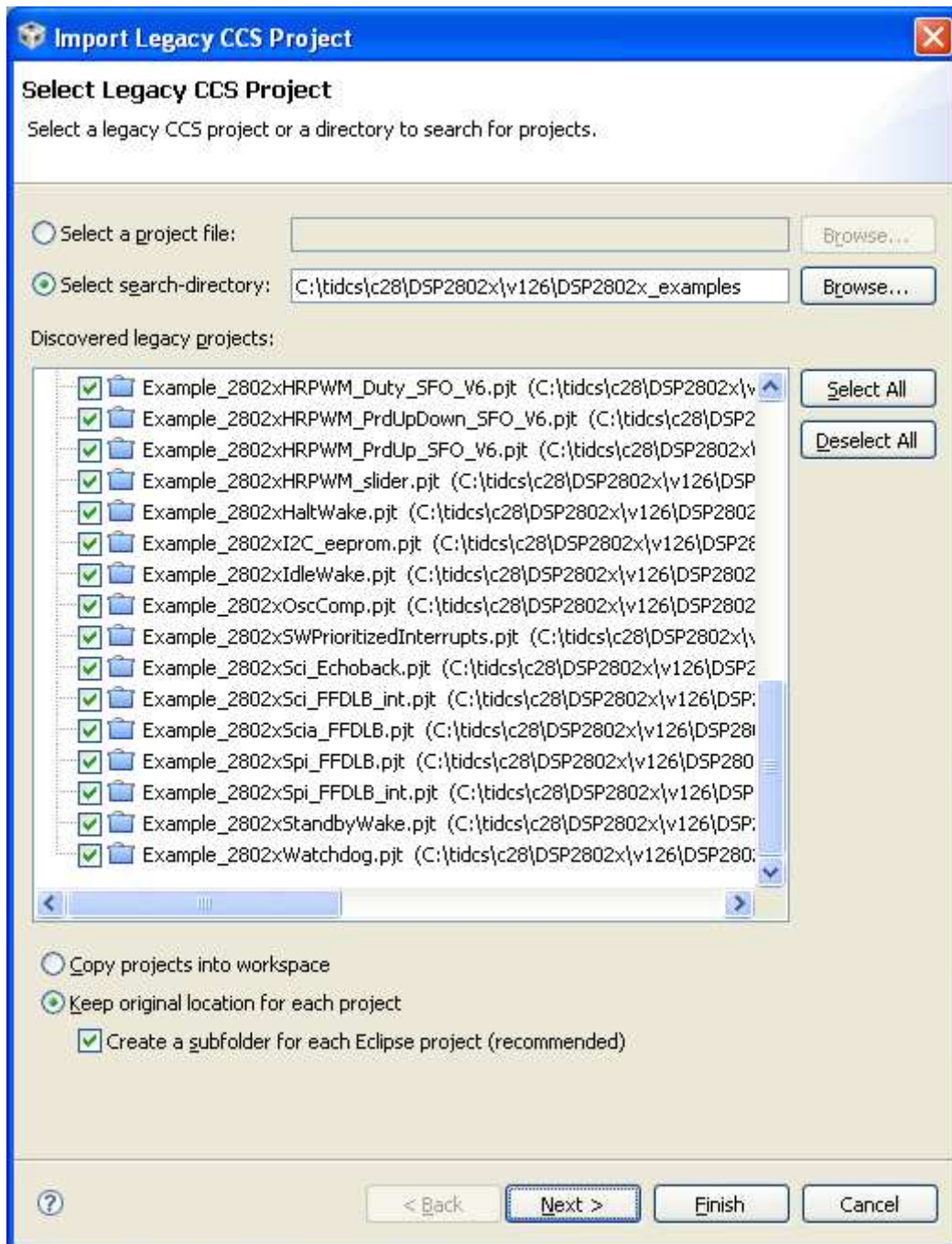
祝贺您！您已经将一个 CCSv3.x 项目成功迁移到 CCSv4。您现在可以从项目文件夹中删除所有文件，无需保存 CCSv3.3 .pj1 文件,但下列文件除外：

1. .cdtproject
2. .cdtbuild
3. .project
4. 项目目录中的所有源文件和头文件（.c、.asm、.h、.inc、.cmd、.lib 等）

5.2 将多个项目同时从 CCSv3.3 迁移到 CCSv4

- 在“Project Wizard（项目向导）”中，选择“Select search-directory:（选择搜索目录：）”单选按钮，然后浏览到要迁移的所有或部分 CCSv3.3 项目文件夹所在的目录。在下图中，选择的目录为：“C:/tidcs/c28/DSP2802x/v126/DSP2802x_examples/”。迁移向导随后将搜索所选目录下的所有旧版项目，而且这些项目也会显示在“Discovered Legacy Projects:（发现的旧版项目：）”框中。选中要迁移的项目，然后单击“Next（下一步）”按钮。
- 下一步，选择构建所选项目所需的“Code Generation Tools（代码生成工具版本）”（与单个项目迁移相同）。

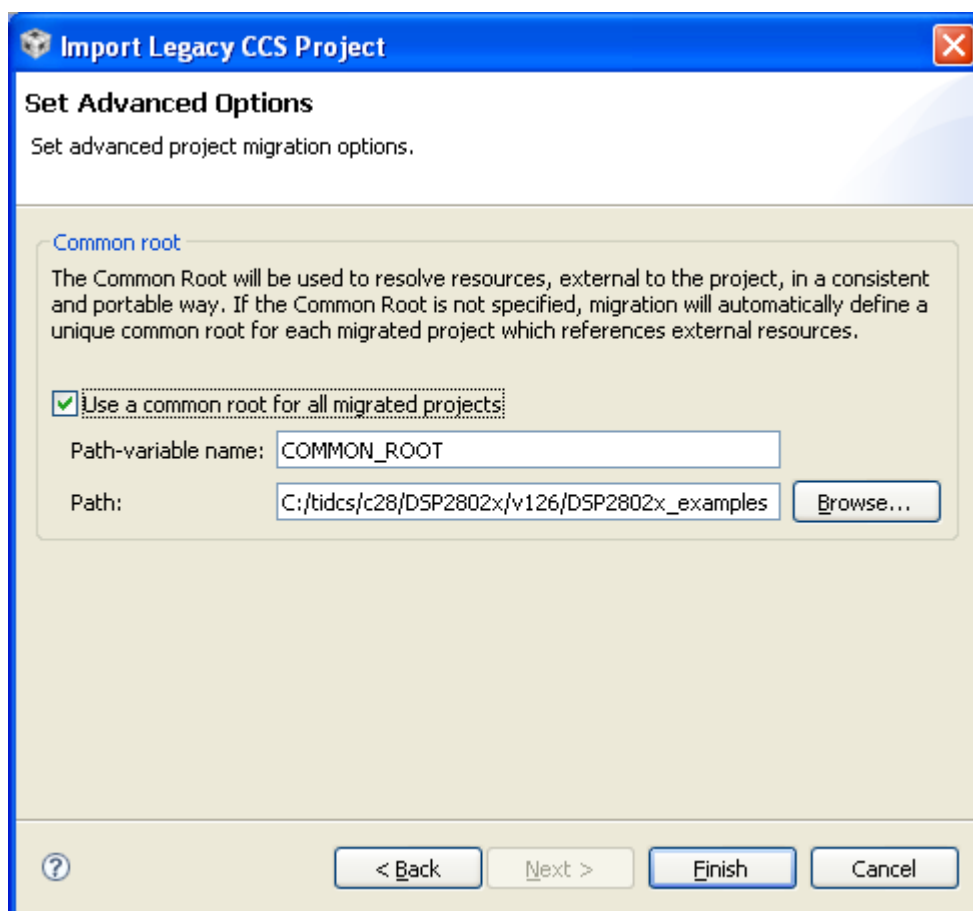
- 如果旧版项目中包含对 DSP/BIOS 的引用，则请选择适用于项目的 DSP/BIOS 工具版本。默认情况下，虽然 C2000 示例不使用 DSP/BIOS，但由于 .PJT 文件中包含一个对“DspBiosBuilder”的空引用，迁移向导将选择默认的 DSP/BIOS 工具版本，单击“Next（下一步）”按钮。（与单个项目迁移相同）



- 下一步，选择构建所选项目所需的“Code Generation Tools（代码生成工具）”版本（与单个项目迁移相同）。
- 如果旧版项目中包含对 DSP/BIOS 的引用，则请选择适用于项目的 DSP/BIOS 工具版本。默认情况下，虽然 C2000 示例不使用 DSP/BIOS，但由于 .PJT 文件中包

含一个对“DspBiosBuilder”的空引用，迁移向导将选择默认的 DSP/BIOS 工具版本，单击“Next（下一步）”按钮。（与单个项目迁移相同）

- 在“Set Advanced Options（设置高级选项）”窗口中，选中“Use a common root for all migrated projects（对所有迁移项目使用公用根目录）”复选框。对于包含多个项目且其文件共享公用根目录的情况，请选中此复选框。

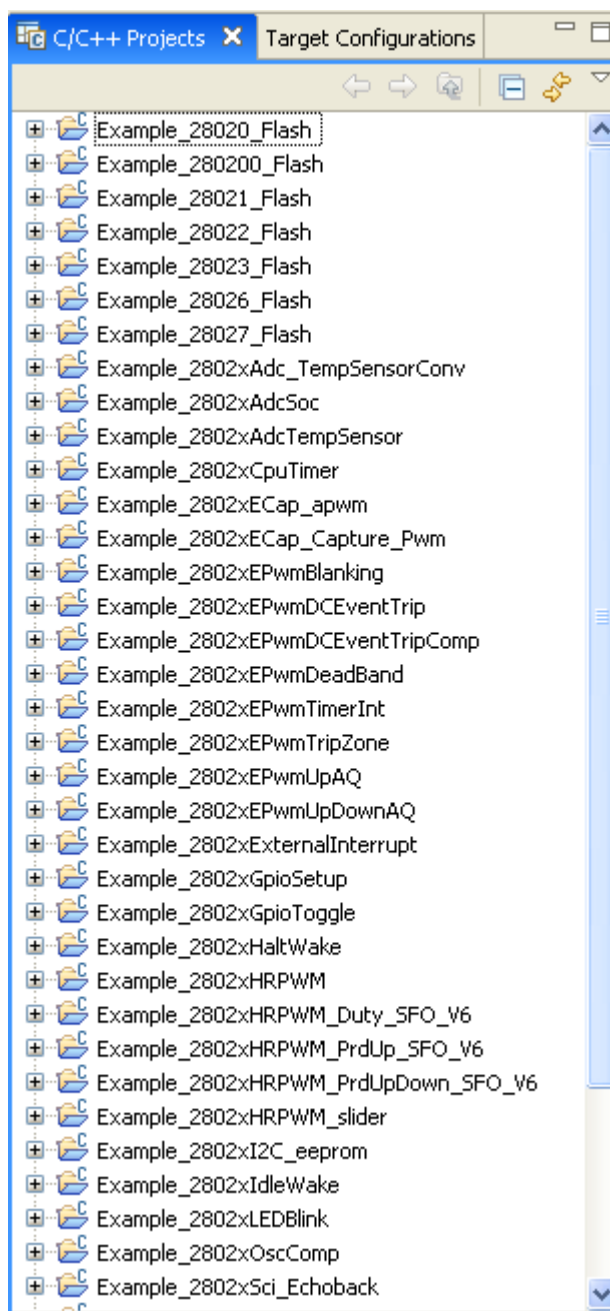


例如，下图显示了要迁移项目的目录文件结构。此示例中，要迁移的所有项目都位于“DSP2802x_examples”目录中的单独文件夹中。所有这些项目在 DSP2802x_common 和 DSP2802x_headers 目录下包含公用头文件、源文件和库，这两个目录又位于“v126”根目录下。

因为所有迁移项目都使用“C:/tidcs/c28/DSP2802x/v126”根目录下的公用文件，所以指定的“路径”应为“C:/tidcs/c28/DSP2802x/v126”。同时也请选择一个唯一的“路径变量名”，代替键入整个路径的宏/替换名称。在此示例中，选择的是“COMMON_ROOT”。

注：在指定工作区中，对某个特定路径只能定义一次路径变量名/宏。也就是说，不能将同一工作区中不同路径下的另一项目再次定义为“INSTALLROOT_2833X_V130”。而必须定义不同的路径变量/宏。您可以在下一页中更加详细地了解从用户到用户，或从个人计算机到个人计算机传输项目时有关宏及其无用情况的信息：[Portable Projects in CCSv4 for C2000（CCSv4 中针对 C2000 的可移植项目）](#)。

完成后，单击“Finish（完成）”按钮。



祝贺您！您已经将多个 CCSv3.x 项目迁移到 CCSv4。现在您可以从项目文件夹中删除所有文件，无需保存任何 CCSv3.3 .pjx 文件。但是下列文件除外：

1. .cdtproject
2. .cdtbuild
3. .project
4. 项目目录中的所有源文件和头文件（.c、.asm、.h、.inc、.cmd、.lib 等）

6、C28x 闪存编程

CCSv4 对 CCSv3.3 的一项改进是闪存编程器插件现在内建于 CCSv4 中。当调试针对闪存配置了源代码和命令链接器文件的项目时，不再需要仅加载符号，然后单独使用闪存编程器插件将代码写到闪存中。取而代之的是，只要加载正确的设备目标配置，如果 CCSv4 通过您项目中的命令链接器文件检测到闪存中有代码，在加载程序时，它会自动将代码写入闪存。

要在 CCSv4 中设置闪存编程设置并将程序写入闪存，请执行下列操作：

1. 首先确保已经为您的设备正确设置了您的目标配置。
2. 转至“Target -> Launch TI Debugger (目标 -> 启动 TI 调试程序)”
3. 在“Debug (调试)”视图中，转至“Tools -> On-chip Flash (工具 -> 片上闪存)”
4. 配置闪存设置 – 也可以在此处擦除闪存。
5. 然后，可以：单击“Debug (调试)”昆虫图标，或转至“Target -> Debug Active Project (目标 -> 调试活动项目)”，或转至“Target -> Load program (目标 -> 加载程序)”并选择要编程的 .out 文件。CCSv4 IDE 将自动编程或将代码加载到闪存和/或 RAM 中相应的存储位置，类似于基于 RAM 的项目代码自动加载到 RAM 的方式。

On-Chip Flash (TMS320C28xx) ?

Clock Configuration

OSCCLK (Mhz): 10

CLKINDV: 2

PLLCR Value: 12

Flash Program Setting:

Erase, Program, Verify

Program, Verify

Load RAM Only

Erase Sector Selection

Sector A: (0x3F6000 - 0x3F7FFF)

Sector B: (0x3F4000 - 0x3F5FFF)

Sector C: (0x3F2000 - 0x3F3FFF)

Sector D: (0x3F0000 - 0x3F1FFF)

Erase Flash

Code Security Password

Key 7 (0xAE7): FFFF

Key 6 (0xAE6): FFFF

Key 5 (0xAE5): FFFF

Key 4 (0xAE4): FFFF

Key 3 (0xAE3): FFFF

Key 2 (0xAE2): FFFF

Key 1 (0xAE1): FFFF

Key 0 (0xAE0): FFFF

Program Password Lock Unlock

注：与 CCSv3.3 一样，源代码和链接器命令文件仍然需要预先针对闪存选项进行配置（有关如何操作的说明，请参阅安装包中 /docs 文件夹内的设备特定 C/C++ 头文件和外围设备示例快速入门自述文件）。

7、C2000 GEL 文件迁移

来自 CCSv3.3 的设备 GEL 文件仍然能够用于 CCSv4.0。不再支持用于处理项目（即生成项目）的调用以及用于向监视窗口添加变量的“WatchAdd()”函数。GEL 热菜单项目现在出现在“Scripts（脚本）”下拉菜单中。

有关 GEL 文件的更多信息，包括如何加载 GEL 以及哪些 GEL 函数仍受支持，请访问 CCSv4 GEL Wiki 网页：[GEL](#)

设备 GEL 中的寄存器 WatchAdd() 由“View -> Registers（查看 -> 寄存器）”替换

C2000 设备 GEL 文件包括用于所有外围设备寄存器的“WatchAdd()”函数，允许用户在不使用 TI 提供的 C/C++ 头文件时将任何外围设备的寄存器快速添加到 CCSv3 中的监视窗口。因为 WatchAdd() 函数在 CCSV4 中不再有效，现在可以通过“Registers（寄存器）”窗口使用外围设备寄存器，该窗口可从“View -> Registers（查看 -> 寄存器）”访问。如果在调试时使用 TI 提供的 C/C++ 头文件（可从 TI 网站的设备产品文件夹下载），在头文件中定义的寄存器结构也可以如以前一样添加到监视窗口。展开这些寄存器结构可访问单独的位字段（例如将“AdcRegs”添加到监视窗口将显示全部 ADC 寄存器以及它们的位字段扩展）。

8、Control Suite 软件说明

8.1 Control Suite 简介

Control Suite 对于 C2000 微处理器来说是一个多钟软件的集合，通过 Control Suite 用户可以显著地减少开发时间，软件中包含了很多特性可以增强软件的可用性。Control Suite 既是一个学习平台也是一个开发平台，能够适应不同的用户的开发经验和编程偏好。

8.2 Control Suite 特性

Control Suite 的主要特性如下：

1. Control Suite 是所有 C2000 软件的一个集合体
2. 智能安装从而减少了版本之间的依赖性和兼容性的问题。
3. 软件安装之后会产生直观的生成文件组织结构
4. 用户可以选择三种不同的 MCU 编程方案
5. 对于用户的整个设计阶段均提供模块化的软件库和历程

6. 针对典型应用领域的应用软件库和程序框架，例如数字电机控制和数字电源。
7. 完整的系统开发套件，硬件和软件完全源码开放。
8. 丰富的说明文档提供原理及执行的细致描述。

8.3 Control Suite 当前版本中的组件

开发套件

- Piccolo controlSTICK (TMDX28027USB)
 - With Piccolo F2802x
- High Voltage Motor Control Kit (TMDSHVMTRPFCKIT)
 - With Piccolo F2803x
 - With Delfino F2833x
- High Voltage Power Factor Correction Kit (TMDSHVPFCKIT)
 - With Piccolo F2802x
- Low Voltage Motor Control Kit (TMDS1MTRPFCKIT and TMDS2MTRPFCKIT)
 - With Piccolo F2803x
- LED DC/DC Lighting Kit (TMDSDCDCLEDKIT)
 - With Piccolo F2803x

芯片支持包

- Piccolo F2802x, Piccolo F2803x, Delfino F2833x, Delfino C2834x
 - Bit Fields
 - Peripheral Examples
 - Hardware Developer's Package

库

- 电机控制库
 - For C28x core (on all C2000 28x microcontrollers)
- 数字电源库
 - For C28x core (on all C2000 28x microcontrollers)
 - For Control Law Accelerator (CLA)
- 数学库
 - Fixed-Point (IQMath)
 - Floating Point (fastRTS Library)
 - CLA
- 工具
 - Flash API (for all Piccolo and Delfino devices)
 - Boot ROM source code (for all Piccolo and Delfino devices)

8.4 Control Suite 支持的芯片

Control SUITE 现在支持所有的 Piccolo 和 Delfino 芯片，具体芯片列表如下

- TMS320F28035
- TMS320F28034
- TMS320F28033

- TMS320F28032
- TMS320F28031
- TMS320F28030
- TMS320F28027
- TMS320F28026
- TMS320F28023
- TMS320F28022
- TMS320F28021
- TMS320F28020
- TMS320F280200
- TMS320F28335
- TMS320F28334
- TMS320F28332
- TMS320C28346
- TMS320C28345
- TMS320C28344
- TMS320C28343
- TMS320C28342
- TMS320C28341

8.5 Control SUITE 下载及安装

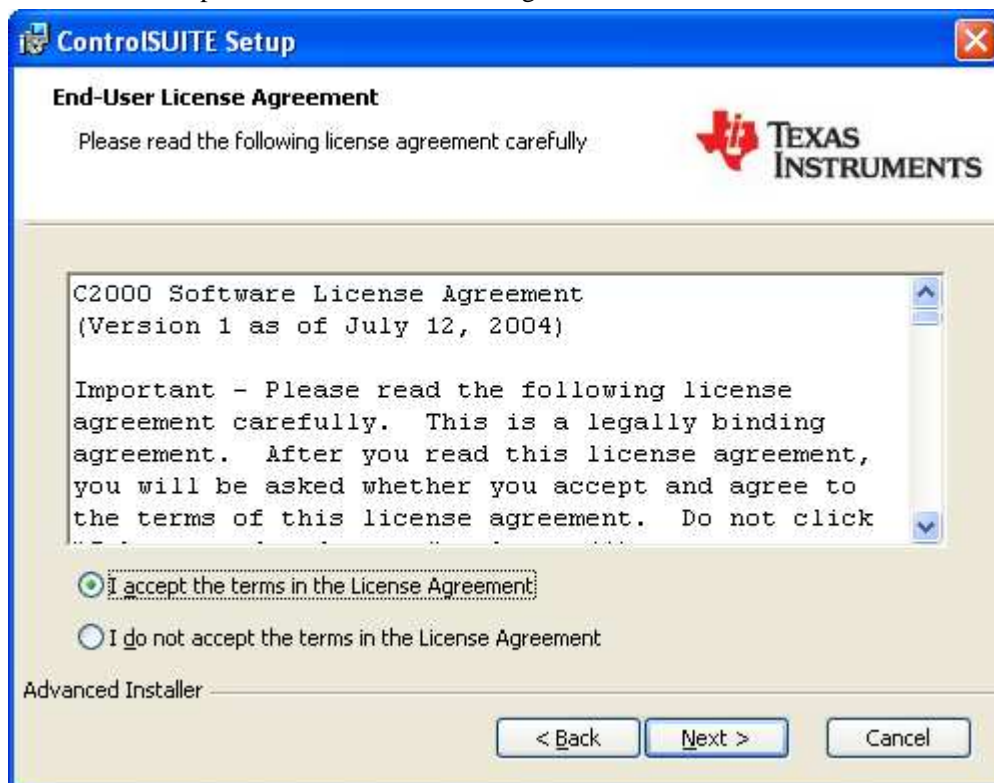
1. 登陆 www.ti.com/c2000, 点击 control SUITE
2. 点击下面的图标



3. 注册一个 TI 的账户即可免费下载 Control SUITE
4. 双击下载的 controlsuiteinstaller.exe 文件,出现以下界面, 点击 Next



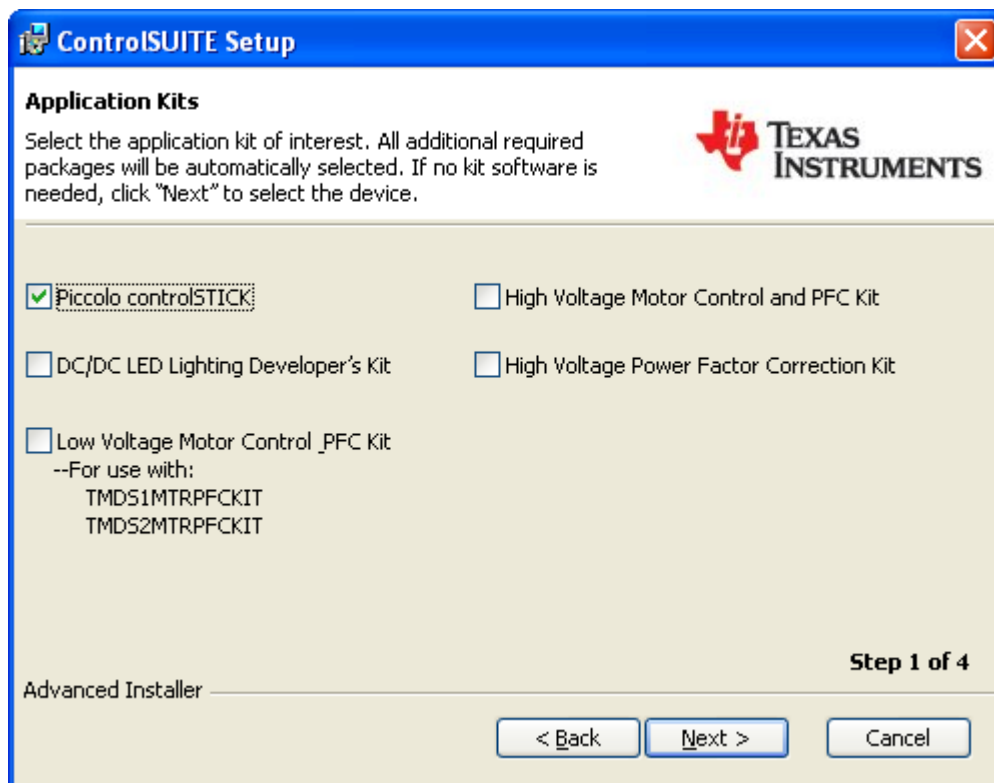
5. 选择 I accept the terms in the License Agreement, 点击 Next



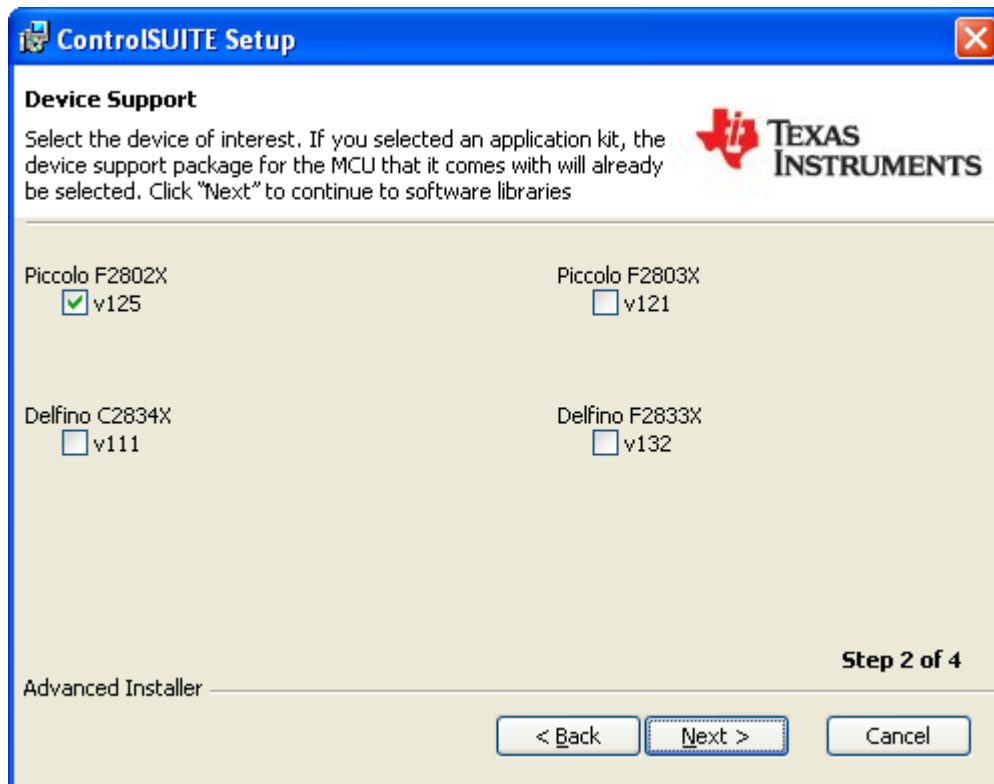
6. 选择安装的路径，默认放在 C:\TI 下，建议安装在此路径下。点击 Next



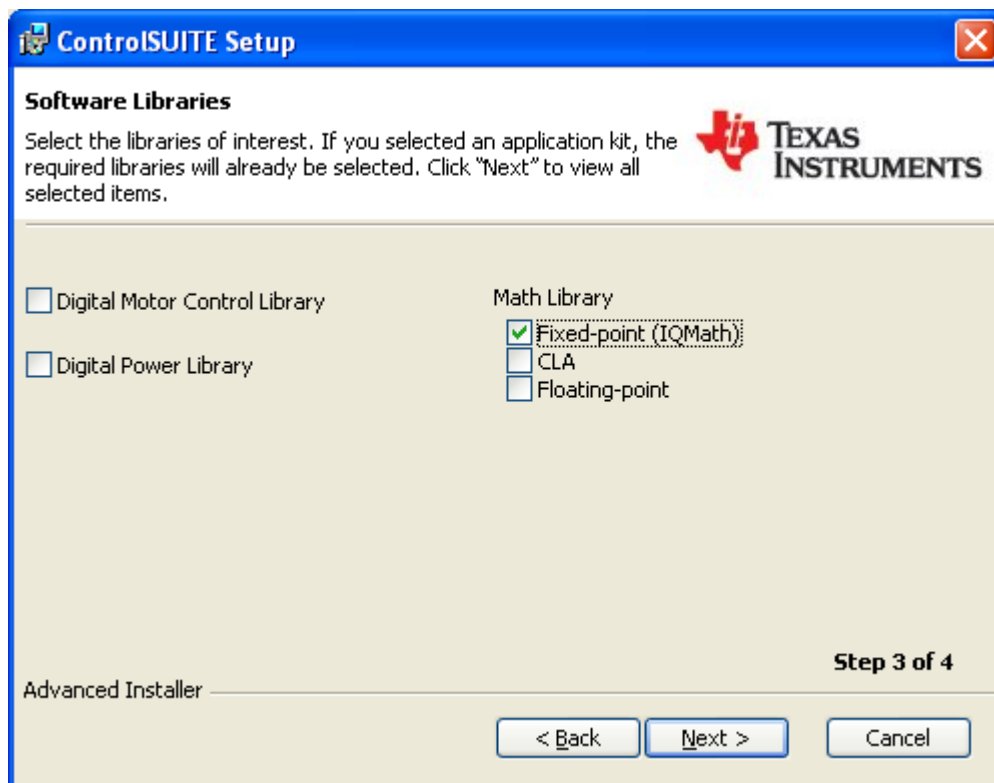
7. 选择要安装的开发套件，点击 Next



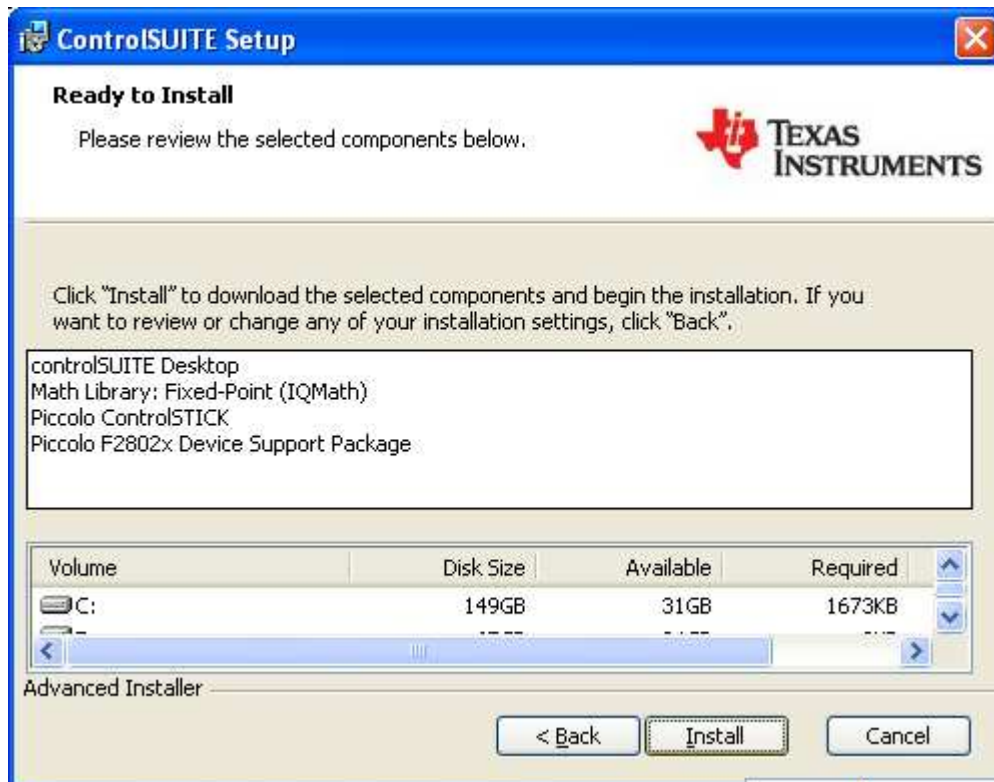
8. 选择所选用的芯片系列，点击 Next



9. 选择项目中所需要添加的库，点击 Next



10. 选择是否自动更新 Control SUITE，点击 Next



11. 上图中，列出了要安装的文件立标，点击 Install，开始下载和安装所选文件。

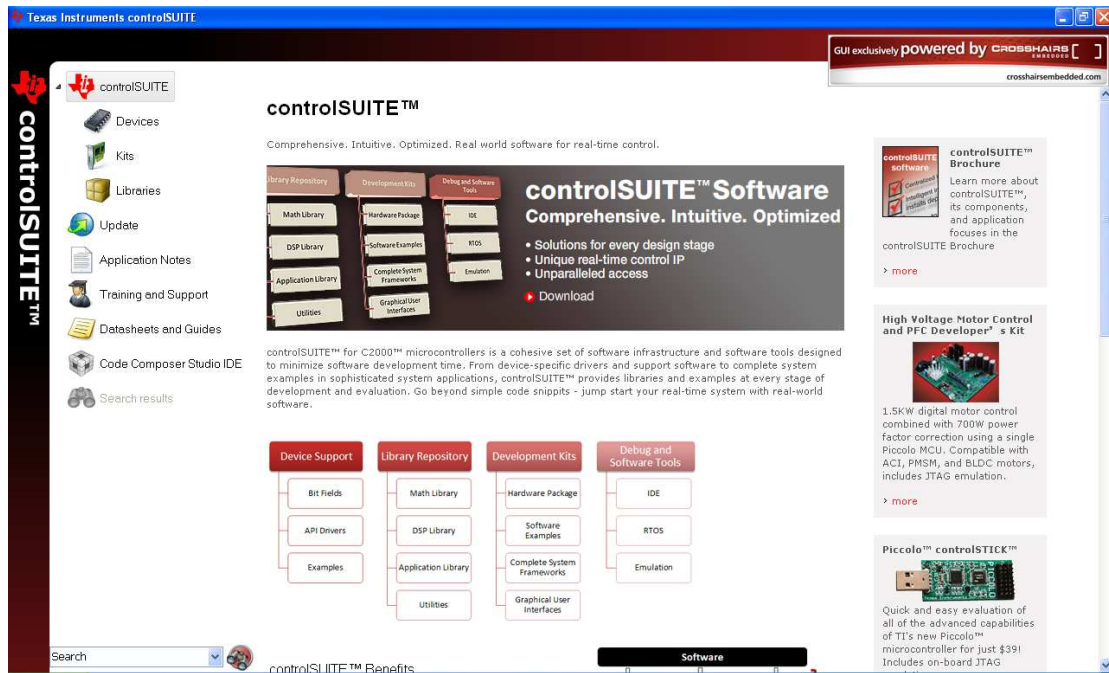


12. 点击 Finish，完成安装，进入 ControlSUITE 界面。

8.6 Control SUITE 的桌面

Control SUITE 桌面允许用户轻松地访问 C2000 的各种开发平台和各个芯片的硬件、软件以

及说明文档，同时 Control SUITE 还可以帮助用户在线查询有关芯片的用户手册（User Guide）、应用文档（Application Notes）和一些培训的教程。



8.7 芯片、开发套件、库

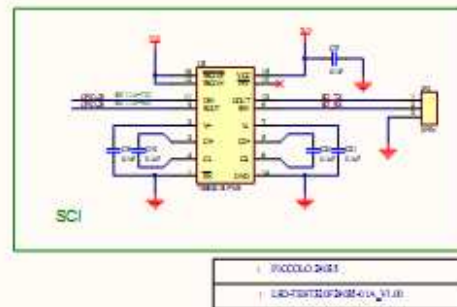
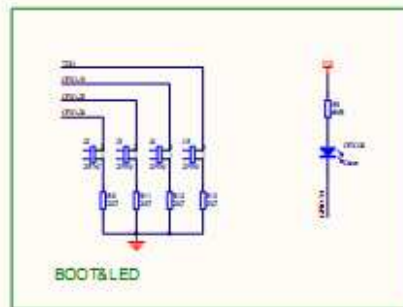
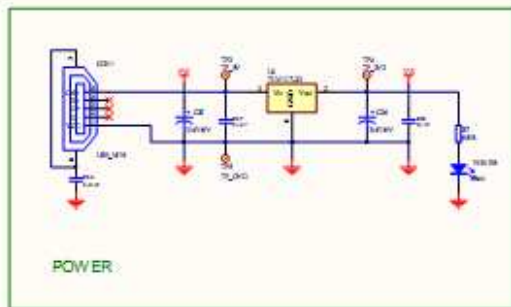
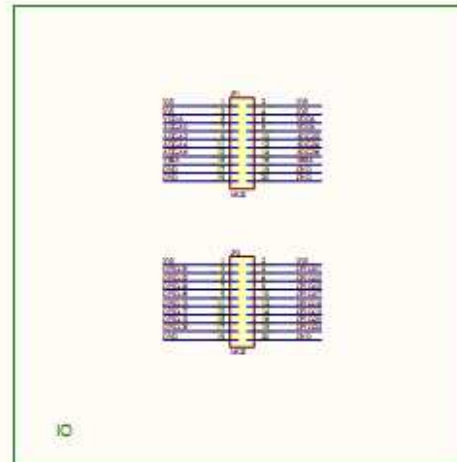
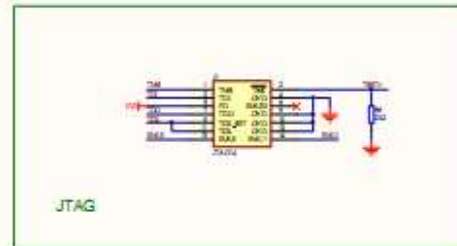
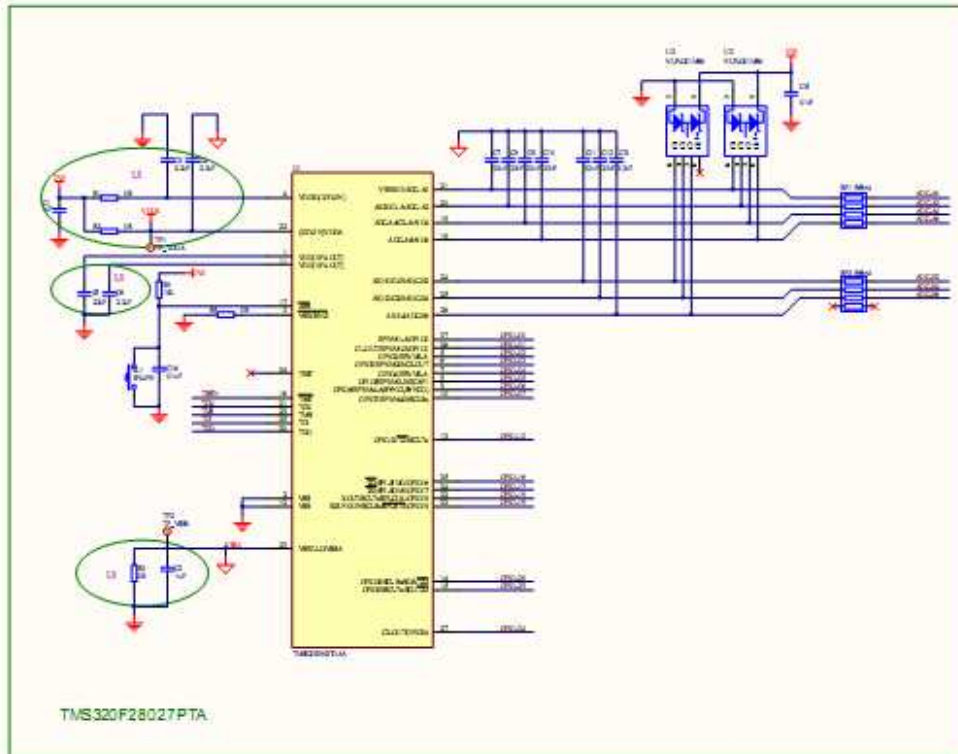
Control SUITE 桌面，包含了 C2000 的芯片，一些开发套件和库文件。其中芯片（Device）、开发套件（Kits）和库（Libraries）中分别包含了该选项的概览、历程、说明文档、在线资源和一些其他的有用的内容。

其中在 Devices 下，有 Piccolo F2802x、Piccolo F2803x、Delfino F2833x、Delfino C2834x 等产品系列，其中每个产品下都有相应的历程。例如在 Piccolo F2802x 下，有 Examples Projects 项目栏，双击其中的任何一个历程，可直接打开对应的程序目录；也可以通过双击 View all 35+ example projects，打开 Piccolo F2802x 所有的历程目录。

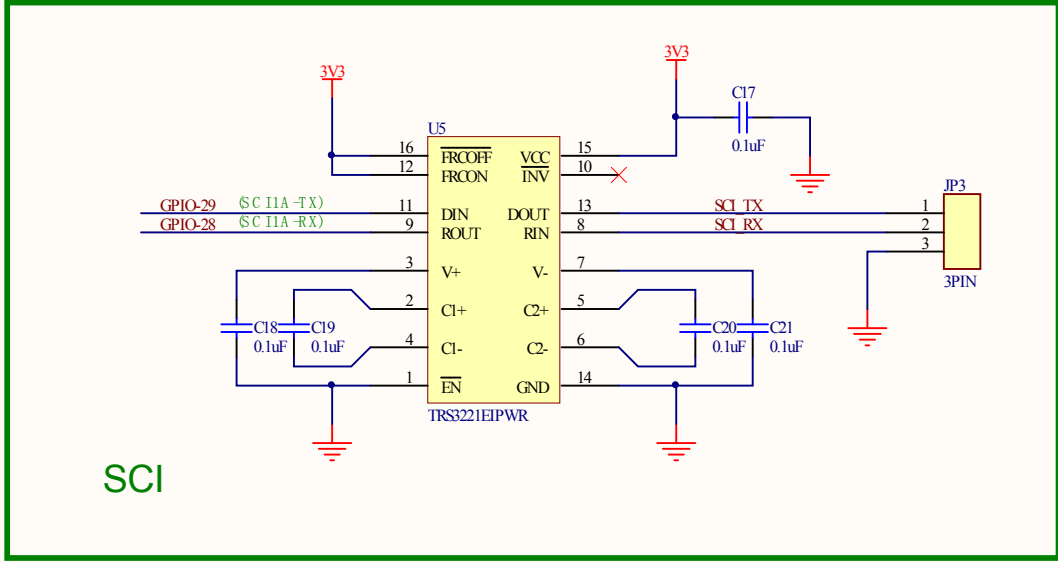
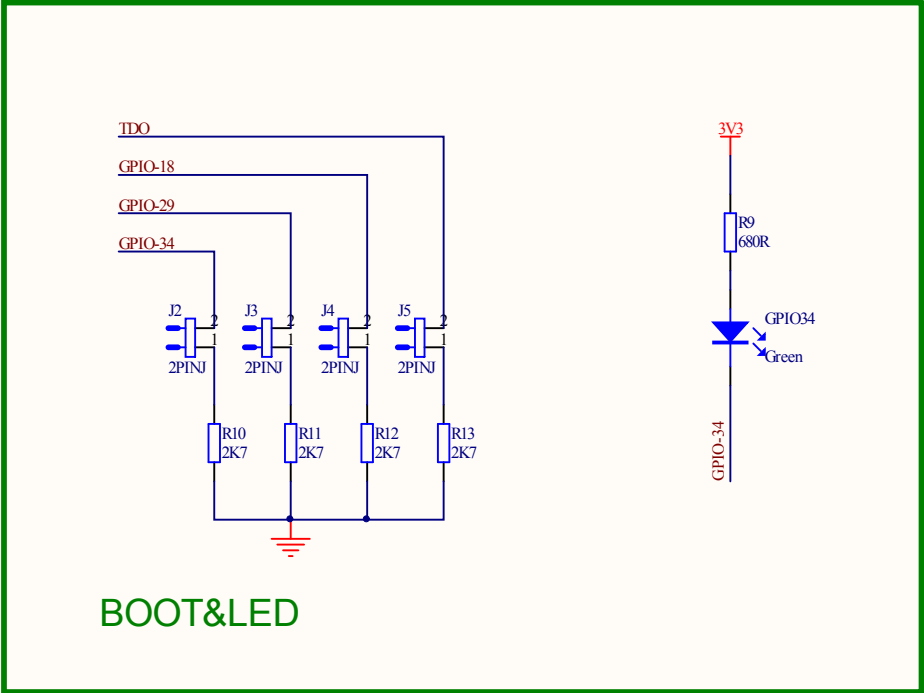
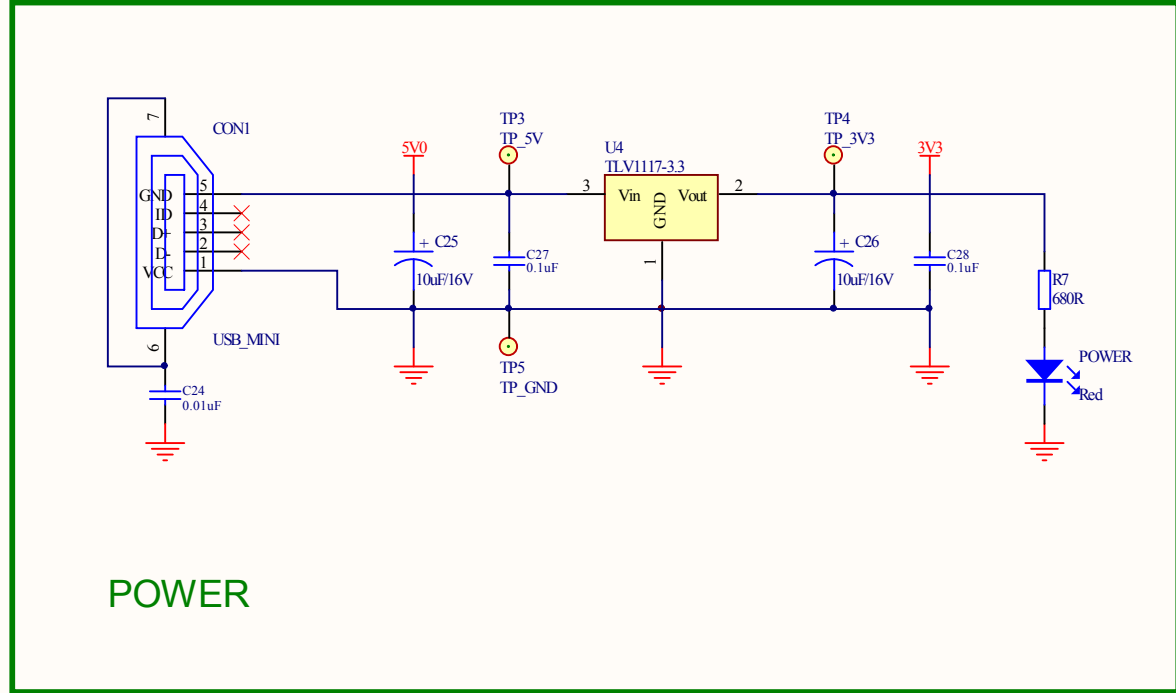
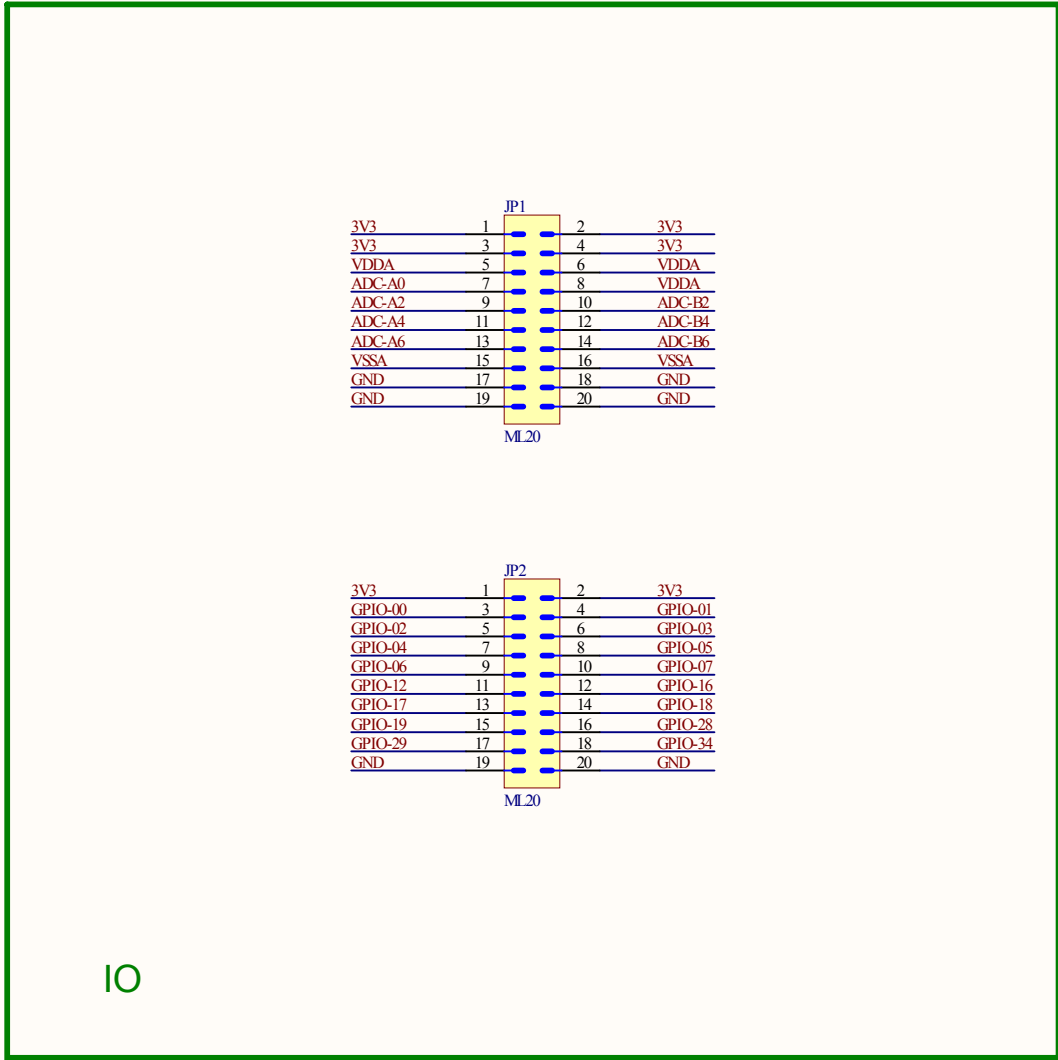
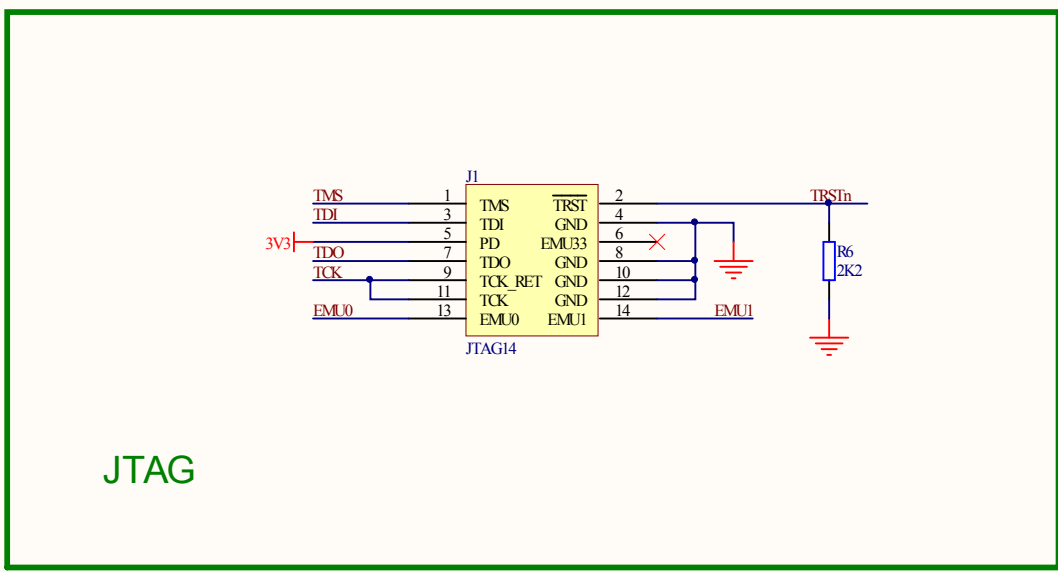
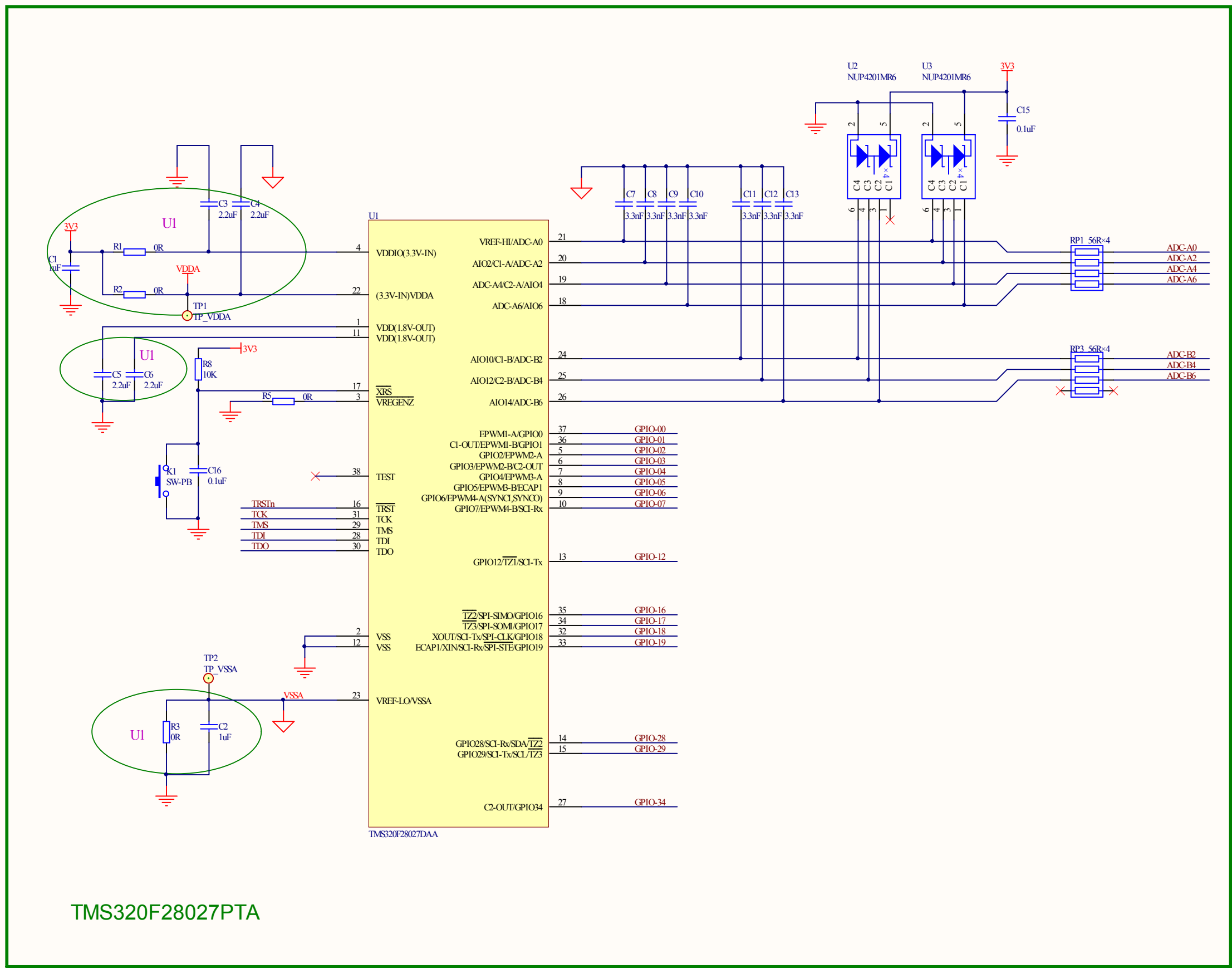
<p>Example Projects (To run, import in CCSv4)</p> <ul style="list-style-type: none"> View all 35+ example projects (To run, import in CCSv4) ADC SOC CPU Timer eCAP Capture PWM ePWM Timer Interrupts SCI Echoback GPIO Setup 	<p>Documentation</p> <ul style="list-style-type: none"> Headerfiles Quickstart ISR Priorities Example Internal Oscillator Functions SFO Library Errata Documentation folder
<p>Online Resources</p> <ul style="list-style-type: none"> Piccolo Product Selector C2000 tools Piccolo F28027 Product Folder Technical Docs Search 	<p>Utilities</p> <ul style="list-style-type: none"> Flash API Boot ROM Source Code F2802x Hardware Developer's Package

9、开发套件电路图

9.1 TMS320F28027DA-38pin 电路图

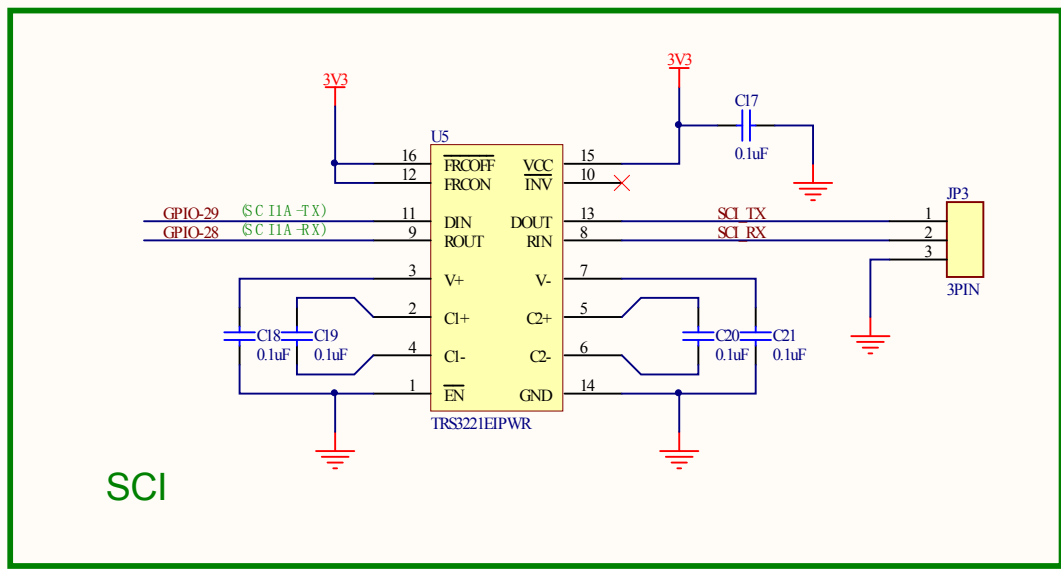
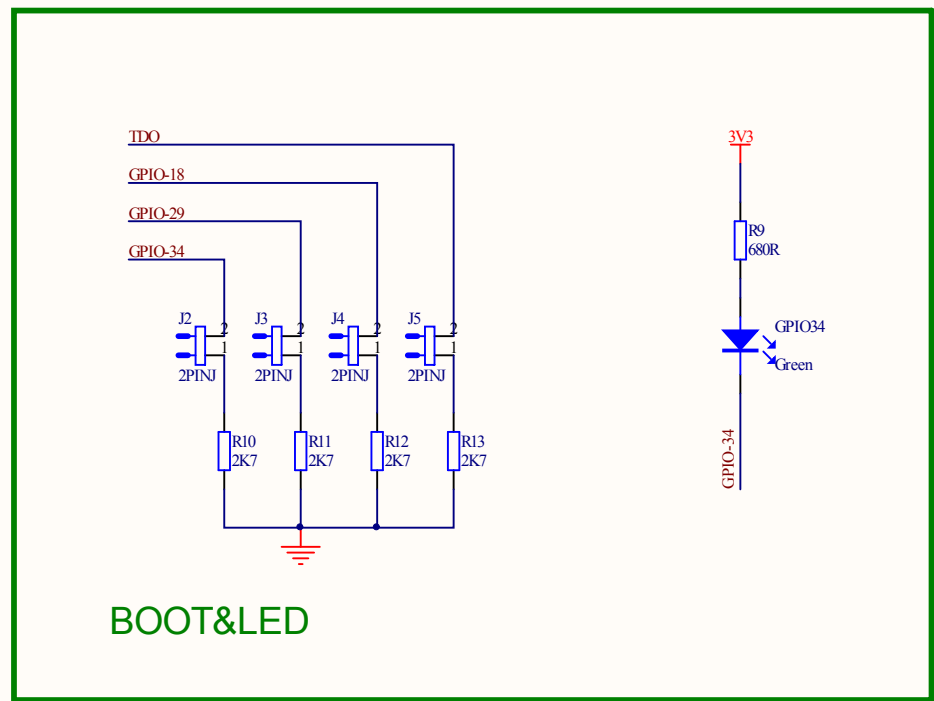
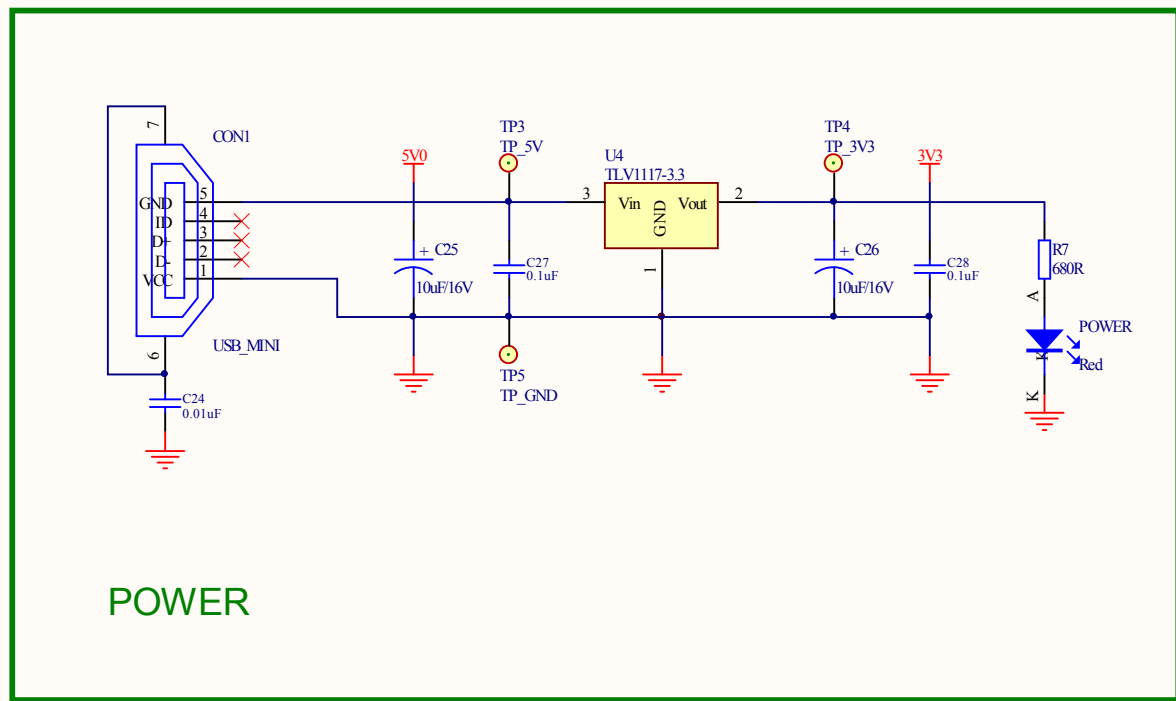
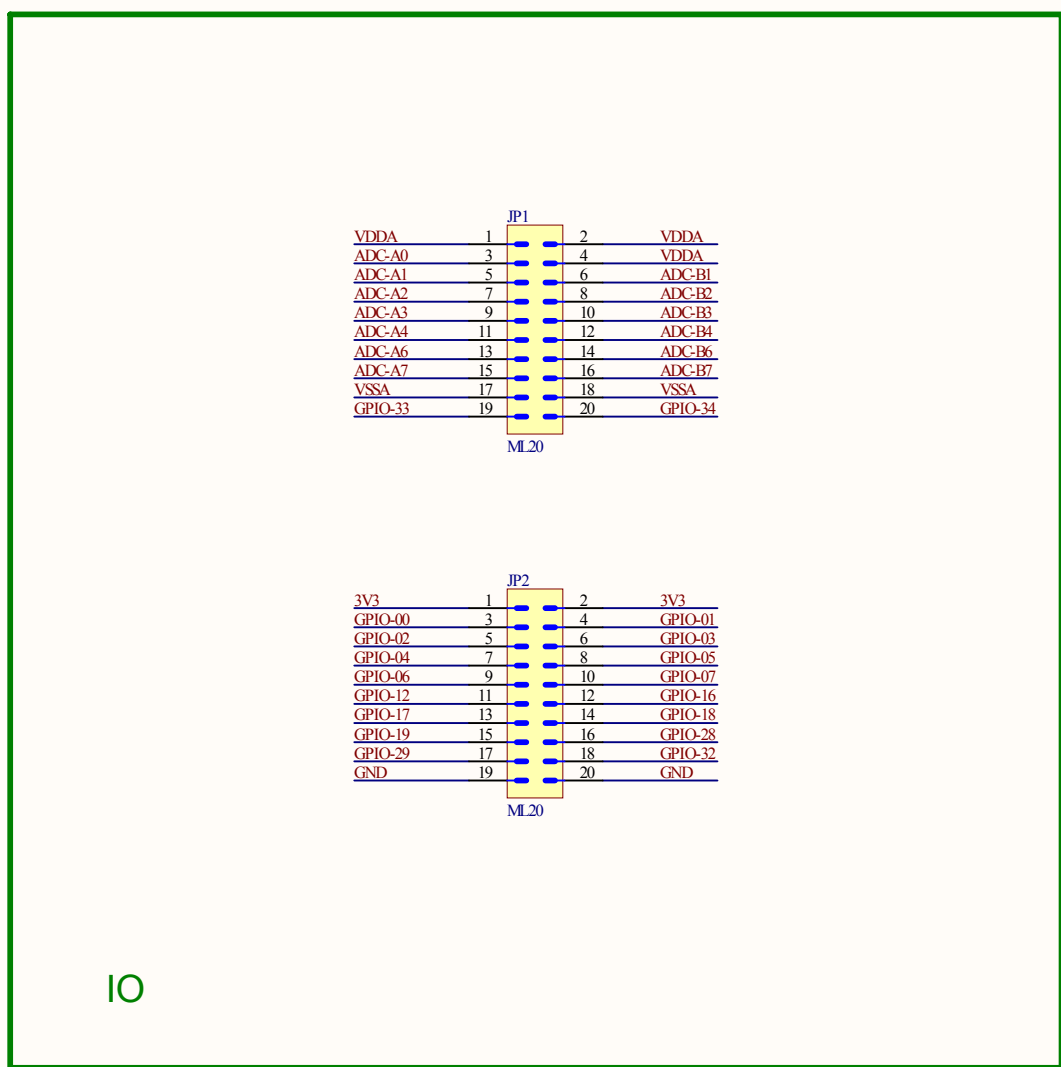
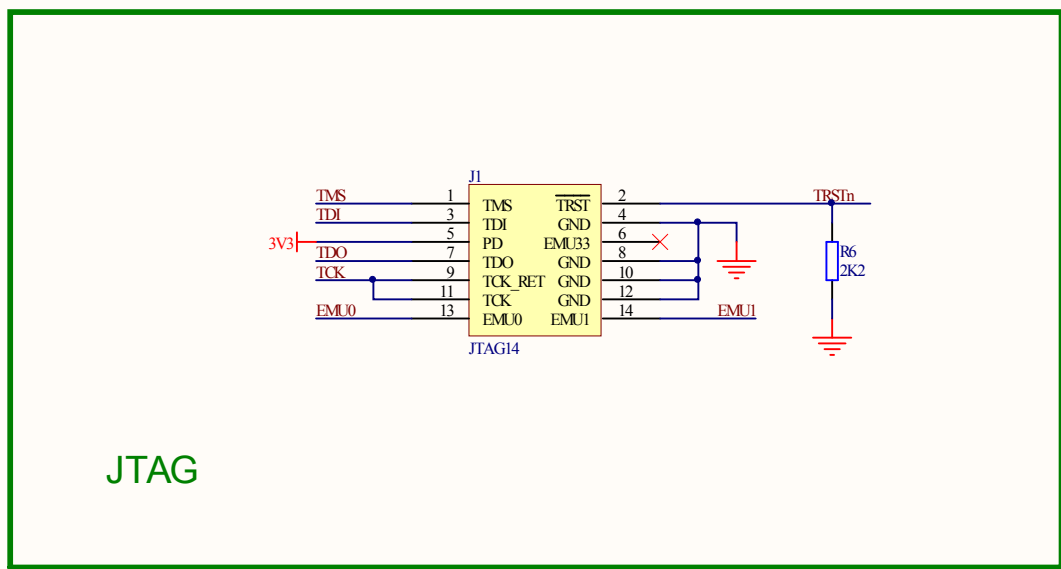
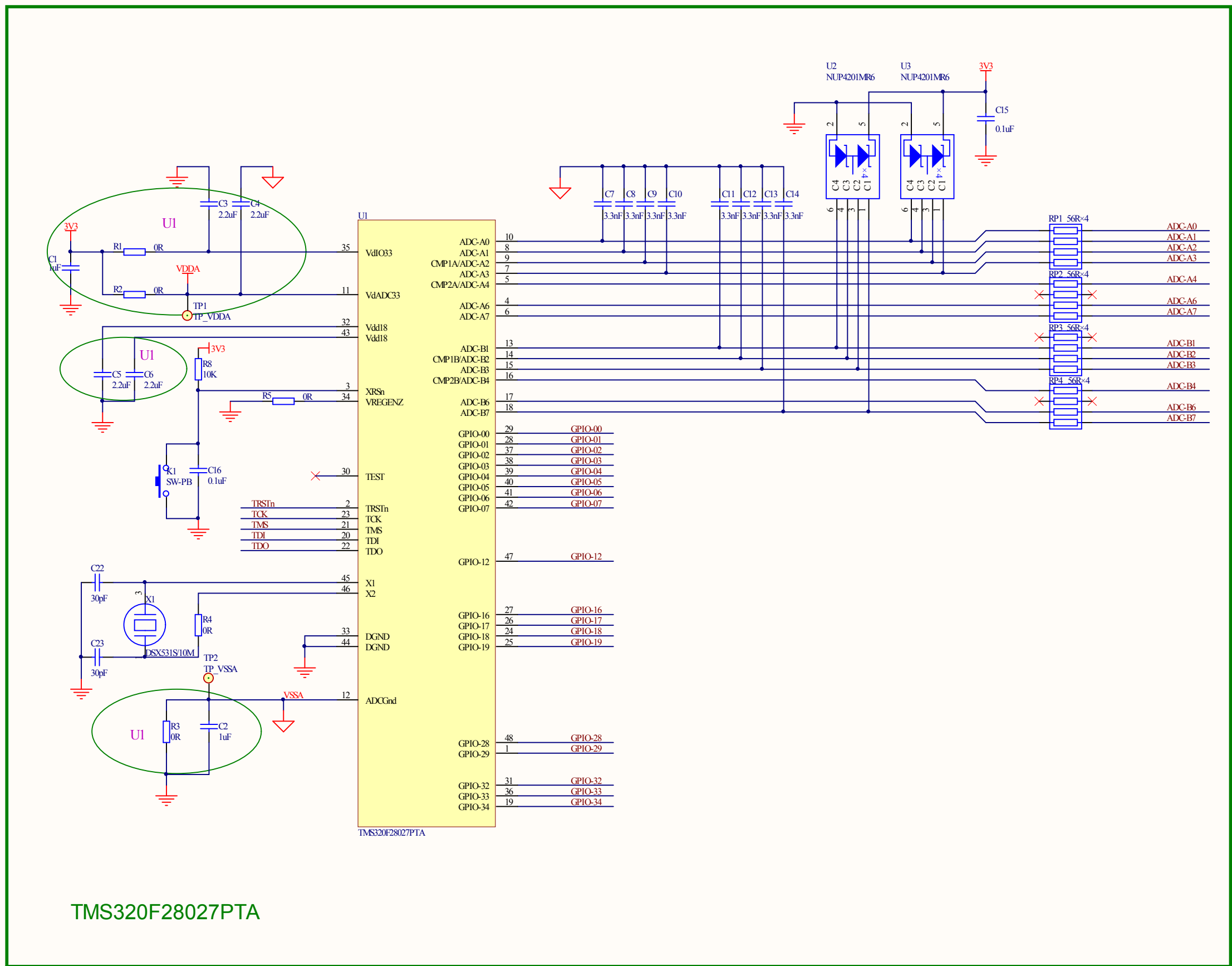


1: PICCOLO DMSI
2: L4970T28027DA_V1.0



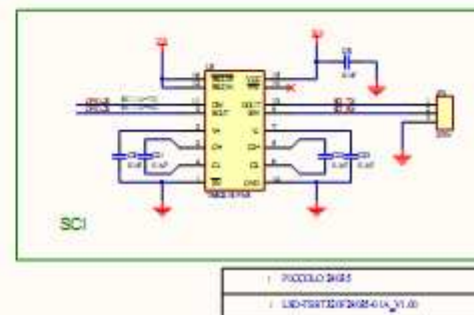
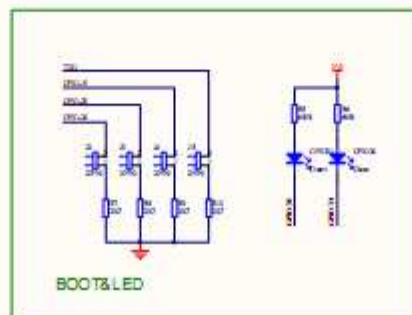
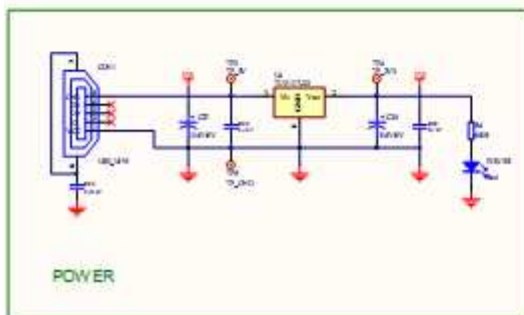
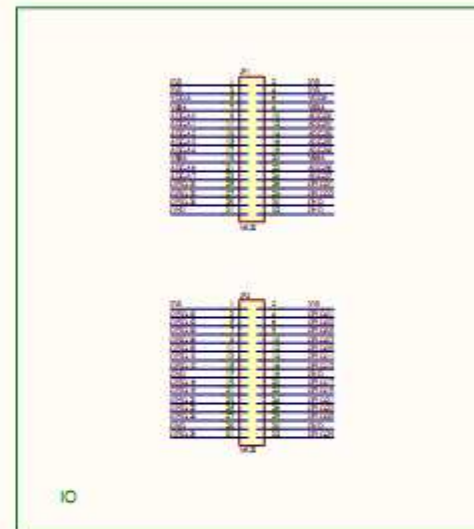
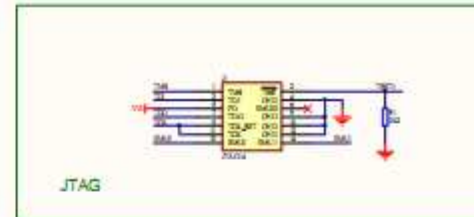
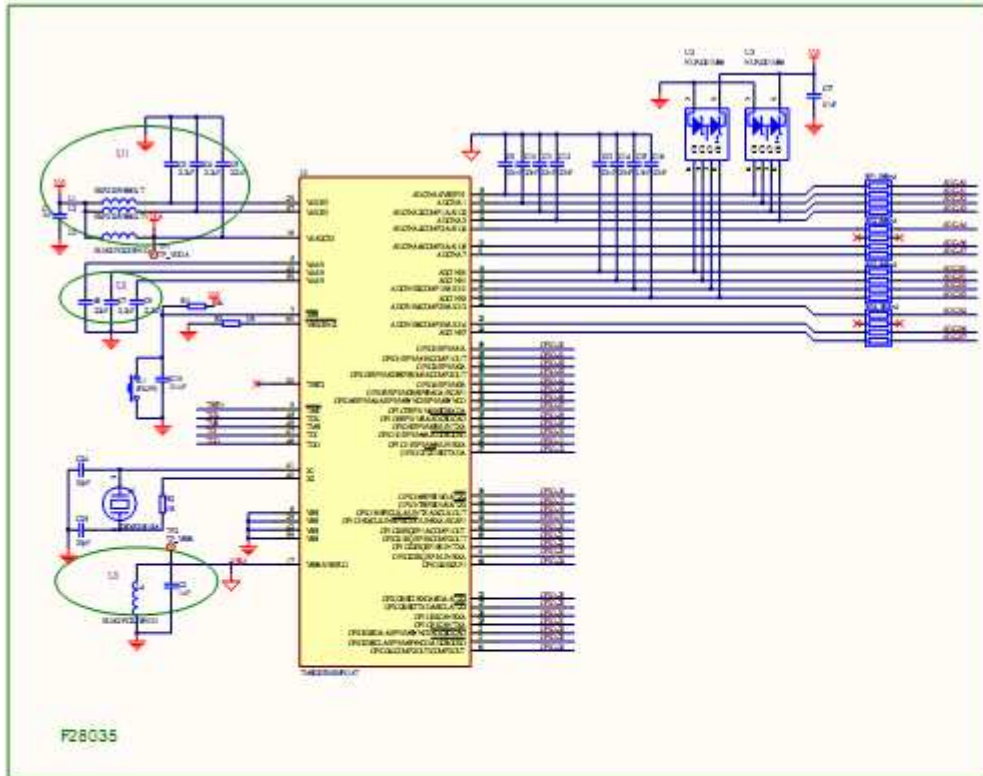
: PICCOLO 28035		
: LSD-TEST320F28035-01A_V1.00		
Rev: V1.00	: donggiao	Size: C
: 1	: 1	: :

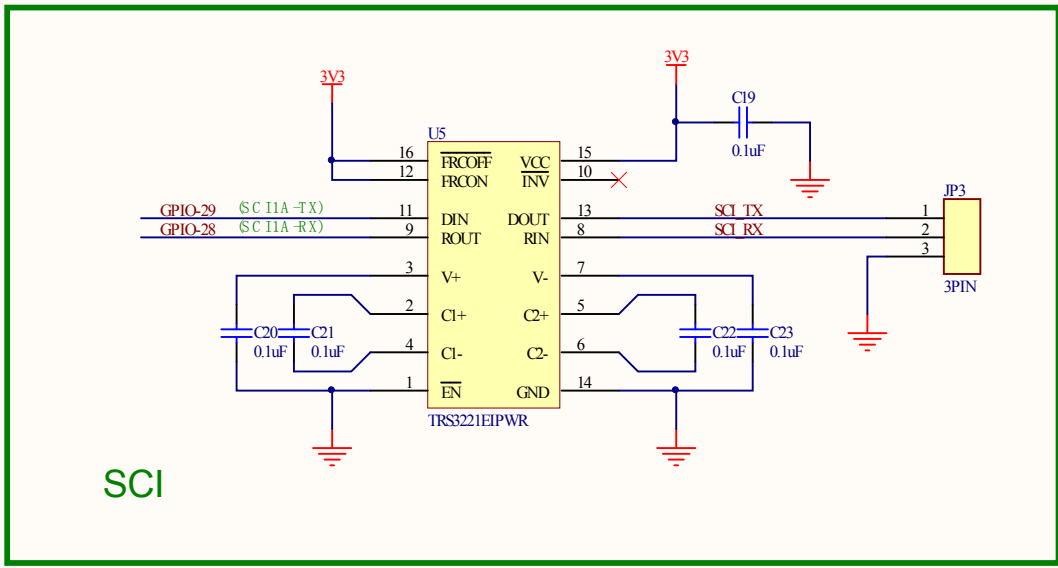
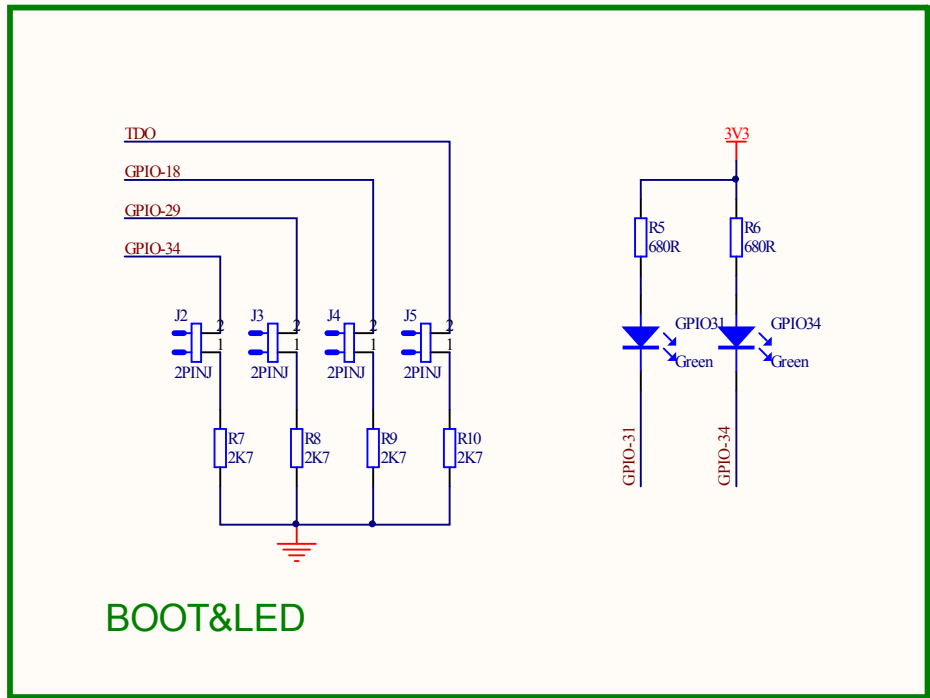
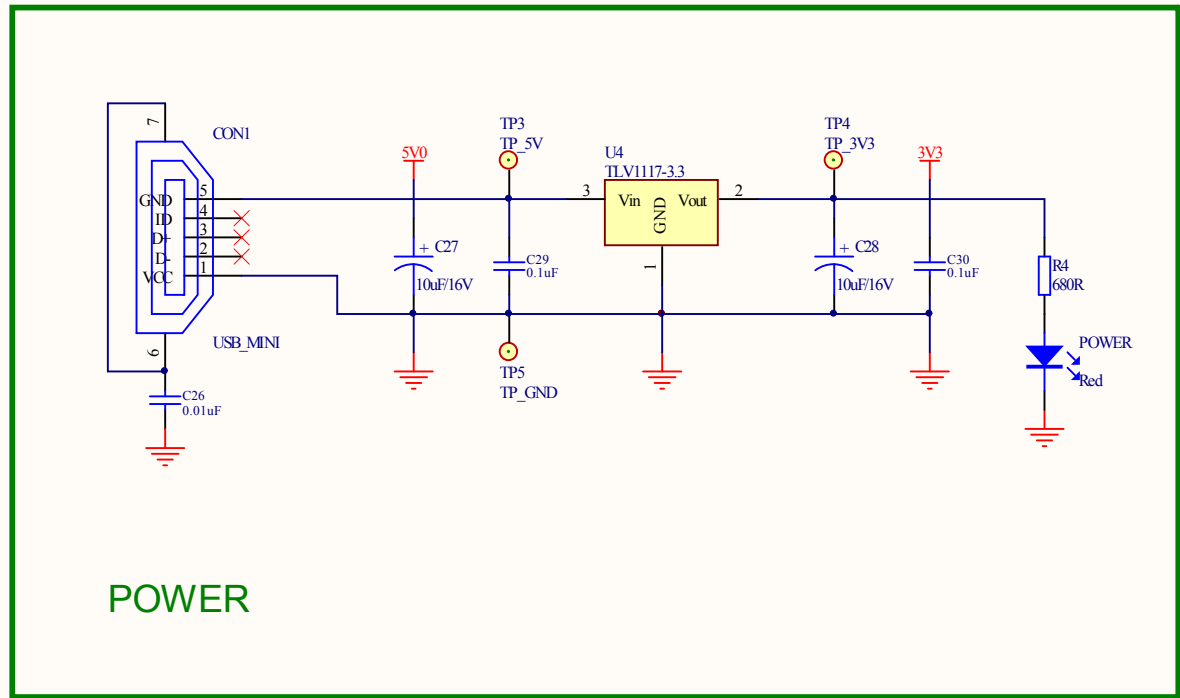
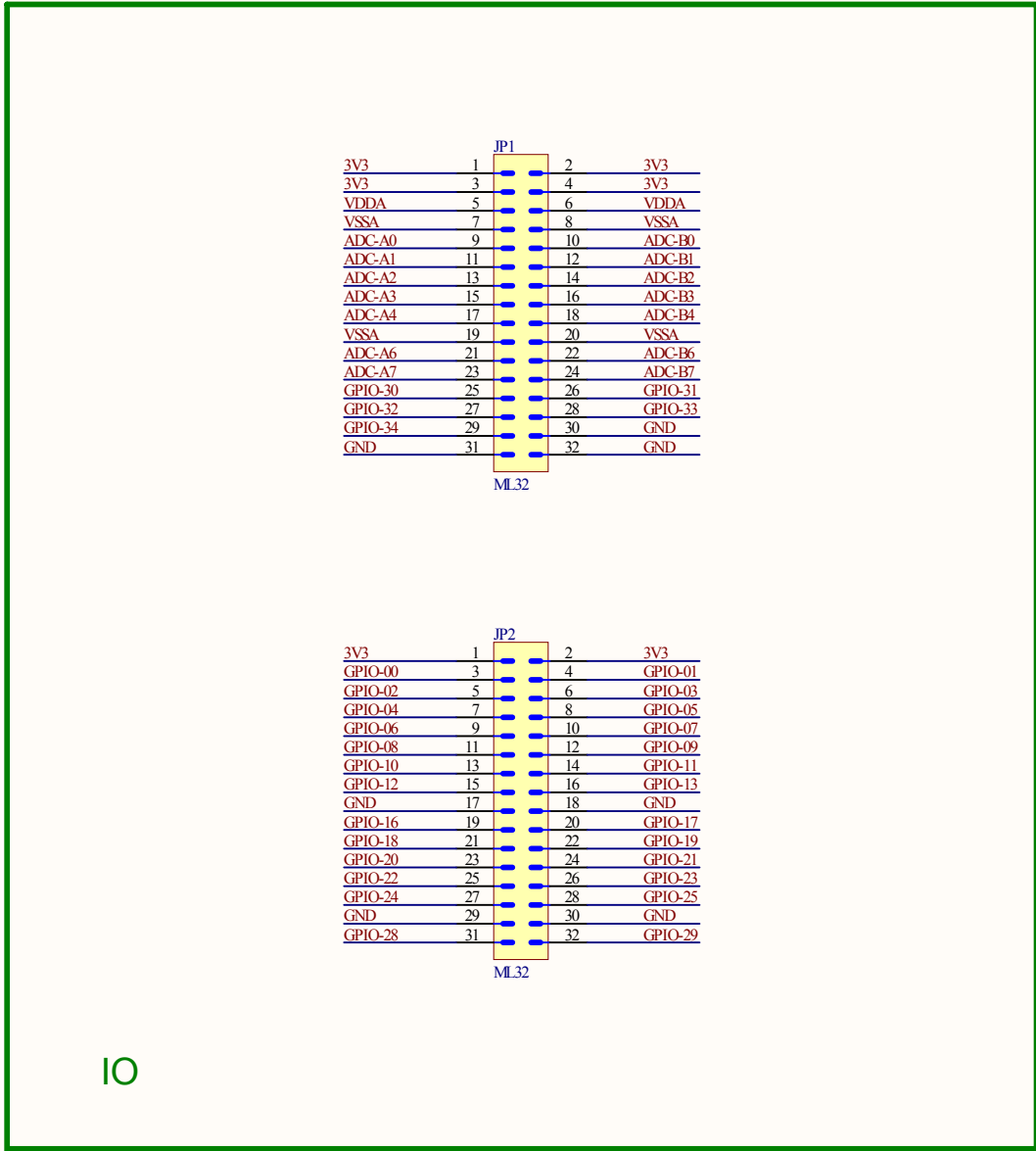
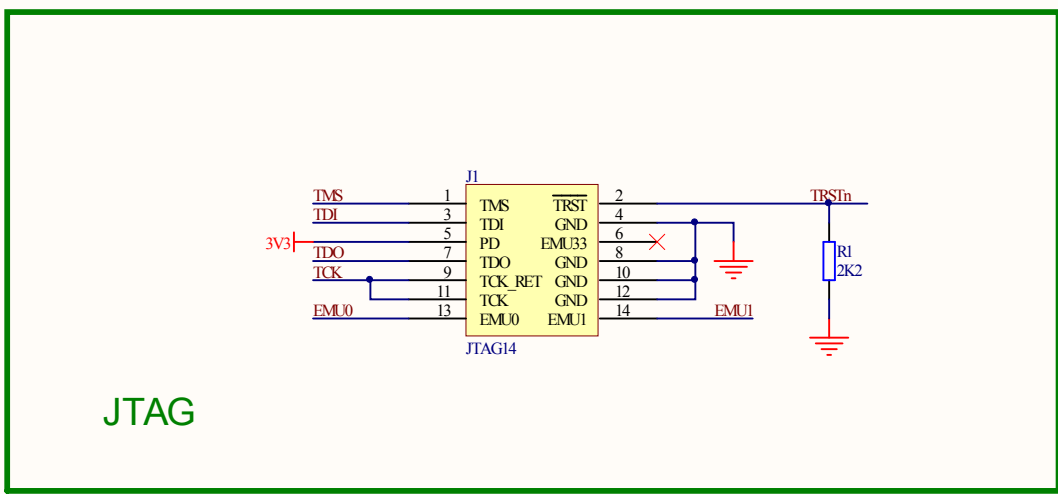
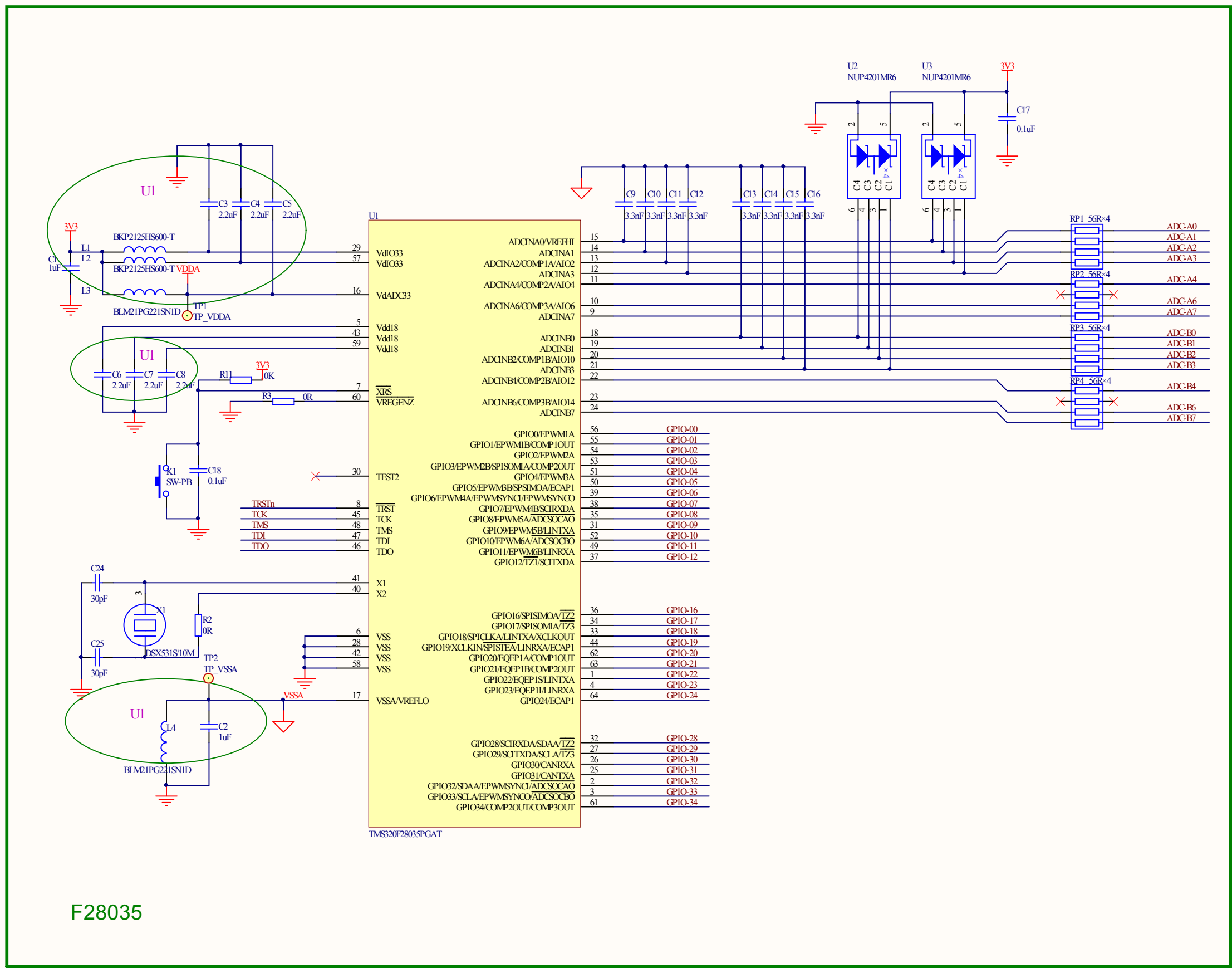




: PICCOLO 28035		
: LSD-TEST320F28035-01A_V1.00		
Rev: V1.00	: donggiao	Size: C
1	1	:

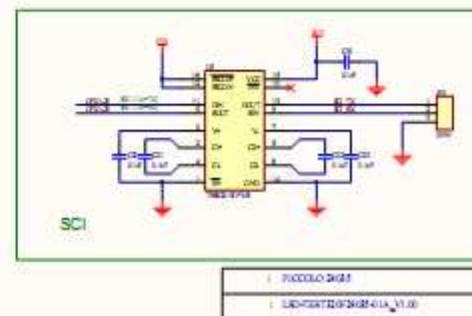
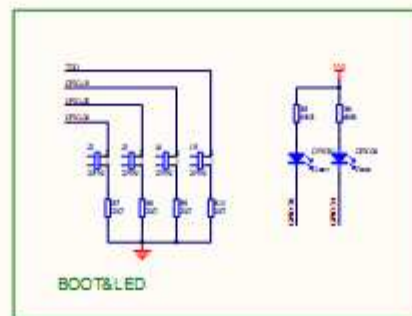
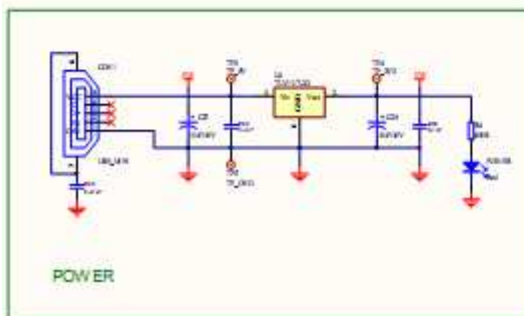
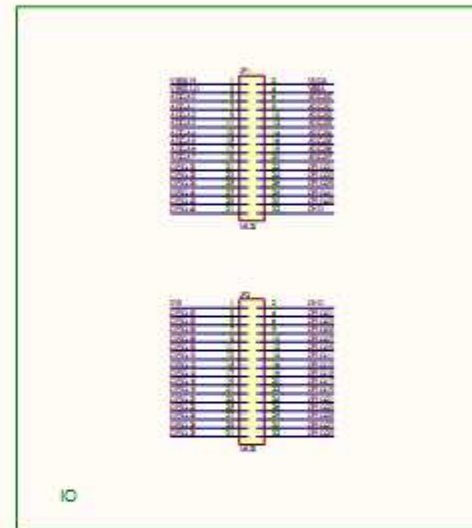
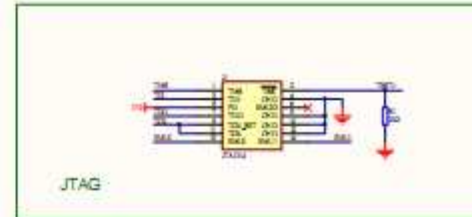
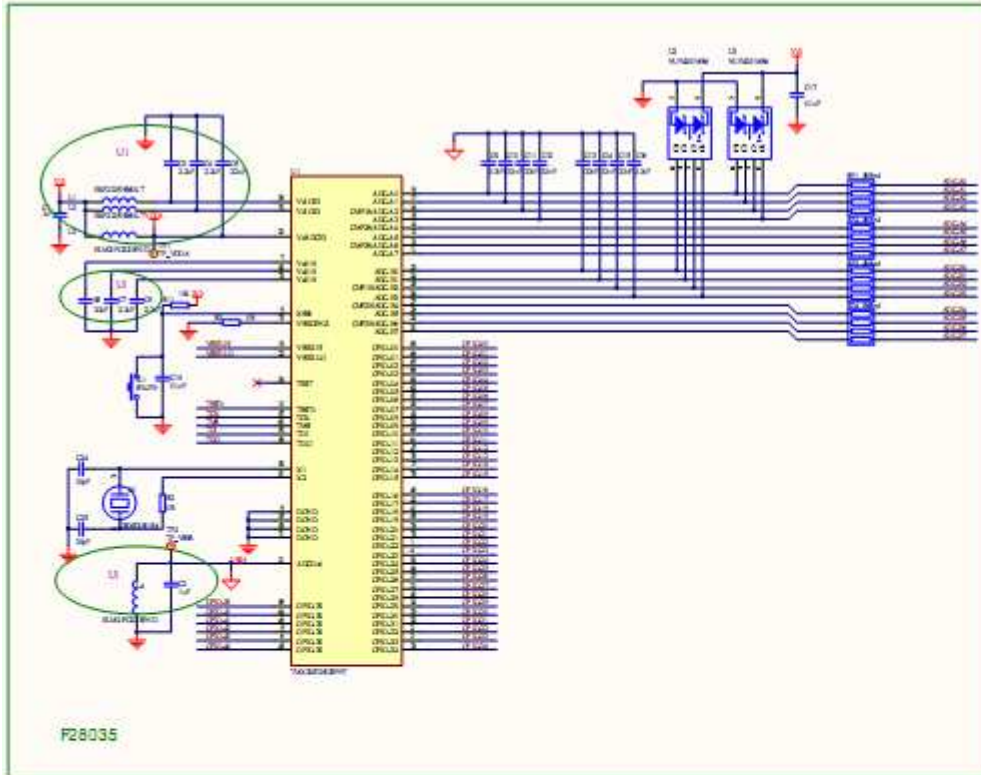
9.3 TMS320F28035PAG-64pin 电路图



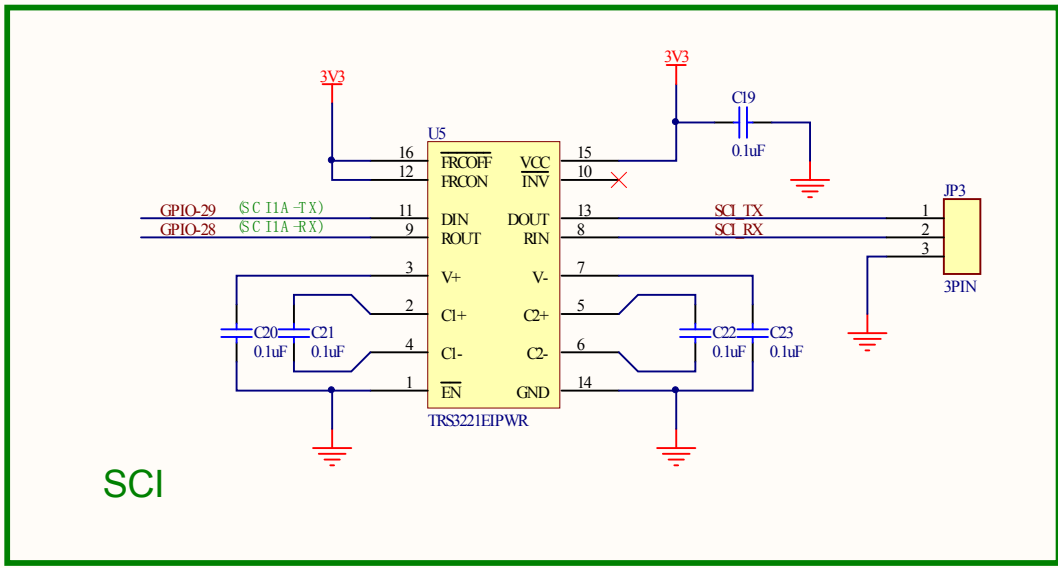
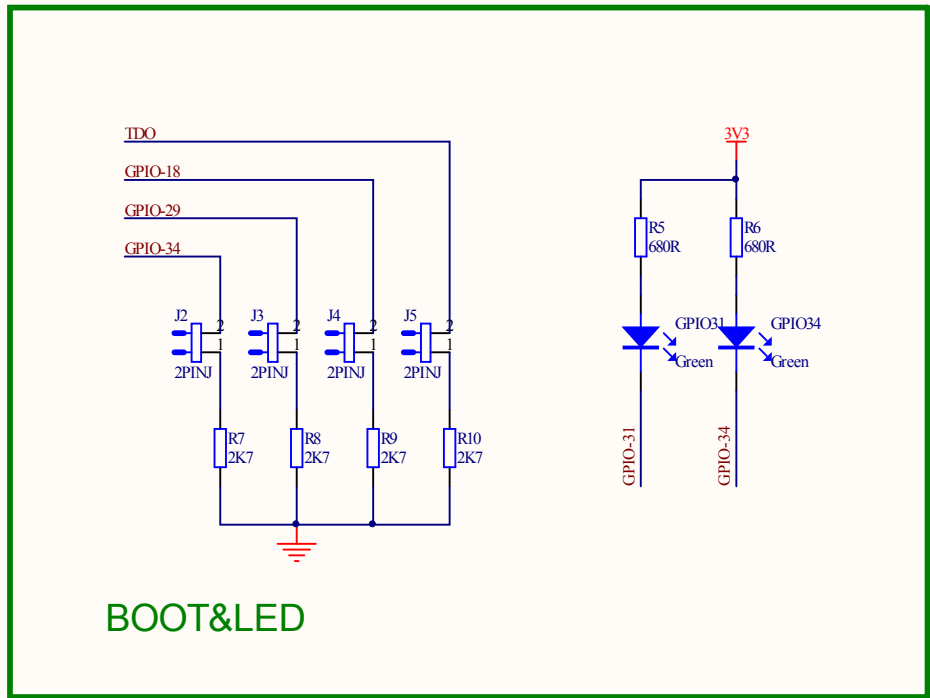
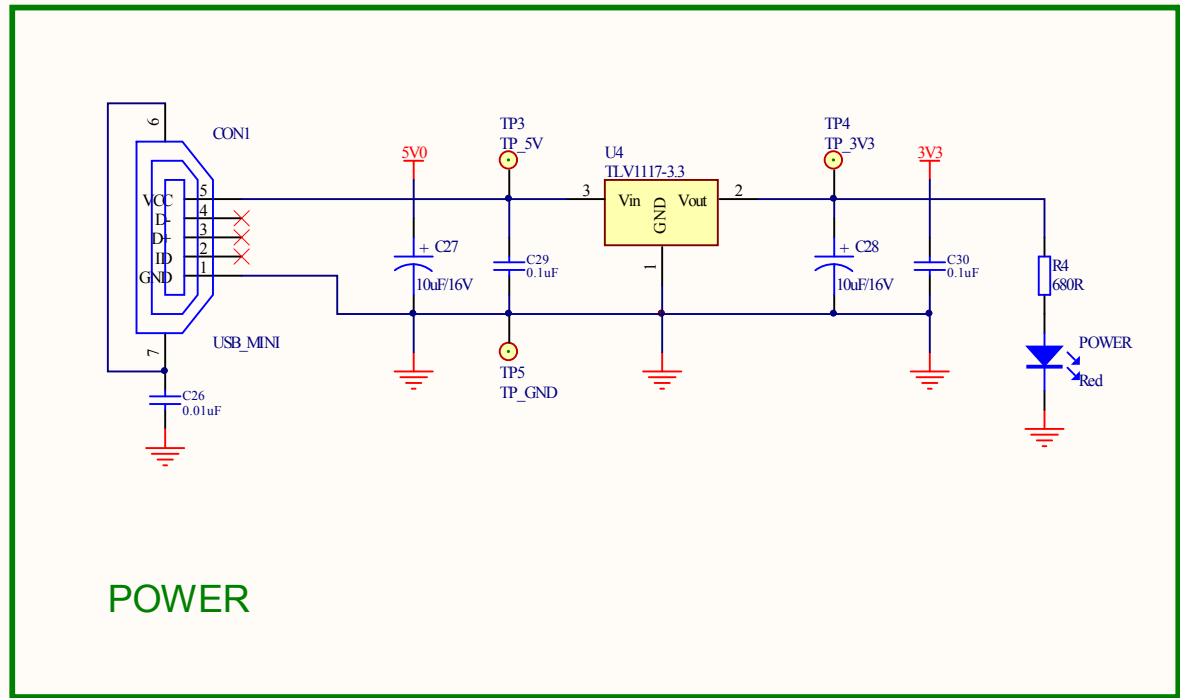
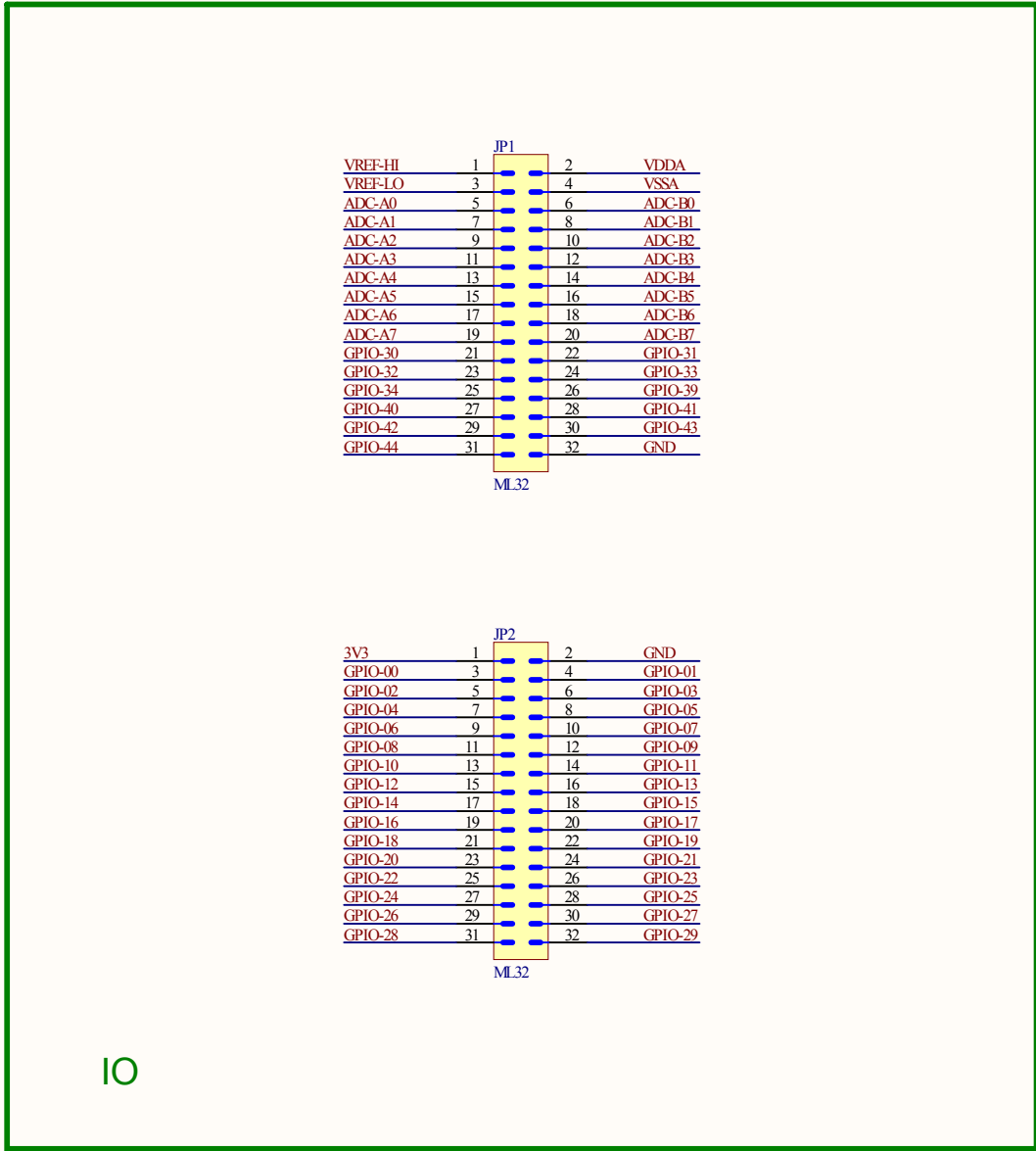
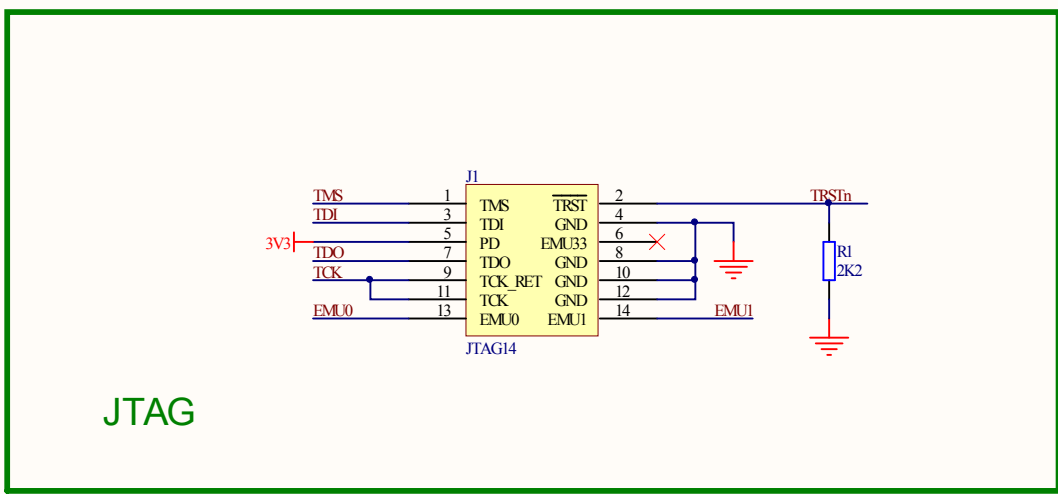
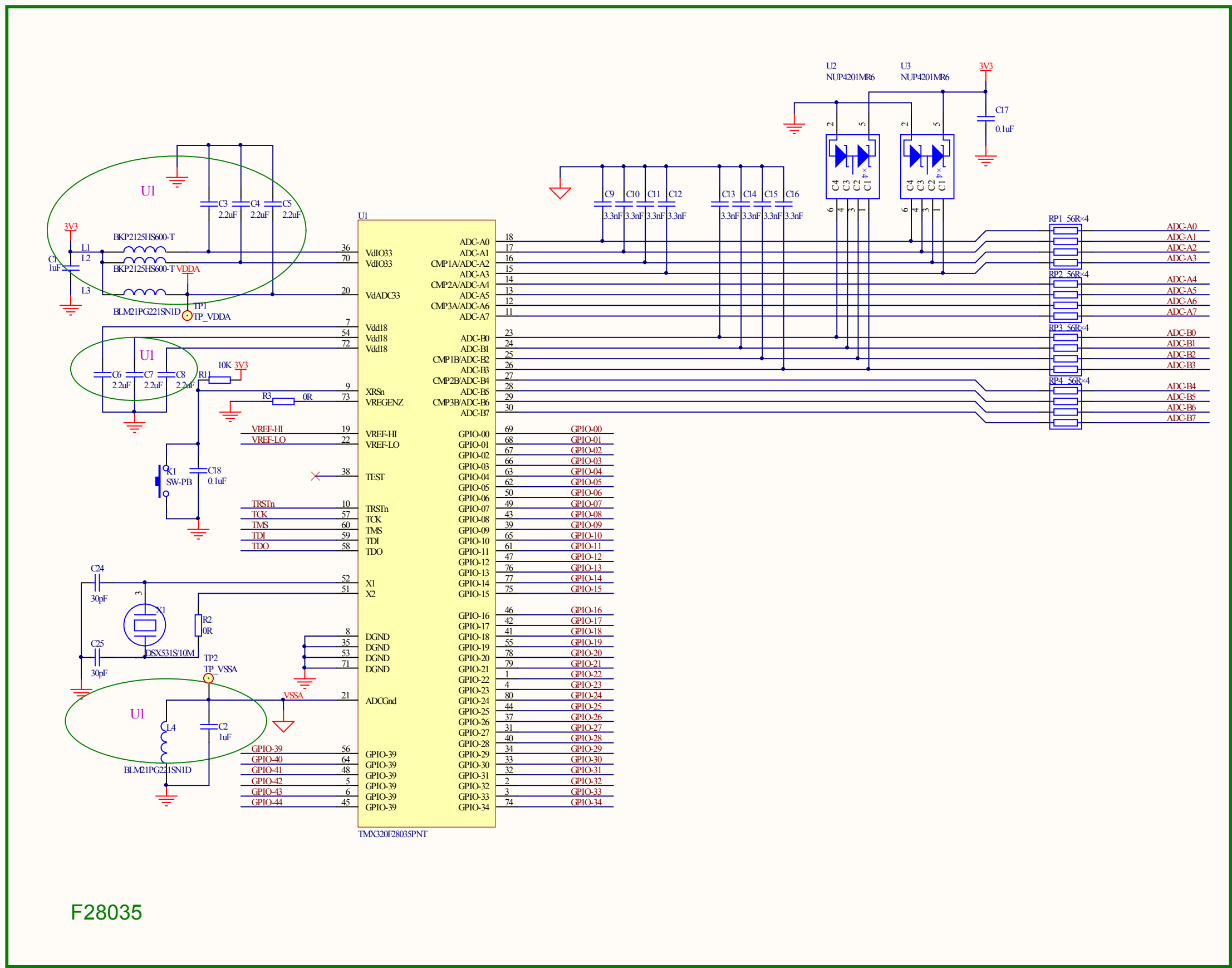


: PICCOLO 28035		
: LSD-TEST320F28035-01A_V1.00		
Rev: V1.00	: donggiao	Size: C
: 1	: 1	: :


9.4 TMS320F28035PN-80pin 电路图



1: PICCOLO 2421
1: LRD-G2720-202401A_V1.0



: PICCOLO 28035			
: LSD-TEST320F28035-01A_V1.00			
Rev: V1.00	: donggiao	Size: C	
1	1		



重要声明

德州仪器(TI) 及其下属子公司有权在不事先通知的情况下 随时对所提供的产品和服务进行更正、修改、增强、改进或其它更改 并有权随时中止提供任何产品和服务。客户在下订单前应获取最新的相关信息 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的硬件产品的性能符合TI 标准保修的适用规范。仅在TI 保证的范围内 且TI 认为有必要时才会使用测试或其它质量控制技术。除非政府做出了硬性规定 否则没有必要对每种产品的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险 客户应提供充分的设计与操作安全措施。

TI 不对任何TI 专利权、版权、屏蔽作品权或其它与使用了TI 产品或服务的组合设备、机器、流程相关的TI 知识产权中授予的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息 不能构成从TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可 或是TI的专利权或其它知识产权方面的许可。

对于TI 的产品手册或数据表 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。在复制信息的过程中对内容的篡改属于非法的、欺诈性商业行为。TI 对此类篡改过的文件不承担任何责任。

在转售TI 产品或服务时 如果存在对产品或服务参数的虚假陈述 则会失去相关TI 产品或服务的明示或暗示授权 且这是非法的、欺诈性商业行为。TI 对此类虚假陈述不承担任何责任。

TI 产品未获得用于关键的安全应用中的授权 例如生命支持应用(在该类应用中一旦TI 产品故障将预计造成重大的人员伤亡) 除非各方官员已经达成了专门管控此类使用的协议。购买者的购买行为即表示 他们具备有关其应用安全以及规章衍生所需的所有专业技术和知识 并且认可和同意 尽管任何应用相关信息或支持仍可能由TI 提供 但他们将独力负责满足在关键安全应用中使用其产品及TI 产品所需的所有法律、法规和安全相关要求。此外 购买者必须全额赔偿因在此类关键安全应用中使用TI 产品而对TI 及其代表造成的损失。

TI 产品并非设计或专门用于军事/航空应用 以及环境方面的产品 除非TI 特别注明该产品属于“军用”或“增强型塑料”产品。只有TI 指定的军用产品才满足军用规格。购买者认可并同意 对TI 未指定军用的产品进行军事方面的应用 风险由购买者单独承担 并且独力负责在此类相关使用中满足所有法律和法规要求。

TI 产品并非设计或专门用于汽车应用以及环境方面的产品 除非TI 特别注明该产品符合ISO/TS 16949 要求。购买者认可并同意 如果他们在汽车应用中使用任何未被指定的产品 TI 对未能满足应用所需要求不承担任何责任。

可访问以下URL 地址以获取有关其它TI 产品和应用解决方案的信息:

产品	应用
放大器	www.ti.com/audio
数据转换器	www.ti.com/automotive
DLP® 产品	www.ti.com/communications
DSP	www.ti.com/computers
时钟和计时器	www.ti.com/consumer-apps
接口	www.ti.com/energy
逻辑	www.ti.com/industrial
电源管理	www.ti.com/medical
微控制器	www.ti.com/security
RFID	www.ti.com/video
RF/IF 和 ZigBee® 解决方案	www.ti.com/wireless-apps
TI E2E 工程师社区	http://e2e.ti.com/cn/

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2010, Texas Instruments Incorporated