应用报告
ZHCA515–February 2013

多 DC/DC 转换和彩色 LED 控制，此转换和控制集成在 C2000™ 微控制器上

Daniel Chang

摘要

这份应用报告给出了一个使用 TMS320F2802x 微控制器来控制一个 LED 照明系统的解决方案。 此 Piccolo™ TMS320F2802x 系列器件是 C2000 微控制器的一部分，它可以实现 LED 照明系统的成本有效性设计。 借助于这些器件，有可能使用一个有效且极精确的方法来控制多个 LED 灯串。 除此之外，C2000 微控制器的速度也使得它能够集成很多附加任务以在一个正常系统中增加芯片数量和复杂性。 这些任务包括 DC/DC 转换级、支持功率因数校正 (PFC) 的 AC/DC 转换级、系统管理和多个诸如 DALI，DMX512 KNX 或者甚至电力线通信 (PLC) 的通信协议。 在本文档所描述的电路板上，8 个独立的 DC/DC 级（6 个升压，2 个单端初级电感转换器 (Sepic)）被用来驱动 8 个单独的 LED 灯串。

这份应用报告涵盖了以下内容：

- LED 照明技术的简要概括
- C2000 可为照明系统带来的优势
- 如何运行并熟悉多 DC/DC 彩色 LED 套件的 LED-ColorMix 项目

内容

1 LED 照明理论 2
2 C2000 在 LED 照明中的优势 6
3 系统概览 7
4 软件概述 8
5 硬体安装 12
6 软件设置 14
7 构建 1：DC/DC 级和 LED 灯串的开环路运行 17
8 构建 2：DC/DC 级和 LED 灯串的闭环路运行 21
9 对于进一步开发研究的思考 23
10 参考书目 23

图片列表

1 1931 CIE 色度图 3
2 LED 电流与 光通量间的关系 3
3 LED 正向电压与 电流间的关系 4
4 每个灯串的独立电源级 4
5 具有一个公共电源的 PWM 调光灯串 5
6 照明系统 6
7 多 DC/DC 彩色 LED 套件电路图 7
8 LED-ColorMix 项目文件 9
9 硬件特性 12
10 从指定位置安装 13
11 搜索最佳驱动程序 13

12 工作区启动器 14
13 创建一个目标配置 15
14 配置一个新目标 16
15 设定一个缺省目标配置 16
16 将多 DC/DC 彩色 LED 套件项目导入到您的工作区 17
17 构建级 1 方框图 18
18 为构建 1 配置观察窗口 20
19 构建级 2 方框图 21
20 为构建 2 配置观察窗口 22

图表列表

1 PWM 和 ADC 资源分配 8
2 主要变量 8
3 使用的宏 10
4 系统特性 11
5 每个递增系统构建中的测试模块 17

1 LED 照明理论

1.1 LED 的优势

作为一个相对较新和正在发展中的技术，在很多照明应用中，LED 已经成为一个有效的解决方案。 LED 为应用带来的一个主要优势就是其高效率。 如今的高亮度 LED 具有高达 60lm/W 的发光效率，不同的 LED 厂商宣称已经生产出发光效率大于 100lm/W 的 LED。 LED 优于其它光源的另外一个优势就是其较长的使用寿命，如果设计正确的话，大约为 5000 小时。 由此，在诸如街道照明的应用中，当将人力和服务缺失考虑在内时，灯泡更换会十分昂贵，LED 可被认为是在这些领域有一定的优势。 LED 还具有出色的振动弹性，从而提供双向照明并可实现几乎完全调光功能。 这些和很对其它特性可使得 LED 被视为照明技术的未来。

1.2 光源和 LED 特点

所有光源都有两个主要特点：光通量和色度。 光通量是视觉的一个属性，在这个属性中一个源看起来发光或反射光。 术语“亮度”常常被专门用来描述这个特性，然而，这个术语经常被用在生理学和非量化方法中。 一个更好的术语是光通量，其单位是流明，它是测得的光功率乘以 V-λ 定标函数。 这个函数用来补偿人眼对于不同波长的灵敏度。

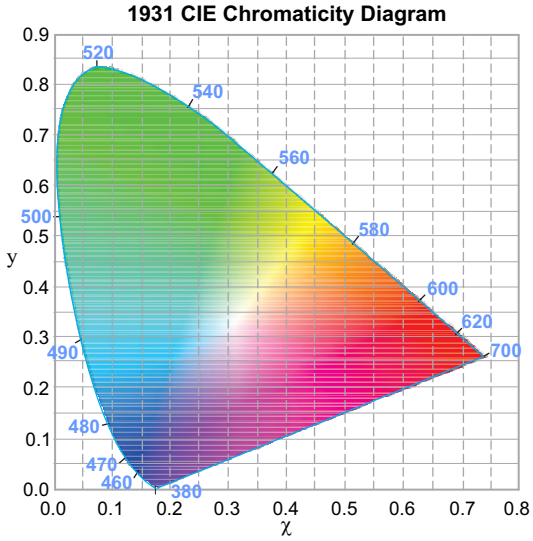


图 1. 1931 CIE 色度图

LED 可被看成一些由电流控制的器件。 光通量和色度在很大程度上由流经器件的电流控制。 在图 2中，可以看到，电流几乎与 LED 输出的光通量成比例。 由于这一事实，在很多系统中，编辑平均电流来调节 LED 或 LED 灯串的亮度。 另外唯一一个可以影响光通量和色度的主要变量是温度。 所以，控制一个 LED 系统中的温度变化很重要。

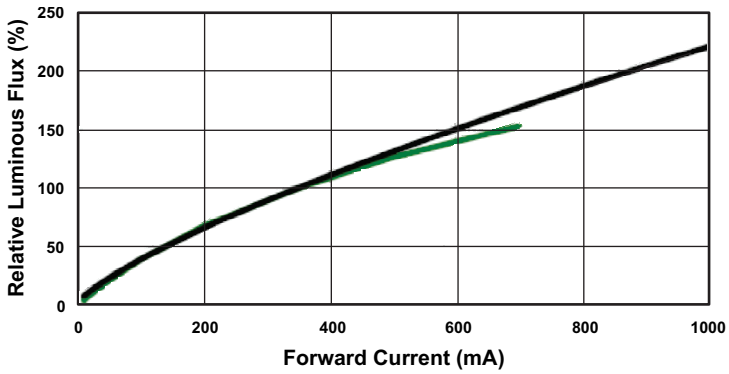


图 2. LED 电流与 光通量间的关系

图 3显示了一个 LED 中正向电压和电流间的关系。 请注意，因为在它开始导电前需要一个特定的阀值电压，所以 LED 的运行方式与一个二极管相类似。 一旦正向电压变得大于阀值电压，电流将成指数增长，直到它达到 LED 器件的最大电流技术规格。 对于一个 6 LED 灯串，为了点亮 LED，将有必要使得正向电压比 LED 阀值电压的 8 倍还要大。

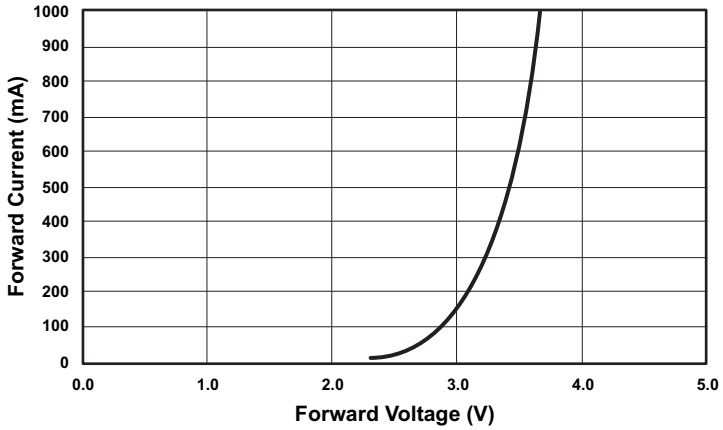


图 **3. LED** 正向电压与 电流间的关系

由于 LED 电压和电流之间的关系，看上去似乎能够控制 LED 电压来指定 LED 电流，因此进而指定 LED 灯串给出的光通量。 问题是，即使当温度轻微上升时，上面曲线图的形状保持不变，但是会向左移动。 例如，如果在 3.0V 电压上控制一个 LED（使用图 3中的技术规格），那么最初从 LED 汲取的电流为 150mA。 然而，随着 LED 温度上升，曲线图向左移动，电流将轻微上升。 然后，这个增加的电流将使温度上升。 这个循环将持续下去，直到 LED 器件发生故障。 因此，优先对 LED 灯串的电流进行控制。 它也将有助于显示 LED 系统中热管理的重要性。

1.3 控制技巧

有多种技巧可用来控制一个 LED 灯串的电流。 在很多情况下，一个简单的解决方案就已经够用了，但是对于很多灯串中 LED 的数量过多或者所需的总光通量有可能很大的应用，情况就并非如此了。 为了保持效率，将有可能需要使用一个开关模式电源。图 4显示了控制其中一个系统的方法。

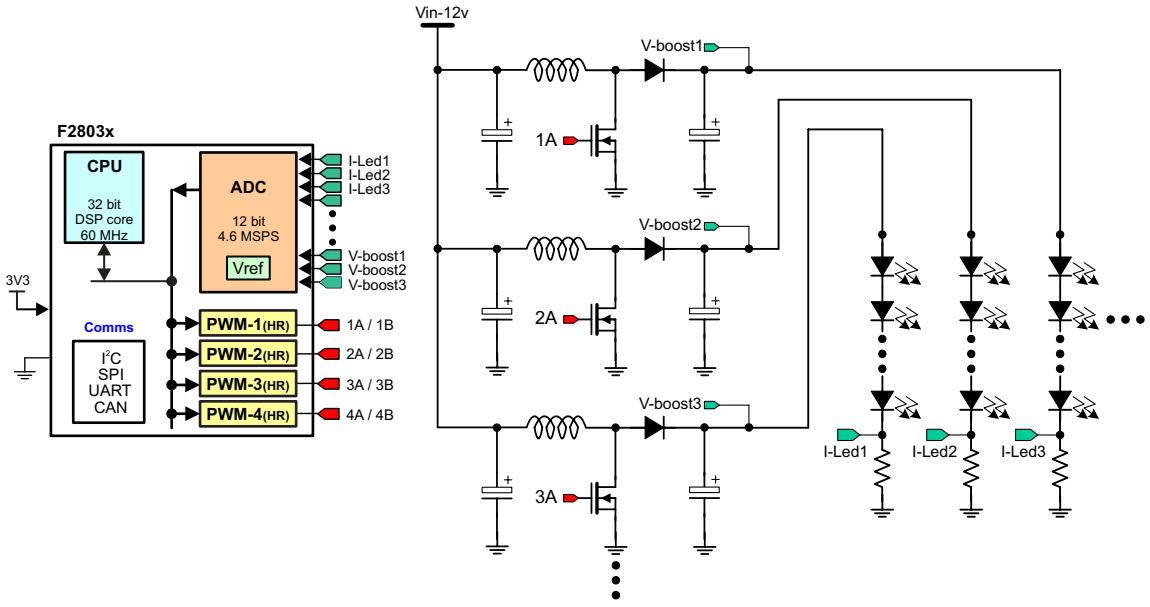


图 **4.** 每个灯串的独立电源级

在上面的解决方案中，每个灯串由一个电流控制升压级供电。 通过一个与 LED 灯串串联的电阻器来采样每个灯串的电流，Piccolo 器件提供反馈环路。 然后将这个被采集的电流与控制器内的一个基准相比较，而这有助于确定被发送到每个独立 DC/DC 升压级中 MOSFET 的下一个占空比的值。 为了保持所需的恒定电流，场效应晶体管 (FET) 的脉宽调制 (PWM) 频率必须相对较大以避免闪烁。

Piccolo 器件上提供的高性能外设能够控制上述系统并仍旧具有大量的带宽来提供系统管理。 此 Piccolo 外设，也就是片载比较器和 ADC 采样的可配置性，这使得 MCU 可分别通过峰值或平均电流模式来控制电流。

下页显示了一个不同的技巧。 一个单 DC/DC 级被用来提供将出现在 LED 灯串上的电压。 然后，这个电压将对应于一个将流经一个 LED 灯串的单电流。 然后，每个单独的灯串由一个 MOSFET 打开和关闭，这样通过一个 LED 灯串发出的平均功率将被减少。 在这种方法中，每个单独的 LED 灯串可被调暗至一个基准平均电流。

在 DC/DC LED 照明套件上，已经执行了后一种方法。 单 DC/DC 级是一个 SEPIC 转换器，而 Piccolo MCU 也可控制过达 8 个 LED 灯串。

在每个策略中，可完成灯串交错以减少峰值电流并提升总体效率。 Piccolo 的 PWM 可实现单独 PWM 模块之间和/或与一个外部同步脉冲的同步。

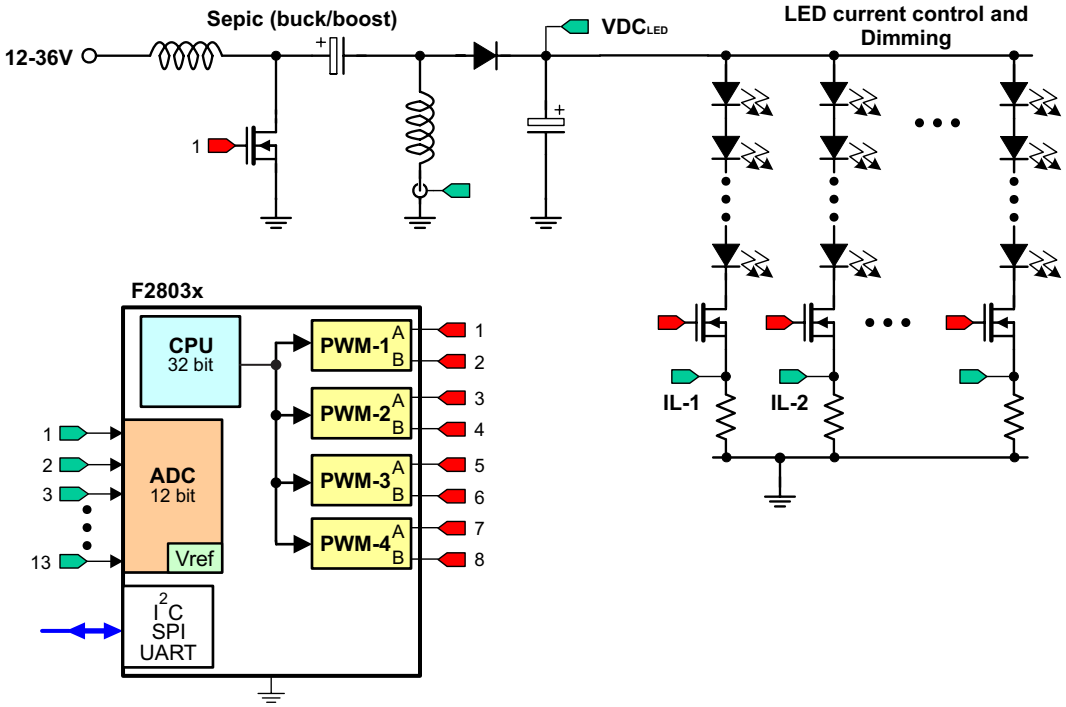


图 5. 具有一个公共电源的 PWM 调光灯串

1.4 一个有效的 LED 系统

一个 LED 系统必须具有功率转换级以将正确电压和电流传递给 LED 灯串。 如图 6中所显示的一个典型系统将在 PFC 升压电路前具有一个 AC/DC 整流器，然后一个或更多并联 DC/DC 级将被用来驱动一个或多个 LED 灯串。 为了创建一个高效系统，每个功率级必须被设计成高效运行并且对这些功率级的控制也必须是有效的。

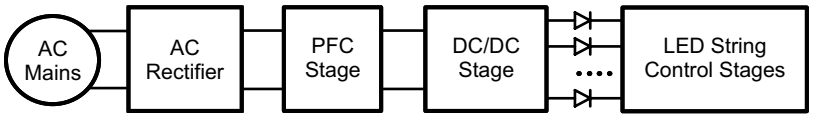


图 6. 照明系统

在开发一个智能、有效的系统中，通信也起到了很大的作用。 一个中心区域，或者一个基于手机的应用可控制住所或办公室的照明而无需单独控制灯串。 很多照明标准已经存在，诸如 DALI，KNX 和 DMX512。 这些标准与其它标准一起作为双绞线、无线或电力线通信 (PLC) 解决方案存在。

随着诸如 C2000 Piccolo 系列产品和 MSP430 等全新且廉价 MCU 的出现，通过改进功率级并使用大多数先进的 LED 即可实现高效系统。 在了解周围环境或由一个网络主机控制的系统中使用的智能照明也可在提升效率和减少维护成本中发挥很大作用。 单独的镇流器可使用光传感器来感测环境光照并只生成区域内所需的光照强度。 一个 MCU 可以计算一个 LED 的使用寿命并对随时间降级的光输出进行补偿，并且在它达到预先的使用寿命时警告外部系统，在警告中此镇流器可以开始显示故障符号。

借助于 C2000 Piccolo 系列产品的高性能和软件灵活性，可在一个单芯片上完成灯串控制、功率级控制、通信和智能照明控制。 很明显，处理器带宽在一定程度上限制了可以进行的操作，但是在很多系统中，LED 灯串控制，DC/DC 级和某些系统控制使用的带宽少于 C2000 上 CPU 带宽的 50%。

2 C2000 在 LED 照明中的优势

C2000 系列器件具有所需的计算能力来执行复杂的控制算法以及功率级控制和 LED 灯串控制中所需的正确外设组合。 这些外设具有符合安全要求且执行系统所需的全部，诸如用于 PWM 的片载比较器和针对 PWM 的触发区。 与这些外设一起，C2000 软件（库和应用软件）和硬件（应用套件）系统有助于减少开发照明解决方案所需的时间和精力。

一个 C2000 照明解决方案能够提供以下优势：

- 灯串间的精确电流匹配
- 准确的颜色匹配
- 大范围调光能力；有可能大于 20000:1
- 可执行 PWM 或恒定电流调光
- 一个灯串中 LED 的数量不受器件的限制。 如果在一个特定的产品中需要更多数量的 LED，只需检查 MOSFET 和 MOSFET 驱动器，而软件更改应该很少。
- 基于 LED 使用、老化、感测到的外部亮度等的自适应调光。
- 通过 CAP 或 QEP 外设可轻松实现与视频时钟的同步
- 由 C2000 外设集功能和灵活性所实现的 PFC，AC/DC，DC/DC 和更多操作的高效控制。 电源控制是市面上 C2000 的主要强项。
- 诸如内部集成电路 (I2C)，串行外设接口 (SPI) 和通用异步接收器/发送器 (UART) 的广泛通信协议。
- 高性能 CPU 使得 C2000 器件能够在一个芯片上完成多个任务，诸如多个灯串的单独灯串控制，DC/DC 控制，AC/DC 控制和通信。
- 由于此器件通过软件设定，针对多个区域和多个配置来制造产品通常只需要很少的软件改动。

SPRABR7 — http://www-s.ti.com/sc/techlit/SPRABR7

3 系统概览

3.1 硬件预览

多 DC/DC 彩色 LED 套机被设计成在与所包含的 LED 控制面板一同使用时可接受 12V - 18V DC 电压（额定值）。 由于 LED 的亮度大体上与电流成比例，8 个 DC/DC 功率级中每一个功率级的输出电流可被单独调节至一个所需的水平以获得所需 LED 灯串的亮度和颜色。 对于这个应用，每个灯串的输出电流被设定在 0-50mA 的范围内。 为了执行独立的 LED 灯串控制，每个 LED 灯串被连接至一个单独的 DC/DC 功率级。每个 DC/DC 功率级包含一个 MOSFET 并且这个 MOSFET 的接通时间控制流经 LED 灯串的平均电流。图 7显示了出现在多 DC/DC 彩色 LED 套件上的硬件。

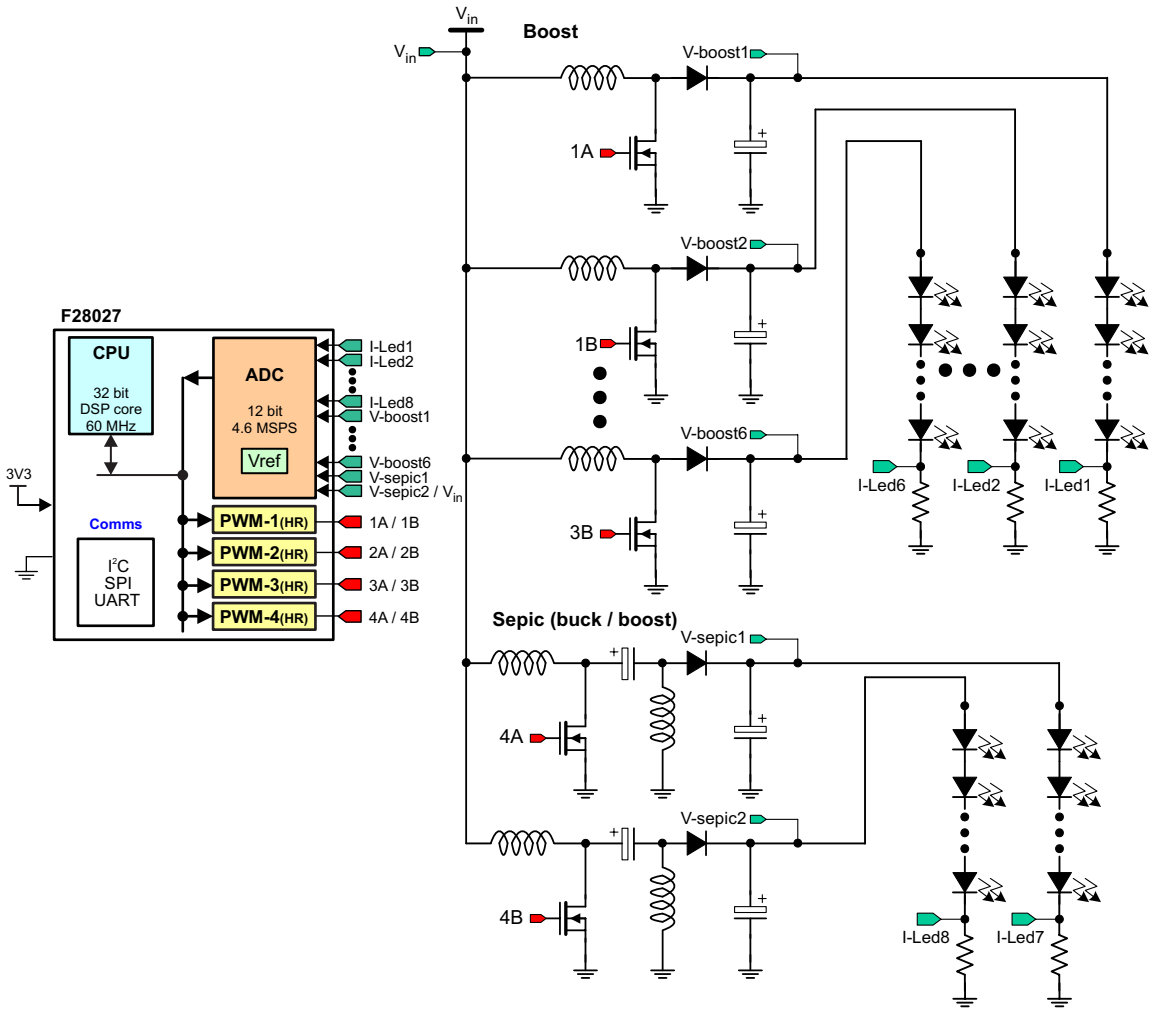


图 7. 多 DC/DC 彩色 LED 套件电路图

表 1中显示了 F2807 微控制器和多 DC/DC 彩色 LED 套件电路板之间关键信号连接。

表 1. PWM 和 ADC 资源分配

宏名称			信号名称	PWM/ADC 通道	说明
主电路板	[Main]	VDCin-meas	VDCin-meas	ADC-B4	与 V-Led-4B 复用的输入电压感测
双升压	[M2]	PWM-1	PWM-1A	PWM-1A	升压 1 PWM 信号
		PWM-2	PWM-1B	PWM-1B	升压 2 PWM 信号
		Vfb-1	V-Led-1A	ADC-B1	升压 1 输出电压感测
		Vfb-2	V-Led-1B	ADC-B2	升压 2 输出电压感测
		Ifb-1	I-Led-1A	ADC-A2	升压 1 输出电流感测
		Ifb-2	I-Led-1B	ADC-A0	升压 2 输出电流感测
双升压	[M3]	PWM-1	PWM-2A	PWM-2A	升压 3 PWM 信号
		PWM-2	PWM-2B	PWM-2B	升压 4 PWM 信号
		Vfb-1	V-Led-2A	ADC-B3	升压 3 输出电压感测
		Vfb-2	V-Led-2B		升压 4 输出电压感测
		Ifb-1	I-Led-2A	ADC-A4	升压 3 输出电流感测
		Ifb-2	I-Led-2B	ADC-A1	升压 4 输出电流感测
双升压	[M4]	PWM-1	PWM-3A	PWM-3A	升压 5 PWM 信号
		PWM-2	PWM-3B	PWM-3B	升压 6 PWM 信号
		Vfb-1	V-Led-3A	ADC-B5	升压 5 输出电压感测
		Vfb-2	V-Led-3B		升压 6 输出电压感测
		Ifb-1	I-Led-3A	ADC-A6	升压 5 输出电流感测
		Ifb-2	I-Led-3B	ADC-A3	升压 6 输出电流感测
双 Sepic	[M5]	PWM-1	PWM-4A	PWM-4A	Sepic 1 PWM 信号
		PWM-2	PWM-4B	PWM-4B	Sepic 2 PWM 信号
		Vfb-1	V-Led-4A	ADC-B7	Sepic 1 输出电压感测
		Vfb-2	V-Led-4B	ADC-B4	与 VDCin-meas 复用的 Sepic 输出电压感测
		Ifb-1	I-Led-4A	ADC-B6	Sepic 1 输出电流感测
		Ifb-2	I-Led-4B	ADC-A7	Sepic 2 输出电流感测

4 软件概述

4.1 构建选项

为了使您能够逐渐构建并理解此项目，此项目被 LED-ColorMix-Main.c 和 LED-ColorMix-ISR.asm 文件中的 #if options 分成了不同的构建。 所使用的构建由 LED-ColorMix-Settings.h 中的变量 INCR_BUILD 设定。下面是 LED-ColorMix 项目中不同可用构建的简要说明。

- 构建 1：开环测试，检查 DC/DC 级和 LED 灯串的基本运行
- 构建 2：DC/DC 级和 LED 灯串的闭环路运行

表 2. 主要变量

Gui_Vin	DC 输入电压 (0V - 36V)，只用于仪器仪表
Gui_Vout{X}	针对 DC/DC 级 {X} 的输出电压 (0V - 50V)，只用于仪器仪表
Gui_Iout{X}	每个 LED 灯串的输出电流 (0A - 0.650A)，只用于仪器仪表
Gui_Iset{X} （只适用于构建 2）	设定针对 LED 灯串{X} 的闭环电流 (0 - 0.100A)
Pgain{X} （只适用于构建 2）	针对 LED 灯串 {X} 的成比例增益；调整值：0 ~ 1000
Igain{X} （只适用于构建 2）	针对 LED 灯串 {X} 的积分增益；调整值：0 ~ 1000

表 2. 主要变量 (continued)

Dgain{X} （只适用于构建 2）	针对 LED 灯串 {X} 的微分增益：调整值：0 ~ 1000
SlewStep{X} （只适用于构建 2）	这个值确定了 LED 电流 {X} 斜升至基准电流的速度
SlewStepAll （只适用于构建 2）	当基准电流被改变时，这个值确定所有 LED 斜升至基准电流的速度
Duty{X} （只适用于构建 1）	设定针对 LED 灯串 {X} 的 DC/DC 级的开环占空比
ChannelEnable{X} （只适用于构建 2）	启用针对 LED 灯串 {X} 的输出
EnableAll （只适用于构建 2）	启用所有 LED 灯串的输出
StopAll （只适用于构建 2）	禁用所有 LED 灯串的输出
MergeLEDs （只适用于构建 2）	• 0：LED 由 Gui_Iset{X} 单独控制 • 1：LED 成对控制；Gui-Iset1 控制 1 和 4，Gui-Iset2 控制 2 和 5，Gui-Iset3 控制 3 和 6，Gui-Iset7 控制 7 和 8（按照颜色配对）
OCtrip （只适用于构建 2）	表示缓过流跳变是否被触发并禁用所有 LED 灯串的输出
OC_limit （只适用于构建 2）	这个值确定触发缓过流保护所需的最小输出电流

4.2 此项目中的关键文件

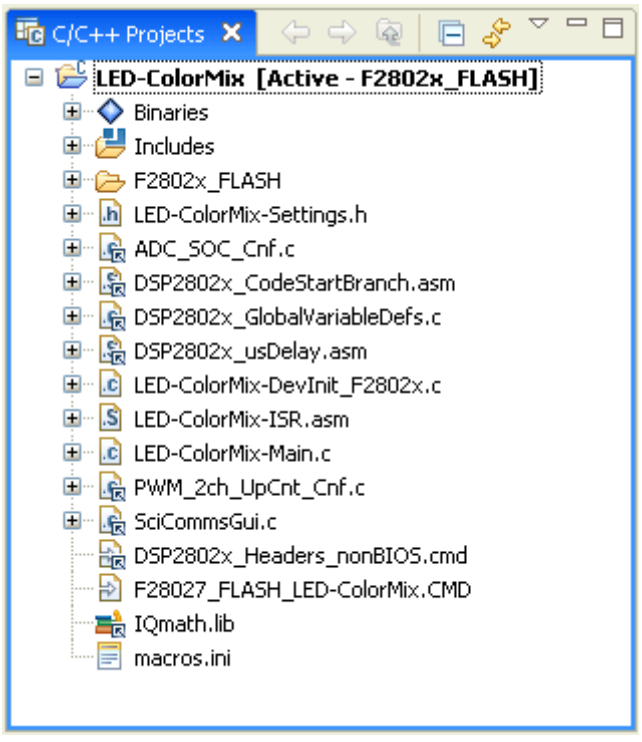


图 8. LED-ColorMix 项目文件

本项目中使用的关键框架文件为：

- LED-ColorMix-Main.c - 这个文件被用来初始化、运行和管理此应用。 这个文件是支持应用的“大脑”。
- LED-ColorMix-DevInit_F2802x.c - 这个文件负责器件（在这个例子中此器件为 F28027）的初始化和配置，并包含设置时钟，锁相环 (PLL)，通用输入输出 (GPIO) 等的函数。
- LED-ColorMix-ISR.asm - 这个文件包含时序关键“控制类型”代码。 这个文件有一个初始化部分，此部分由 C 语言可调用的汇编程序 subroutine _DPL_Init 执行一次。 这个文件还包含 _DPL_ISR 例程，此例程的执行速度由 PWM 或者用来触发它的定时器确定。
- LED-ColorMix-Settings.h - 这个文件被用来设定项目的全局定义（也就是构建选项）。 请注意，它被链接至 LED-ColorMix-Main.c 和 LED-ColorMix-ISR.asm 两个文件。
- LED-ColorMix-Calculations.xls - 这是一个数据表文件，此文件计算将 MCU 使用的 Q 值数转换为实际值时所使用的不同比例因数。 变量 K-Iout 和 iK_Iset 是比例因数的示例。

4.3 使用的宏

为了减少每次开发代码所花费的精力，使用了一个采用宏区块的方法。 这些宏区块被写入 C 语言可调用汇编程序并且可具有一个 C 语言和汇编语言元素。表 3提供了一个本项目中所使用的宏的列表。

表 3. 使用的宏

C 语言配置函数	**ASM** 初始化宏	**ASM** 运行时间宏
PWM_2ch_UpCnt_Cnf	PWMDRV_2ch_UpCnt_INIT n	PWMDRV_2ch_UpCnt n
无	CNTL_2P2Z_INIT n	CNTL_2P2Z n
ADC_SOC_Cnf	ADCDRV_1ch_INIT n	ADCDRV_1ch n

由于外设的配置由 C 语言中完成，此配置的修改十分方便。 ASM 驱动程序宏为实时运行提供所需的紧凑代码。 下面列出了每个宏以更好地理解它们在此项目的作用。

- **PWM_2ch_UpCnt_Cnf()**

  在 PWM_2ch_UpCnt_Cnf.c 中定义，这个函数按照自变量指定的那样来配置 PWM 通道以驱动功率级。与自变量如何传递以及传递什么自变量有关的细节请参见源文件。 这是一个宏，并且这个宏是 TI PowerLib 的一部分。

- **PWMDRV_2ch_UpCnt n**

  在 PWMDRV_2ch_UpCnt.asm 定义，这个宏被用来按照 CNF 文件配置的那样来更新中断处理例程 (ISR) 中的 PWM[n]A 和 PWM[n]B。 这是一个宏，并且这个宏是 TI PowerLib 的一部分。

- **CNTL_2P2Z n**

  在 CNTL_2P2Z.asm 中定义，这个宏是一个使用无限脉冲响应 (IIR) 滤波器构建实现的二阶补偿器。 这个函数所需的 5 个系数在 C 语言后台循环中被声明为一个长数组。 这个函数与任何外设无关，因此就不需要 CNF 函数调用。 这是一个宏，并且这个宏是 TI PowerLib 的一部分。

- **ADC_SOC_Cnf()**

  在 ADC_SOC_Cnf.c 中定义，这个函数按照自变量中指定的那样配置 ADC 外设。 与自变量如何传递以及传递什么自变量有关的细节请参见源文件。 这是一个宏，并且这个宏是 TI PowerLib 的一部分。

- **ADCDRV_1ch n**

  在 ADCDRV_1ch. 中定义，这个宏对 ADC 模块的用法进行了摘要。 此驱动程序宏将 ADCRegister 的结果复制到一个 NetBus 数组变量中，此结果可被用来连接不同的库文件宏。 这是一个宏，并且这个宏是 TI PowerLib 的一部分。

LED-ColorMix 项目具有以下属性：

LED-ColorMix 项目的 RAM 存储器用量		
构建级	程序存储器用量 2802x	数据存储器用量 2802x (1)
构建 2（闪存）	594 个字	949 个字

(1) 不包括堆栈尺寸

LED-ColorMix 项目构建 2 的 CPU 初始化	
模块名称 (1)	周期数量
运行环境保存、恢复等	22
ADCDRV_1ch (1)4	27
时间分片开销	11
LED 控制（每个 ISR 更新四个灯串）	
ADCDRV_1ch (1) 4	27
CNTL_2P2Z (1) 4	192
PWMDRV_2ch_UpCnt (1) 2	31
总的周期数量	310
CPU 初始化 @ 60Mhz	68.7% (2)

(1) 此模块已经在头文件中被定义为“宏”。
(2) 在 133kHz ISR 频率下。

表 4. 系统特性

系统特性	
开发和仿真	Code Composer Studio™ v4.1（或更高版本），支持实时调试
目标控制器	TMS320F2802x
PWM 频率	400kHz PWM
PWM 模式	在 A 和 B 模块上具有独立输出的上数模式
中断	每个第三时基周期时间上的 EPWM1 - 执行 133kHz ISR 执行速率

5 硬体安装

在这本指南中，每个组件的名称以它们的宏编号开始，然后是参考名称。 例如，[M2]-J1 是指宏 M2 内的跳线 J1，而 [Main]-J1 是指位于其它已定义的宏区块外电路板上的 J1。图 9显示了多 DC/DC 彩色 LED 电路不能的某些主要连接器和特性。

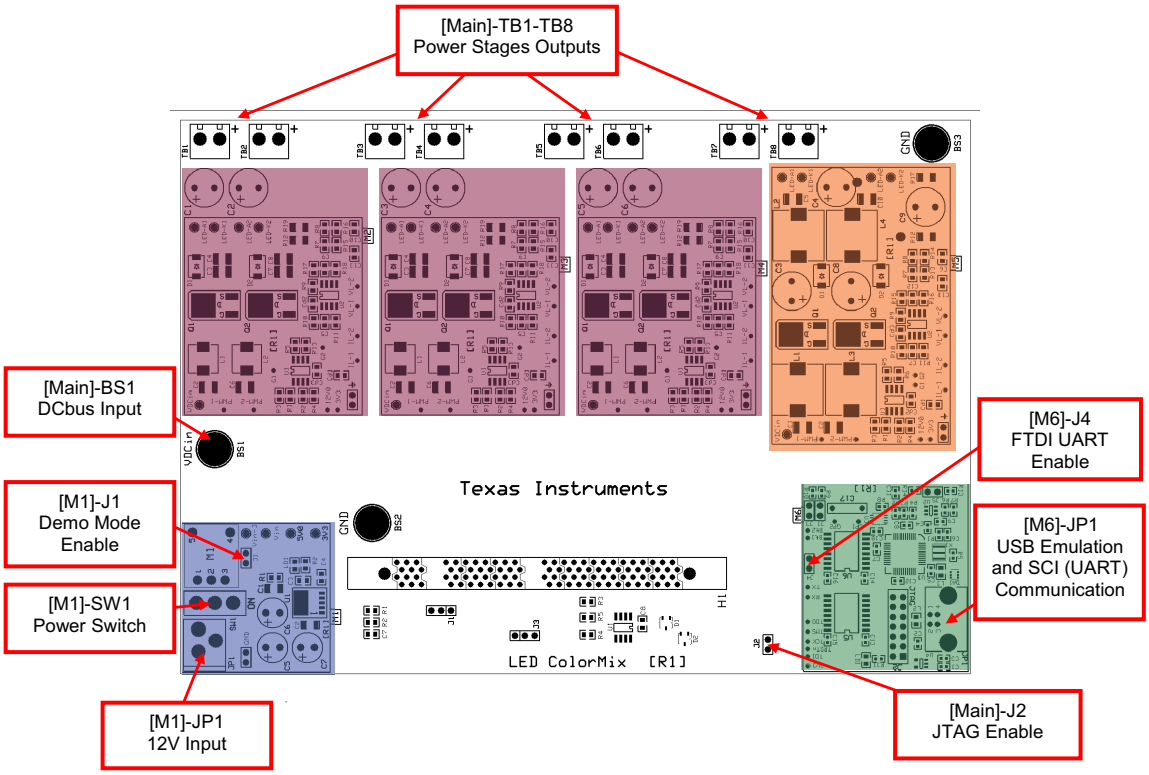


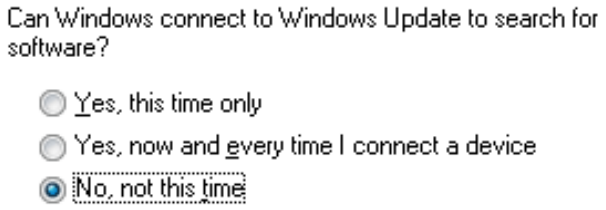
图 **9.** 硬件特性

[Main] - controlCARD 插槽、电源连接器和跳线
[M1] - 12V，5V 和 3.3V DC 电源轨
[M6] - 通过 USB 的已隔离的 JTAG 和串行通信接口 (SCI)（通用异步接收器/发送器 (UART)）通信
[M2]，[M3]，[M4] - 两个独立的升压级，per macro instance
[M5] - 两个独立的 sepic 级

1. 将一个 F28027 控制卡插入到插槽 [Main]-H1 中。
2. 使用一条 USB 线缆将您的计算机连接到电路板。 靠近 USB 连接器 [M6]-JP1 的 [M6]-LD1 应该被打开。

注: 如果从未安装过 Code Composer Studio，也许需要安装驱动程序以使电路板正常工作。 如果当 USB 线缆被连接在电路板和计算机之间时出现一个弹出窗口，使用安装向导从包含在此套件中的 XDS100v1 USB 驱动目录中安装驱动程序。

3. 当 Windows 询问搜索 Windows 更新时，请选择“No, not at this time”（不，这次不）并单击 Next（下一步）。



4. 在下一个屏幕显示上选择“Install from specific location”（从指定位置安装）（请见图 10）并单击 Next。

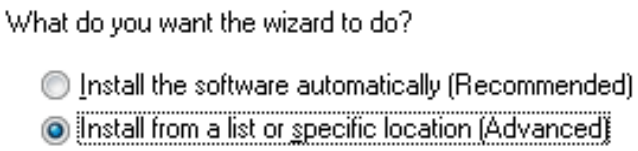


图 10. 从指定位置安装

5. 选择“Search for Best Driver”（搜索最佳驱动程序）（请见图 11）。 取消选中的搜索可拆除介质，选中包含指定位置并浏览至目录 [USB Drive]:\XDS100 Drivers。

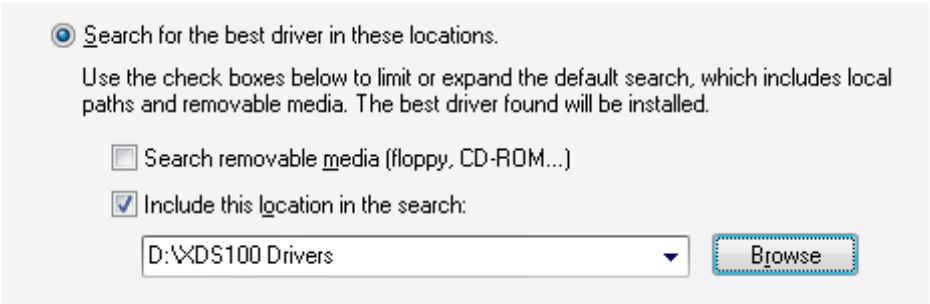


图 11. 搜索最佳驱动程序

6. 单击下一步，驱动程序将被安装。 此驱动程序安装信息将出现三次，每次都重复这个过程。
7. 通过 [Main]-TB1-TB8 将 LED 控制面板连接至电路板。
8. 检验以下的跳线设置：
   (a) [M1]-J1 上未放置跳线。
   (b) 一个跳线被放置在 [M6]-J4 上。
   (c) 一个跳线被放置在 [Main]-J2 上。
9. 将 12V DC 电源连接至 [M1]-JP1 来为辅助电源轨供电。
10. 将一个 12V - 18V DC 电源的输出连接至 [Main]-BS1 并将 [Main]-BS2/BS3 接地来为主电源轨供电。

6 软件设置

6.1 安装 Code Composer Studio 和 controlSUITE

1. 如果未经安装，请从套件中的 USB 驱动从安装 Code Composer Studio v4.x。
2. 打开http://www.ti.com/controlsuite并运行 controlSUITE 安装程序。 选择安装“Multi-DC/DC Color LED Kit”（多 DC/DC 彩色 LED 套件）软件并且还允许安装程序自动下载将校验的软件。

6.2 将 Code Composer Studio 设置成与多 DC/DC 彩色 LED 套件一同工作

1. 打开 “Code Composer Studio v4”。
2. 一旦 Code Composer Studio 打开，会出现工作区启动器来要求您选择一个工作区位置。 （请注意工作区是硬驱动上的一个位置，在这个位置上存放着所有针对 LED 的用户设置：打开哪个项目、选择什么配置以及将项目保存至何处等等。 这可以是磁盘上的任一位置；下面提到的位置仅供参考。 还要注意的是如果这不是您第一次运行 Code Composer Studio，那么将不会出现这个对话框。）

   (a) 单击“Browse…”（浏览）按钮。

   (b) 通过生成所需的新文件夹来创建下面的路径。

   (c) “C:\Documents and Settings\<username>\My Documents\CCSv4_workspaces\LED-ColorMix”。

   (d) 取消选中“Use this as the default and do not ask again”（使用这个作为缺省路径，不要再次询问）。

   (e) 单击“OK”。

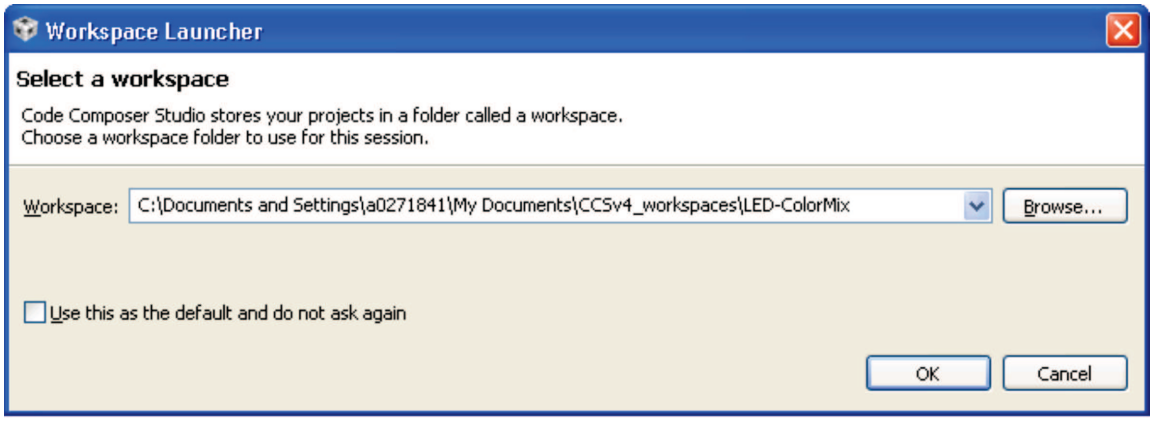


图 **12.** 工作区启动器

3. 配置 Code Composer Studio，使其知道连接到哪个 MCU。 单击“Target → New Target Configuration…”（目标 → 新目标配置）。 将新配置命名为“XDS100 F28027.ccxml”。 确保选中“Use shared location”（使用共用位置），然后单击“Finish”（完成）。

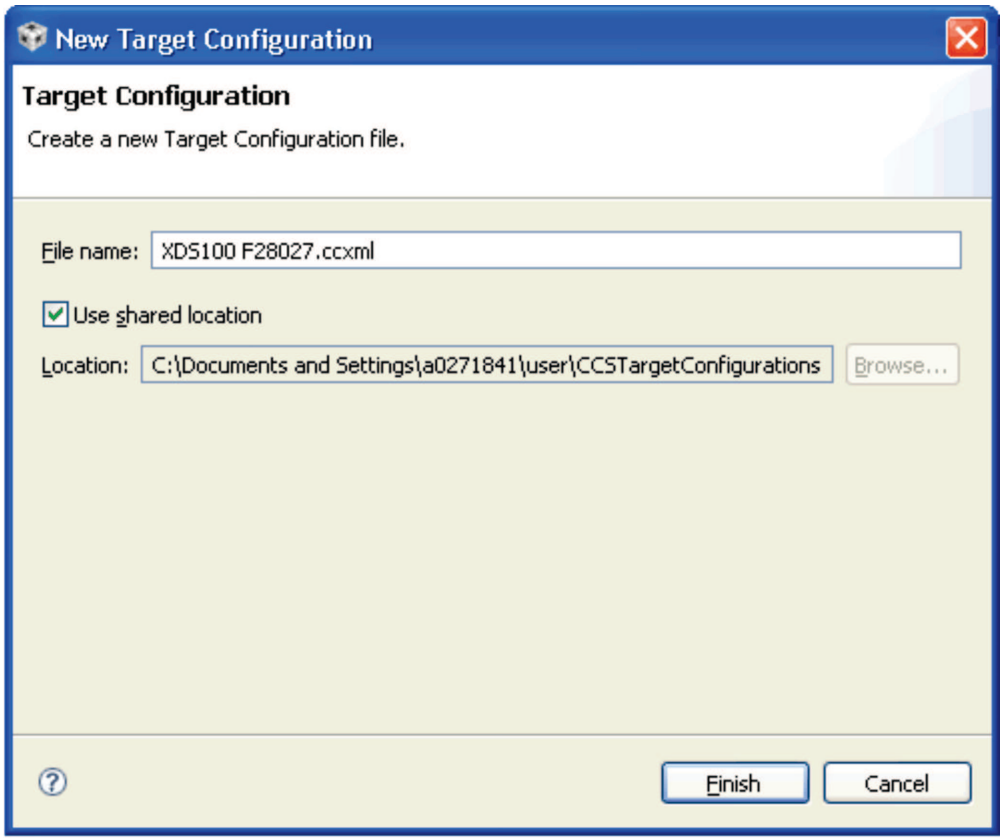


图 **13.** 创建一个目标配置

4. 这将打开一个如图 14中所见的新标签。 如下选择并进入选项：
   (a) 连接 - 德州仪器 (TI) XDS100v1 USB 仿真器。
   (b) 器件 - TMS320F28027.
   (c) 单击保存。
   (d) 关闭 "XDS100 F28027.ccxml”标签页。

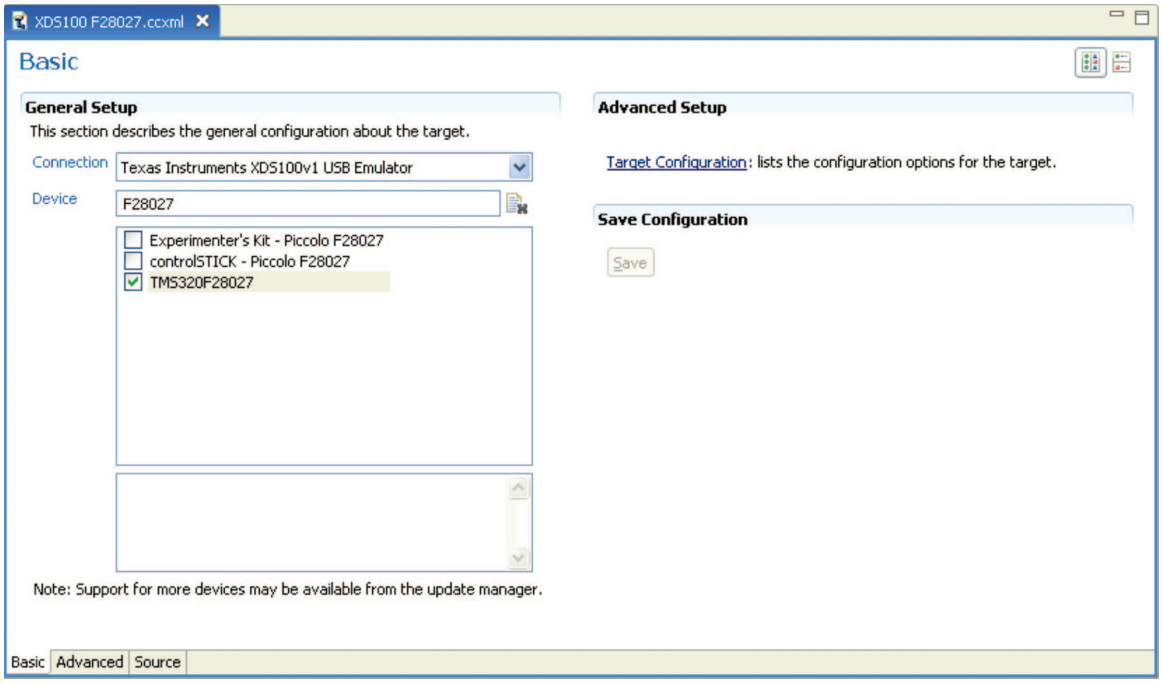


图 **14.** 配置一个新目标

5. 假定这是您第一次使用 Code Composer，现在“XDS100 F28027”配置被设定为针对 Code Composer Studio 的缺省目标配置。 这通过进入“View → Target Configurations”（查看 → 目标配置）来选中。 在“User Defined”（由用户定义）部分，右键单击“XDS00 F28027.ccxml”文件并选择“Set as Default”（设定为缺省值）。 这个标签页还使得您能够再次使用已有的目标配置并将它们连接至特定项目。

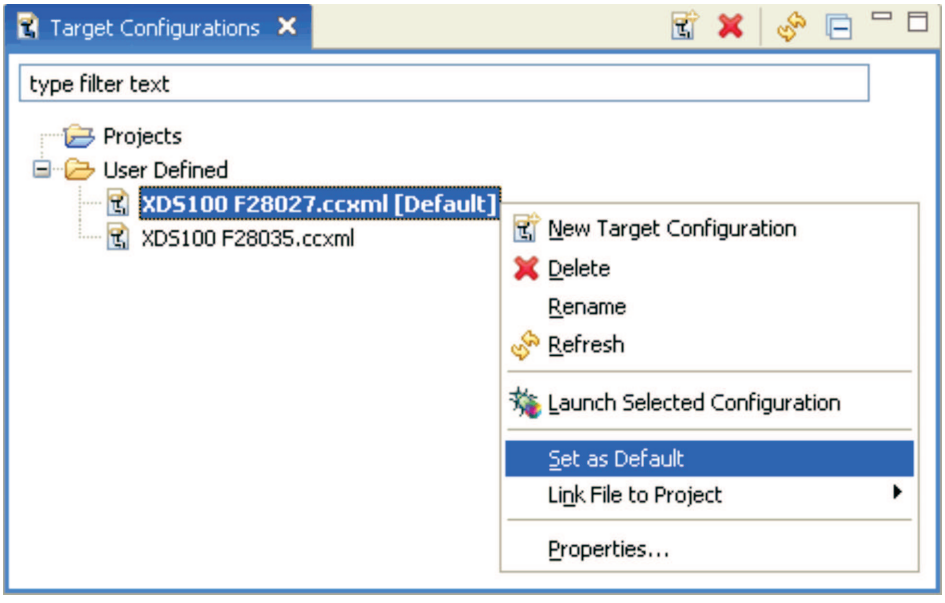


图 **15.** 设定一个缺省目标配置

6. 通过单击“Project → Import Existing CCS/CCE Eclipse Project”（项目 → 导入已有的 CCS/CCE Eclipse 项目）来将 LED-ColorMix 项目添加到您当前的工作区中。
   (a) 浏览多 DC/DC 彩色 LED 套件文件夹内的 LED-ColorMix 项目文件夹。 此文件将为：“C:\TI\controlSUITE\development_kits\Multi-DCDC-Color-LED-Kit_v1.0\”。
   (b) 单击“Finish”来将LED-ColorMix 项目载入工作区。

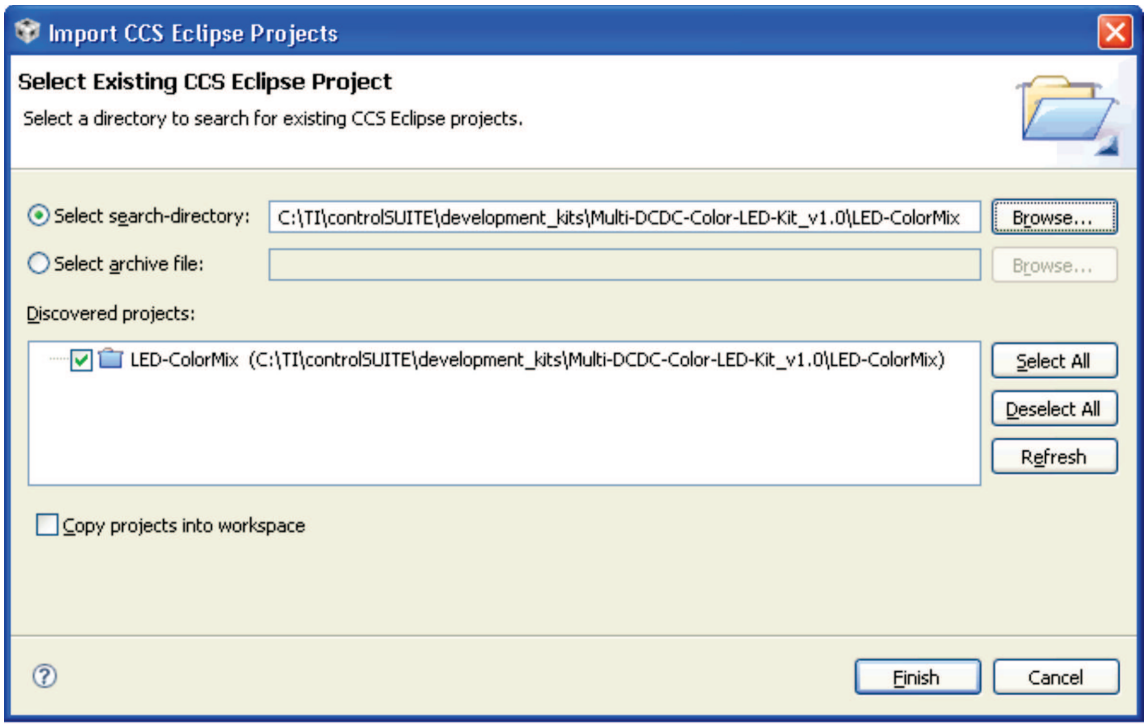


图 16. 将多 DC/DC 彩色 LED 套件项目导入到您的工作区

7. LED-ColorMix 项目将被设定为激活的项目。 右键单击项目名称并单击“Save as Active Project”（保存为激活的项目）。 展开项目的文件构建。

6.3 针对 *Lighting_DCDC* 项目的递增系统构建

逐步构建照明系统以使最终系统能够放心运行。 设计了两个阶段（系统构建）来验证系统中使用的主要系统模块。

- 构建 1：开环测试，检查 DC/DC 级和 LED 灯串的基本运行
- 构建 2：DC/DC 级和 LED 灯串的闭环运行

表 5. 每个递增系统构建中的测试模块(1)

软件模块	阶段 1	阶段 2
PWMDRV_2ch_UpCnt	√√	√
ADCDRV_1ch	√√	√
CNTL_2P2Z		√√

(1) √ 符号意味着这个模块被用在这个阶段，而√ √ 符号意味着在这个模块在这个阶段被测试。

7 构建 1：DC/DC 级和 LED 灯串的开环路运行

注： 这个部分认为5 节和6 节已经完成。 如果未完成，在继续操作前完成这些部分。

这个构建的目标如下：

- 验证电路板上的功率级运转正常。
- 控制不同功率级上的占空比并评估它们的输出。

图 17中描述了软件中使用的系统组件。

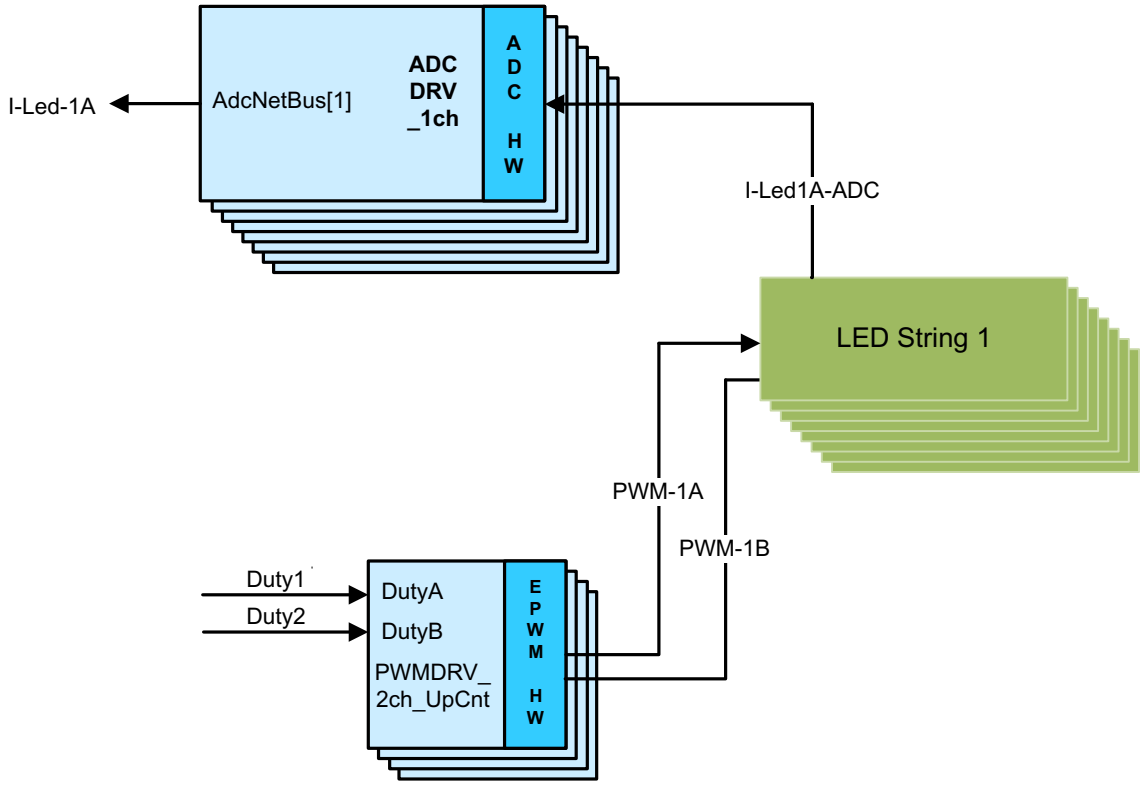


图 17. 构建级 1 方框图

7.1 检查此项目

在最初的引导过程完成后，软件从主函数中开始执行。

1. 打开 LED-ColorMix-Main.c 并找到*main()*函数（第 372 行）。 软件在*main()*函数中做的第一件事情就是调用一个 LED-ColorMix-DevInit_F2802x.c 中被称为*DeviceInit()*的函数。
2. 通过双击项目窗口中的文件名可打开并检查 LED-ColorMix-DevInit_F2802x.c。 在这个文件中，不同的外设时钟被启用或被禁用并且定义了功能引脚分配，此引脚分配配置了哪个外设从哪个引脚输出。
   (a) 请确认 ADC 和 PWM-14 外设时钟被启用（第 99-117 行）。 还需要确认 GPIO00-GPIO07 被配置为 PWM 输出（第 136-182 行）。
3. LED-ColorMix 与递增构建一同提供，在这个构建中系统的不同组件和宏区块被逐一连接来形成完整系统。 这有助于逐步调试并了解此系统。
   (a) 从 C/C++ 项目标签页上打开文件 LED-ColorMix-Settings.h 并确保 INCR_BUILD 被置位为 1 且保存这个文件。 在您测试了构建 1 后，这个变量需要被重新定义来移到到构建 2，以此类推，直到完成所有构建。
4. 返回 LED-ColorMix-Main.c 文件并留意不同的递增构建配置代码。 构建级 1 配置代码出现在 595-630 行上。 其中配置了 ADC，PWM 和宏区块连接。 请注意，如果在 LED-ColorMix-Settings.h 中，INCR_BUILD 被置位为 1，634-710 行将被编译器忽略，这是因为它们不属于当前的构建。 根据 INCR_BUILD 的值来启用或禁用代码的相似方法也在 LED-ColorMix-ISR.asm 完成。

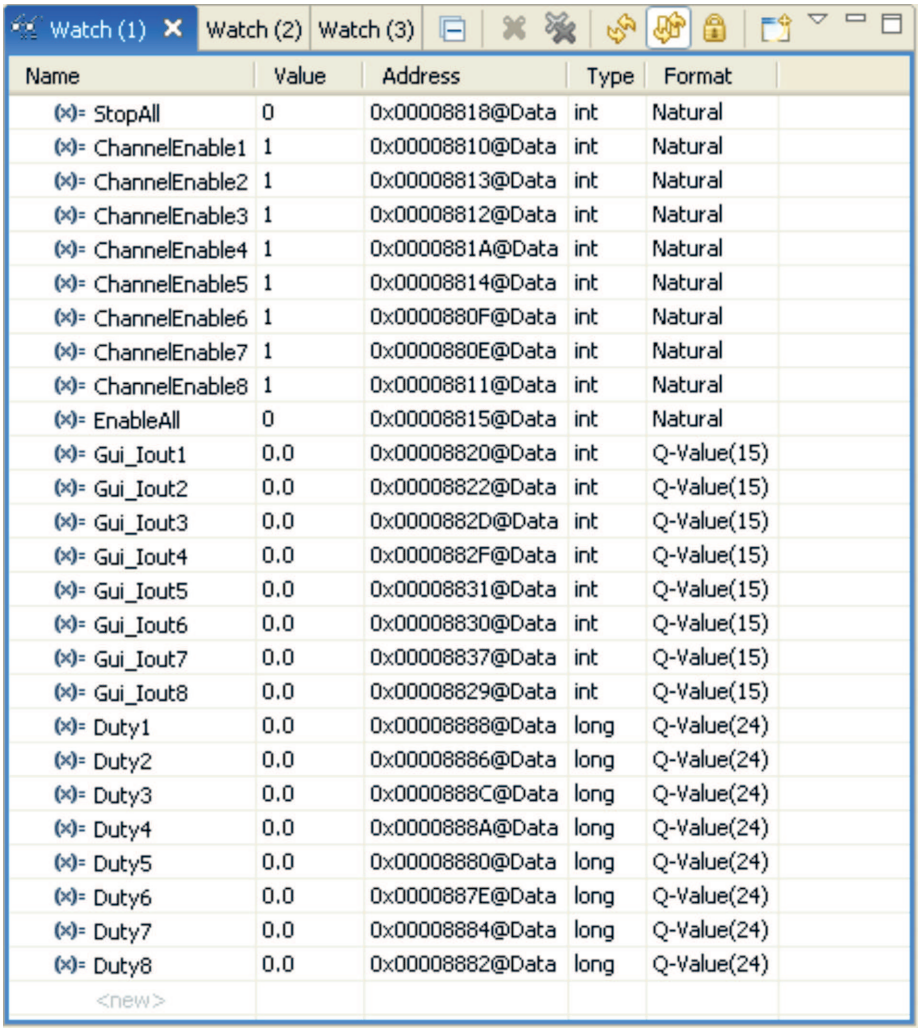
7.2 建立和加载项目

1. 右键单击项目名称并单击“Rebuild Project”（重建项目），请观察控制台窗口。 项目中的任何错误被显示在控制台窗口中。
2. 在成功完成项目建立时，单击“Debug”（调试）按钮  ，此按钮位于屏幕的左上侧。 现在，IDE 将自动连接至目标，将输出文件载入到器件内并换到调试视图。
3. 现在单击实时模式按钮  ，此按钮名为“Enable silicon real-time mode”（启用芯片实时模式）。 这使得您能够实时编辑并查看变量，而无需暂停程序。
4. 也许会出现一个消息框。 如果是这样的话，选择 YES 来启用调试事件。 这将把状态寄存器 1 (ST1) 的位 1（DGBM 位）置位为“0”。 DGBM 是调试使能屏蔽位。 当 DGBM 位被置位为“0”时，为了更新调试器窗口，存储器和寄存器值可被传递到主机处理器。

7.3 设置观察窗口和曲线图

1. 单击：菜单条上的 View → Watch 打开一个*观察窗口*来查看项目中正在使用的变量。 将图 18中出现的变量添加到观察窗口。 通过右键单击一个特定变量然后选择一个 Q 值来改变一个变量的格式。 改变每个变量的格式来与图形相匹配。 在构建 1 中时，这个观察窗口中的变量被用来控制和监控电路板的状态。

   提示：在按住 Shift 键的同时单击鼠标可以选择多个变量或项，然后右键单击并改变所有选中的变量的格式。



Name	Value	Address	Type	Format
StopAll	0	0x00008818@Data	int	Natural
ChannelEnable1	1	0x00008810@Data	int	Natural
ChannelEnable2	1	0x00008813@Data	int	Natural
ChannelEnable3	1	0x00008812@Data	int	Natural
ChannelEnable4	1	0x0000881A@Data	int	Natural
ChannelEnable5	1	0x00008814@Data	int	Natural
ChannelEnable6	1	0x0000880F@Data	int	Natural
ChannelEnable7	1	0x0000880E@Data	int	Natural
ChannelEnable8	1	0x00008811@Data	int	Natural
EnableAll	0	0x00008815@Data	int	Natural
Gui_Iout1	0.0	0x00008820@Data	int	Q-Value(15)
Gui_Iout2	0.0	0x00008822@Data	int	Q-Value(15)
Gui_Iout3	0.0	0x0000882D@Data	int	Q-Value(15)
Gui_Iout4	0.0	0x0000882F@Data	int	Q-Value(15)
Gui_Iout5	0.0	0x00008831@Data	int	Q-Value(15)
Gui_Iout6	0.0	0x00008830@Data	int	Q-Value(15)
Gui_Iout7	0.0	0x00008837@Data	int	Q-Value(15)
Gui_Iout8	0.0	0x00008829@Data	int	Q-Value(15)
Duty1	0.0	0x00008888@Data	long	Q-Value(24)
Duty2	0.0	0x00008886@Data	long	Q-Value(24)
Duty3	0.0	0x0000888C@Data	long	Q-Value(24)
Duty4	0.0	0x0000888A@Data	long	Q-Value(24)
Duty5	0.0	0x00008880@Data	long	Q-Value(24)
Duty6	0.0	0x0000887E@Data	long	Q-Value(24)
Duty7	0.0	0x00008884@Data	long	Q-Value(24)
Duty8	0.0	0x00008882@Data	long	Q-Value(24)
<new>				

图 **18.** 为构建 **1** 配置观察窗口

2. 单击观察窗口中的 Continuous Refresh（持续刷新）按钮。 这将启用窗口的实时运行模式。 通过单击任一观察窗口中的下箭头，您可以选择“Customize Continuous Refresh Interval”（定制持续刷新间隔时间）并编辑针对此观察窗口的刷新率。 请注意，将间隔时间选的过短会影响性能。

7.4 运行代码：

1. 通过按下 Debug Tab（调试标签页）内的 Run（运行）按钮来运行代码 。
2. 现在项目应该运行，而观察窗口中的值应该持续更新。 您也许想按照您的喜好来重新调整窗口的大小并重新组织此窗口。 通过拖拽和放置不同的窗口可轻松实现此操作。
3. 当主电源被置位为 0V 或被关闭时，验证 PWM 和 ADC 电路。
   (a) 将 Duty[1-8] 设定为 0.50，这个值对应 50% 占空比。 通过仔细检查位于电路板上的 PWM 测试点来验证。
   (b) 验证所有电压和电流反馈变量接近于 0（少于 0.005）。 由于常常在 ADC 上出现某些低电平噪声，所以它们不太可能正好为 0.0。
4. 一旦完成，将 Duty[1-8] 设定为 0.0。

5. 在主电源被设定为 12V 时将 Duty 设定为 0.57。这将把控制 LED 灯串 1 的 MOSFET 的占空比设定为 57%。 证实 LED 灯串 1 打开并且 Gui_Iout1 变为大约 0.010。 小心调整 Duty1 并证实在您增加或减少 Duty1 时，Gui_Iout1 增加或减少。 请注意，不要超过电路板的电流额定值。
6. 通过分别调整 Duty[2-8] 并且监控 Gui_Iout 可以检查 LED 灯串 2-8。 使用以下的 Duty{X} 值并校验相应的 Gui_Iout{X}：
   (a) Duty2：0.59，Gui_Iout2：0.010
   (b) Duty3：0.40，Gui_Iout3：0.010
   (c) Duty4：0.57，Gui_Iout4：0.010
   (d) Duty5：0.59，Gui_Iout5：0.010
   (e) Duty6：0.40，Gui_Iout6：0.010
   (f) Duty7：0.33，Gui_Iout7：0.010
   (g) Duty8：0.33，Gui_Iout8：0.010
7. 一旦完成，将 Duty[1-8] 设定为 0.0。
8. 禁用实时模式  并复位处理器  (Target → Reset → Reset CPU)，然后通过单击  (Target → Terminate All) 来终止所有调试会话。 这将暂停程序并从 MCU 上断开 Code Composer Studio。

8 构建 2：DC/DC 级和 LED 灯串的闭环路运行

这个构建的目的如下：

使用支持闭环路反馈的平均电流模式控制来调节每个 LED 灯串汲取的电流。

图 19中描述了软件中使用的系统组件。

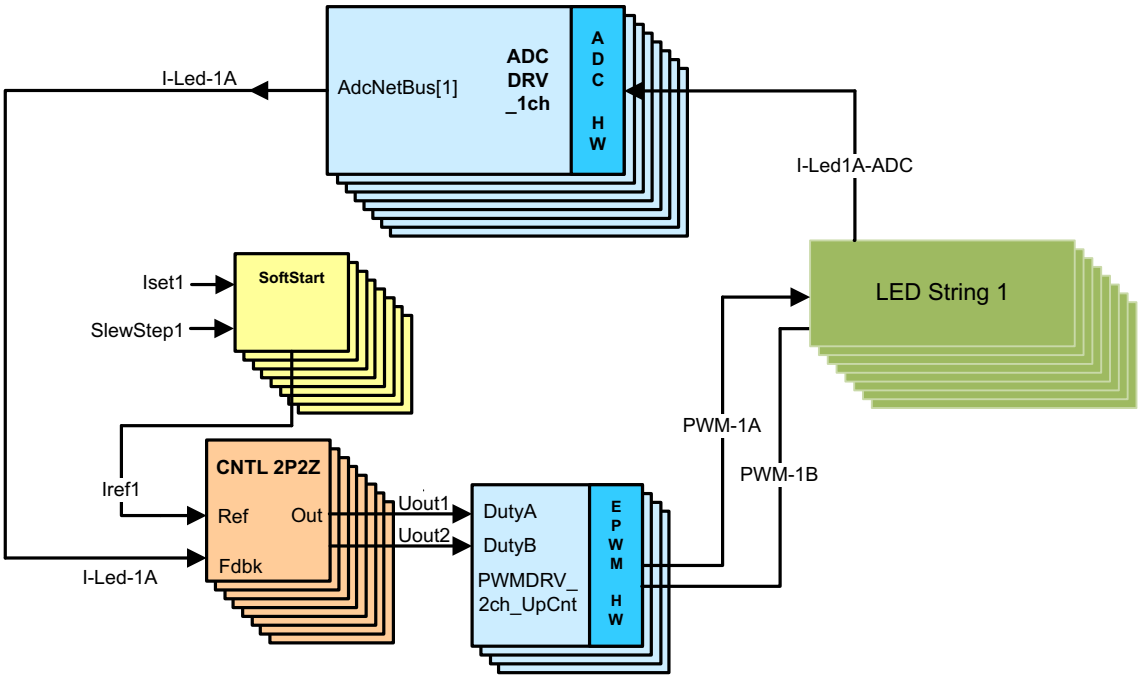


图 **19.** 构建级 **2** 方框图

*针对此构建的指令比第一个构建中的指令更简洁。 要获得进一步的指导，请参见**7** 节中的指令。 在运行构建 **2** 之前运行构建 **1**。*

8.1 建立和加载项目

1. 打开 LED-ColorMix-Settings.h 并将递增构建级改为 2 (#define INCR_BUILD 2)。 保存此文件。

2. 右键单击文件名并选择“Rebuild Project”。
3. 在成功完成项目建立时，单击“Debug”（调试）按钮  ，此按钮位于屏幕的左上侧。 现在，IDE 将自动连接至目标，将输出文件载入到器件内并换到调试视图。
4. 单击实时模式按钮  ，此按钮名为“Enable silicon real-time mode”。 这使得您能够实时编辑并查看变量，而无需暂停程序。

8.2 设置观察窗口和曲线图

1. 将一个观察窗口添加到工作区并添加变量，并将其设定为使用如图 20中所示的正确格式。 在构建 2 中时，这个观察窗口中的变量被用来控制和监控电路板的状态。

Watch (1) | Watch (2) | Watch (3)

Name	Value	Address	Type	Format
StopAll	0	0x00008818@Data	int	Natural
ChannelEnable1	1	0x00008810@Data	int	Natural
ChannelEnable2	1	0x00008813@Data	int	Natural
ChannelEnable3	1	0x00008812@Data	int	Natural
ChannelEnable4	1	0x0000881A@Data	int	Natural
ChannelEnable5	1	0x00008814@Data	int	Natural
ChannelEnable6	1	0x0000880F@Data	int	Natural
ChannelEnable7	1	0x0000880E@Data	int	Natural
ChannelEnable8	1	0x00008811@Data	int	Natural
EnableAll	0	0x00008815@Data	int	Natural
Gui_Iout1	0.0	0x00008820@Data	int	Q-Value(15)
Gui_Iout2	0.0	0x00008822@Data	int	Q-Value(15)
Gui_Iout3	0.0	0x0000882D@Data	int	Q-Value(15)
Gui_Iout4	0.0	0x0000882F@Data	int	Q-Value(15)
Gui_Iout5	0.0	0x00008831@Data	int	Q-Value(15)
Gui_Iout6	0.0	0x00008830@Data	int	Q-Value(15)
Gui_Iout7	0.0	0x00008837@Data	int	Q-Value(15)
Gui_Iout8	0.0	0x00008829@Data	int	Q-Value(15)
Gui_Iset1	0.0	0x00008824@Data	int	Q-Value(15)
Gui_Iset2	0.0	0x00008827@Data	int	Q-Value(15)
Gui_Iset3	0.0	0x00008826@Data	int	Q-Value(15)
Gui_Iset4	0.0	0x0000882E@Data	int	Q-Value(15)
Gui_Iset5	0.0	0x0000881E@Data	int	Q-Value(15)
Gui_Iset6	0.0	0x0000881D@Data	int	Q-Value(15)
Gui_Iset7	0.0	0x0000881F@Data	int	Q-Value(15)
Gui_Iset8	0.0	0x00008821@Data	int	Q-Value(15)
<new>				

图 20. 为构建 2 配置观察窗口

2. 单击观察窗口中的 Continuous Refresh（持续刷新）按钮。 

8.3 运行代码

1. 通过按下 Debug Tab（调试标签页）内的 Run（运行）按钮来运行代码  。
2. 项目应该运行，而观察窗口中的值应该持续更新。 您也许想按照您的喜好来重新调整窗口的大小并重新组织此窗口。 通过拖拽和放置不同的窗口可轻松实现此操作。

3. 在针对构建 2 的观察窗口中，将变量 Gui_Iset1 设定为 0.01，对应于 0.010A。 当电流基准被改变时，应该更改 Gui_Iout1 的值以与 0.01 的基准值相匹配。
4. 将 Gui_Iset1 的值改为 0.03。 请注意，LED 灯串的亮度已经随着增加的电流而增加。
5. 继续用 0.0 至 0.100 之间不同的值来编辑 Gui_Iset1 以及 Gui_Iset[2-8]。

注: 不要将每个灯串的输出电流增加到 0.100A 以上。 这是因为升压级电感器只被设计用来处理 0.350A 的电流。 如果需要更大的输出电流，用那些具有更高电流额定值的电感器来替换此升压电感器。

注: 通过改变红色、绿色和蓝色 LED 灯串间的电流比可生成不同的颜色。

6. 一旦完成，将 Gui_Iset[1-8] 设定为 0.0 或者将 StopAll 设定为 1。
7. 禁用实时模式  并复位处理器  (Target → Reset → Reset CPU)，然后通过单击  (Target → Terminate All) 来终止所有调试会话。 这将暂停程序并从 MCU 上断开 Code Composer Studio。

9 对于进一步开发研究的思考

- I2C 温度感测：

  为了提供与面板上与 LED 温度有关的系统信息，应该将一个温度传感器放置在 LED 面板上。 然后，这个温度信息可被发送到一个外部主机，此主机被用来提供针对 LED 的过热保护，或（与一个查找表一起）被用来提供热补偿。
- LED 电流偏移补偿

  电阻器、运算放大器和其它离散组件在它们设计中有一个固有的额定误差。 这些误差常常表现为系统中的 ADC 电压偏移误差。 为了减少这个可变性和误差，ADC 电压可在一个已知状态上测量一个电压并推导出此误差。 然后，数字控制器可以在中断内减去这个误差。
- 通信协议：

  很多照明系统还利用通信协议来增加特性并提升性能和效率。 照明系统使用诸如 DMX512，DALI，甚至电力线通信 (PLC) 的通信。 电路板上包含了一个用作示例的实验 DMX512 电路。
- 使用多 DC/DC 彩色 LED 套件来启动您的原型机：

  所有与这个套件捆绑在一起的软件和硬件无需许可证免费使用。 请自由取用任一软件和硬件作为您自己的设计资源。

10 参考书目

要获得更多信息，请参见位于以下链接内的文档：www.ti.com/controlsuite：

- LED-ColorMix-AppNote - 提供于 CCS4 项目相关的详细信息。

  \TMDSRGBLEDKIT_v1.0\~Docs\LED-ColorMix-AppNote[R1].pdf
- LED-ColorMix-HWGuide - 提供与电路板上的硬件相关的详细信息。

  \TMDSRGBLEDKIT_v1.0\~Docs\LED-ColorMix-HWGuide[R1].pdf
- LED-ColorMix-HWdevPkg - 一个包含与电路板上硬件相关的文件的文件夹（电路原理图，物料清单，Gerber 文件，印刷电路板 (PCB) 布局布线等）。

  \TMDSRGBLEDKIT_v1.0\~LED-ColorMix-HWdevPkg[R1]\

重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 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， 并有权根据 JESD48 最新标准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 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。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。除非适用法律做出了硬性规定，否则没有必要对每种组件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 组件或服务的组合设备、机器或流程相关的 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中 TI 信息的重要部分，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。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时，如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分，则会失去相关 TI 组件 或服务的所有明示或暗示授权，且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意，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与其产品及在其应用中使用 TI 产品 相关的所有法律、法规和安全相关要求。客户声明并同意，他们具备制定与实施安全措施所需的全部专业技术和知识，可预见 故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因 在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中，为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特 有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此，此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III（或类似的生命攸关医疗设备）的授权许可，除非各方授权官员已经达成了专门管控此类使 用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同 意，对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用，其风险由客户单独承担，并且由客户独 力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品，这些产品主要用于汽车。在任何情况下，因使用非指定产品而无法达到 ISO/TS16949 要求，TI不承担任何责任。

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP应用处理器	www.ti.com/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity	德州仪器在线技术支持社区	www.deyisupport.com

邮寄地址： 上海市浦东新区世纪大道 1568 号，中建大厦 32 楼 邮政编码： 200122