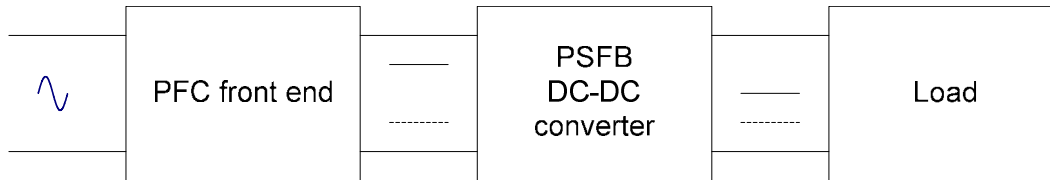




摘要

本文详细介绍了一种数字控制无桥式功率因数校正 (BL PFC) 转换器的实现过程。使用一个 C2000™ F28035 controlCARD 和一个 300W BL PFC EVM 来实现整个系统。

根据限制输入电流谐波含量的各种规定，特别是 IEC 61000-3-2 标准（规定电子负载可向供电线路注入的谐波成分），功率因数校正 (PFC) 级已成为大多数整流器设计中必不可少的一部分。此 PFC 级构成了隔离式交流/直流整流器系统的前端，如下所示。



PFC 转换器从交流电源中提取正弦输入电流，同时调节其输出电压，以便向下游直流/直流转换器提供稳定的高直流总线电压。直流/直流级通常是一个移相全桥 (PSFB) 转换器，它将 PFC 级的高直流总线电压转换为较低的电压（如 +12V）或中间配电电压（通常接近 48V）。移相全桥 (PSFB) 级让此离线整流器系统实现电压转换和高频隔离。本文详细介绍了 PFC 级的实现，专门介绍了控制无桥功率因数校正 (BL PFC) 前端的硬件设计和相应软件。

此 PFC EVM 使用 F28035 controlCARD 而非 F28027 controlCARD，其原因是 F28027 controlCARD 中没有 RC 滤波器。在 F28035 controlCARD 上可感测两个 BL PFC 开关电流的两个模数转换器 (ADC) 通道中都提供这种 RC 滤波器。然而，在 F28027 controlCARD 上，两个 ADC 通道中只有一个具有此 RC 滤波器。此 F28027 controlCARD 的修改版可轻松实现对此 BL PFC EVM 的完全控制。

内容

1 引言	3
1.1 PFC 级实现	3
1.2 BL PFC 电气规范	4
2 软件概述	5
2.1 软件控制流程	5
2.2 增量构建	7
3 运行增量构建的流程	8
3.1 构建 1：通过 ADC 测量进行开环升压	8
3.1.1 构建 1 目标	8
3.1.2 构建 1 概述	8
3.1.3 构建 1 保护	11
3.1.4 构建 1 流程	11
3.1.4.1 第 1.1 步：启动 CCS 和打开项目	11
3.1.4.2 第 1.2 步：器件初始化、主菜单和 ISR 文件	12
3.1.4.3 第 1.3 步：构建和加载项目	13
3.1.4.4 第 1.4 步：调试环境窗口	14
3.1.4.4.1 第 1.5 步：使用实时仿真	14
3.1.4.4.2 第 1.6 步：运行构建 1 的代码	14
3.2 构建 2：具有闭合电流环路的 BL PFC	16
3.2.1 构建 2 目标	16
3.2.2 构建 2 概述	16

3.2.3 构建 2 流程.....	18
3.2.3.1 第 2.1 步：构建和加载项目.....	18
3.3 构建 3：具有闭合电压和电流环路的 BL PFC.....	20
3.3.1 构建 3 目标.....	20
3.3.2 构建 3 概述.....	20
3.3.3 构建 3 流程.....	21
3.3.3.1 第 3.1 步：构建和加载项目.....	21
4 测试结果.....	23
5 参考文献.....	25
6 修订历史记录.....	25

商标

C2000™ and Code Composer Studio™ are trademarks of Texas Instruments.

所有商标均为其各自所有者的财产。

1 引言

PFC 级的作用是在提取正弦波输入电流的同时，将交流电源电压转换为稳压直流总线电压。通常，这是在使用桥式整流器后再实施升压 PFC 级实现的。然而，若要降低二极管电桥中的功率损耗和提高系统效率，另一种方法是使用一种无桥系统，并使用两个升压级来实现整流和功率因数控制。在这种方法中，两个升压级在交流电源电压的正负半周期中交替运行。C2000 微控制器及其片上 PWM 和 ADC 模块能够实现对这种无桥式 PFC (BL PFC) 系统的完全数字化控制。

1.1 PFC 级实现

图 1-1 介绍了一种基于 C2000 的无桥式 PFC 转换器控制系统。输入交流电压通过输入 EMI 滤波器和浪涌控制继电器应用到 PFC 转换器。PFC 级由两个升压转换器组成，每个升压转换器在半周期输入交流线路周期内以升压模式交替工作。

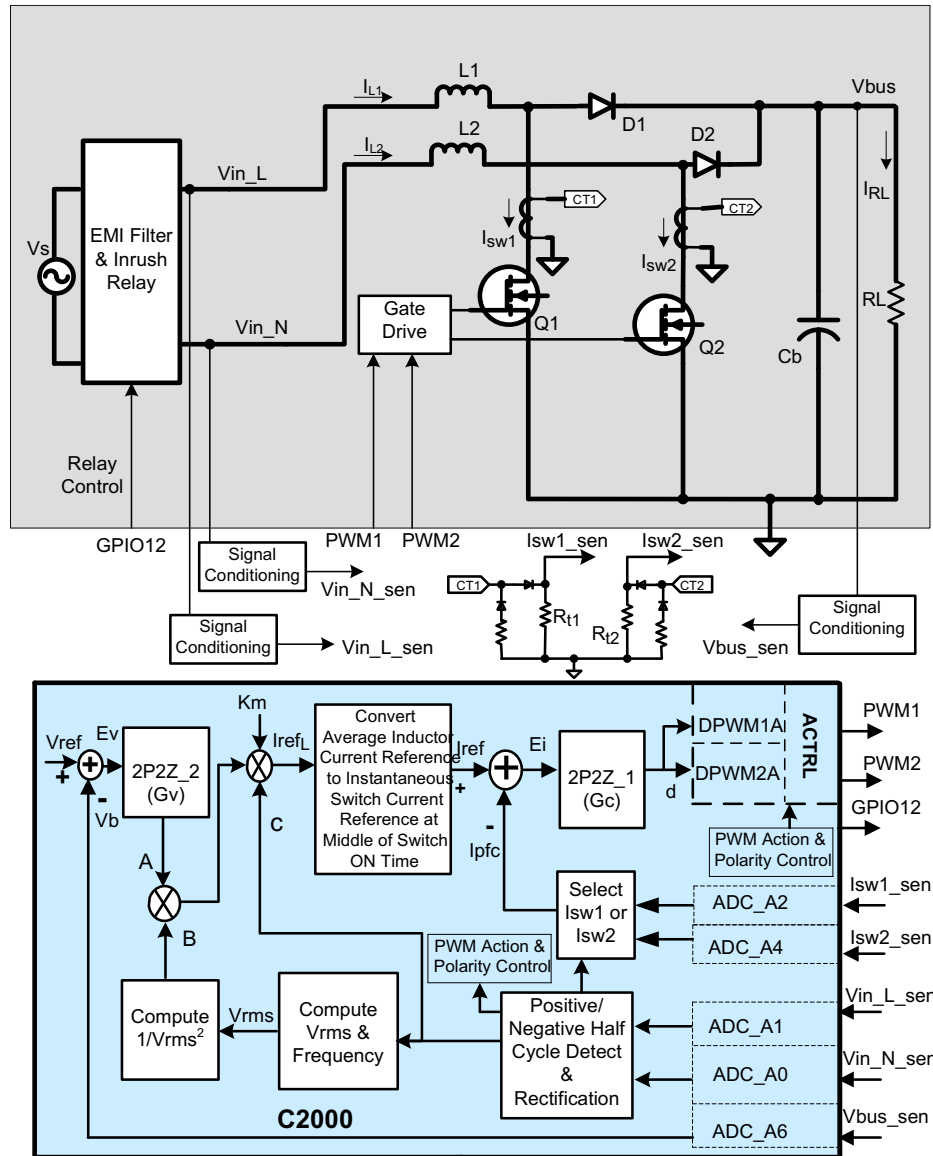


图 1-1. 使用 C2000 微控制器的无桥式 PFC 转换器控制

在半周期内，一个转换器将交流线路电压转换为直流总线电压，而另一个转换器则在另外半个周期内执行相同的操作。电感器 $L1$ 、MOSFET 开关 $Q1$ 、电流感应变压器 $CT1$ 和二极管 $D1$ 一起形成一个升压级，而 $L2$ 、 $Q2$ 、 $CT2$ 和 $D2$ 则形成另一个升压级。升压转换器输出上的一个电容器 Cb 作为一个储能器，为 RL 指定的 PFC 负载提供稳压直流电压。

图 1-1 指出使用 C2000 微控制器 (MCU) 完全控制此无桥式 PFC 转换器所需的所有接口信号。MCU 通过五个反馈信号、两个 PWM 输出和一个 GPIO 输出来控制硬件。感测并反馈给 MCU 的信号包括线电压和中性点电压 (V_{in_L} 和 V_{in_N})、两个 PFC 开关电流 (I_{sw1} 、 I_{sw2}) 和升压输出电压 (V_{bus})。这些感测到的信号用来执行此 BL PFC 转换器的电压和电流控制环路。

将通过一个 ADC 通道感测到的直流总线电压 V_{bus} 与参考总线电压 V_{ref} 进行比较。由此产生的误差信号 E_v 随后被输入电压环路控制器 G_v ，该控制器将总线电压调节到基准水平。电压控制器 G_v 具有双极点-双零点 (2P2Z) 补偿器的形式。 G_v 的输出由图 1-1 中的字母 A 表示，与 PFC 转换器的功率传输量成正比。然后将输出 A 乘以图 1-1 中由 B、C 和 K_m 表示的三个参数，以形成 PFC 电流控制环路的参考电流指令 I_{ref} 。由 B 表示的信号是 RMS 输入电压平方的倒数，这使得 PFC 系统能够快速实现前馈控制。信号 C 与整流输入电压成正比，它调制电压控制器输出 A，使得 PFC 输入电流具有与 PFC 输入电压相同的形状。参数 K_m 被称为倍增增益，用于调整与 PFC 转换器的全输入电压范围相对应的 I_{ref} 范围。乘法器的输出为控制平均电感器电流 I_{refL} 提供参考信号。然而，对于 BL PFC 系统，电流反馈来自 PFC 开关 ($Q1$ 和 $Q2$)，而不是来自升压电感器 ($L1$ 和 $L2$)。这意味着平均电感器电流控制的参考信号 I_{refL} 在用于 PFC 开关电流控制之前必须进行转换。在图 1-1 中，这由 I_{refL} 和 I_{ref} 之间的转换块表示。然后将 PFC 电流控制环路的参考电流指令 I_{ref} 与通过两个 ADC 通道感测到的 PFC 开关电流 I_{pfc} 进行比较。然后，将产生的电流误差信号 E_i 输入电流环路控制器 G_c ，该电流控制器 G_c 生成 PFC 占空比命令 d ，使得 PFC 开关电流能够跟踪参考电流 I_{ref} 。

除了实现电压和电流环路控制器之外，C2000 MCU 还利用感测到的线电压和中性点电压信号来确定输入电压的极性 (+ve 和 -ve 半周期)，并计算整流输入电压、RMS 输入电压和输入线路频率。根据输入电压的极性，C2000 控制器会选择合适的 PFC 开关电流 (I_{sw1} 或 I_{sw2}) 作为 PFC 电流反馈 I_{pfc} 。极性信息还用于将相应的 PFC 开关 ($Q1$ 和 $Q2$) 设置为 PWM 模式 (升压 PFC) 或强制开启模式。所有这些时间关键型功能都在一个由 C2000 微控制器高速 CPU、中断、片上 12 位 ADC 模块和高频 PWM 模块打造出的快速采样环路中实现。以下各章详细介绍了此软件算法。

1.2 BL PFC 电气规范

下文列出了 C2000 BL PFC EVM 的主要亮点：

- 输入电压 (交流线)：95V (最小值) 至 240V (最大值)，47~63Hz
- 400Vdc 输出
- 300W 输出功率
- 满载效率大于 93%
- 50% 或更大负载下的功率因数：0.98 (最小值)
- PWM 频率为 200kHz

2 软件概述

2.1 软件控制流程

C2000 BLPFC 项目主要使用“C-background/ASM-ISR”框架。主快速 ISR (100kHz) 在汇编环境下运行。然而，一个慢速 ISR (10kHz) 也在 C 语言环境下运行。这个慢速 ISR 可由快速 ISR 中断。

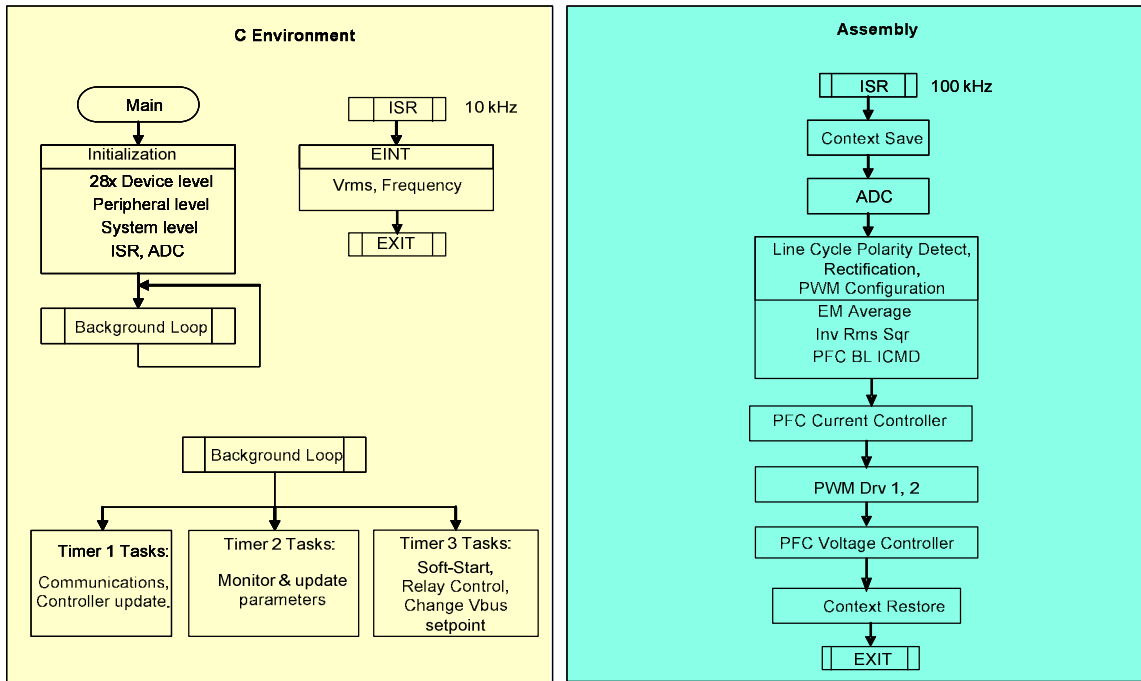


图 2-1. BL PFC 软件流程图

此项目使用 C 语言代码作为应用的主要支持程序，并负责所有系统管理任务、决策制定、智能和主机交互。此汇编代码被严格限制为快速中断处理例程 (ISR)，此例程运行全部关键控制代码。通常，这包括读取 ADC 值、输入线路周期极性检测、感应线路电压整流、控制计算和 PWM 更新。在 C 环境中，慢速 ISR 负责计算输入线路电压的 RMS 电压和频率。图 2-1 描述了此项目的通用软件流程。

本项目中使用的关键框架 C 语言文件为：

BridgelessPFC-Main.c 此文件用于应用程序的初始化、运行和管理。

**BridgelessPFC-
DevInit_F2802x.c 或
BridgelessPFC-
DevInit_F2803x.c**

根据 BL PFC EVM 中使用的控制卡 (分别为 F2802x 或 F2803x)，其中一个文件将位于 Code Composer Studio™ 项目中。此文件负责 F280xx 器件的一次性初始化和配置，并且包括诸如设置时钟、PLL 和 GPIO 等功能。

快速 ISR 包含一个文件：

**BridgelessPFC-
DPL-ISR.asm**

此文件包含所有时间关键的“控制类型”代码。此文件有一个初始化部分 (一次执行) 和一个运行时部分，此部分以用来触发它的基于 PWM 时间 (200kHz) 一半的速率 (100kHz) 执行。

慢速 ISR 包含一个文件：

SineAnalyzer.h

此文件包含用于计算 RMS 电压和输入线路电压频率的代码。此文件有一个初始化部分 (一次执行) 和一个运行时间部分，此部分以 10kHz 的速率运行。

从快速 ISR 架构“调用”电源库函数（模块）。

库模块或许具有一个 C 语言和一个汇编组件。在此项目中，使用了 7 个库模块。可以在表 2-1 中找到 C 和相应的组件模块名称。

表 2-1. 库模块

C 配置函数	ASM 初始化宏	ASM 运行时间宏
PWM_1ch_UpDwnCnt_Cnf.c	PWMDRV_1ch_UpDwnCnt_INIT n	PWMDRV_1ch_UpDwnCnt n
ADC_SOC_Cnf.c	ADCDRV_1ch_INIT m,n,p,q	ADCDRV_1ch m,n,p,q
	PFC_InvRmsSqr_INIT n	PFC_InvRmsSqr n
	MATH_EMAVG_INIT n	MATH_EMAVG n
	PFC_BL_ICMD_INIT n	PFC_BL_ICMD n
	CNTL_2P2Z_INIT n	CNTL_2P2Z n

此组件模块也可以像图 2-2 中那样以图形方式表示。

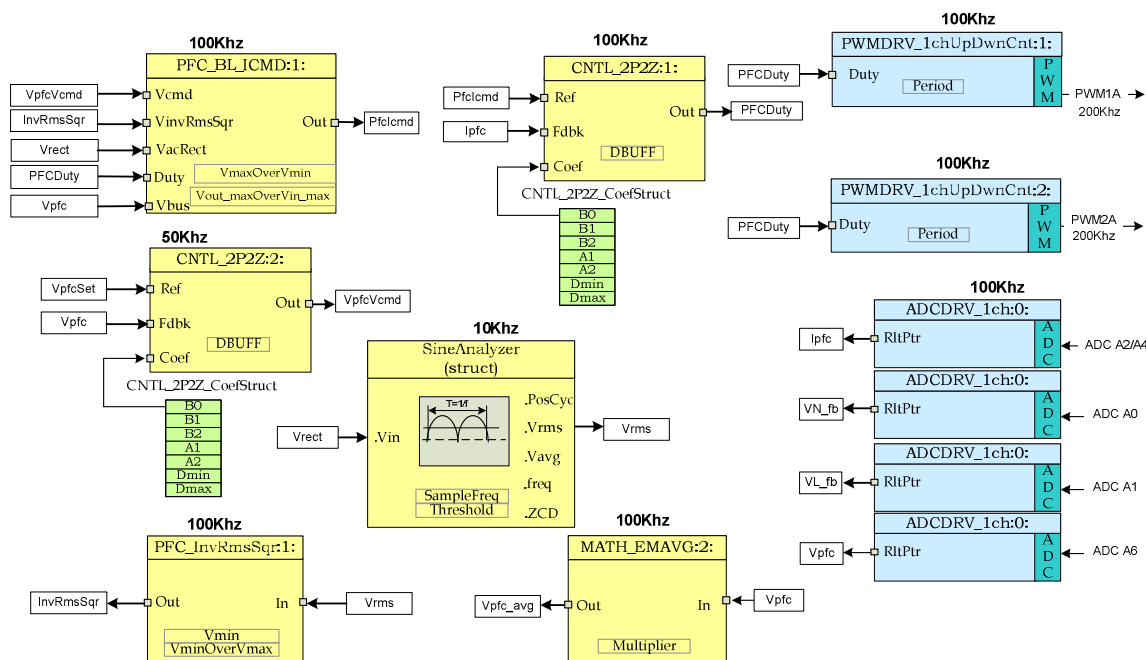


图 2-2. 软件区块

请注意对图 2-2 中的模块使用的颜色编码。‘深蓝色’的区块代表 C2000 控制器上的片载硬件模块。‘蓝色’区块是这些模块的驱动程序。‘黄色’区块是在不同信号上执行的计算部分。用于电压和电流环路的控制器的形式为一个 2 极 2 零比较器。然而，也可以采用其他形式，例如 PI、PID、3 极 3 零或任何其他适合该应用的控制器的形式。模块化的库结构使得可视化以及理解如图 2-3 中所示的整个系统软件流程变得非常方便。此外，还可以方便地使用和添加/删除各种功能。通过执行一个增量构建方法可在这个项目中充分证明这一事实。节 2.2 中对这一点进行了更详细的论述。

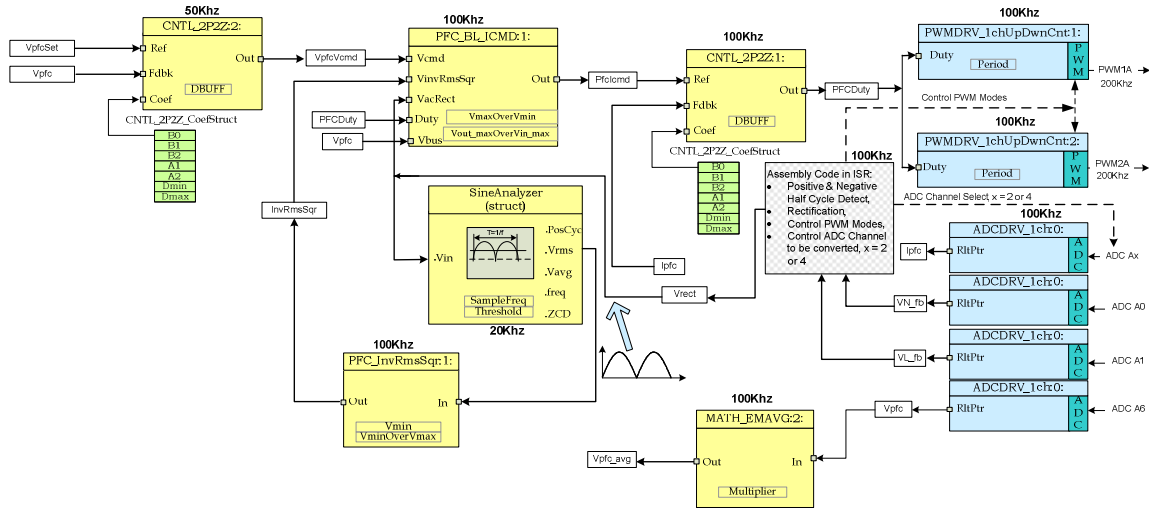


图 2-3. 软件控制流程

如节 1.1 中所述，BL PFC 系统由两个反馈环路控制。外部电压环路调节直流总线电压，而更快速的内部电流环路波则形成输入电流，以保持较高的输入功率因数。图 2-3 还提供了软件模块的执行速率。例如，电流控制器的执行速率为 100kHz (PWM 开关频率的一半)，而电压控制器的执行速率为 50kHz。

2.2 增量构建

此项目被分成 3 个增量构建。这种方法为用户提供了一种逐步熟悉软件并了解它如何与 BL PFC 硬件交互的方法。这个方法还简化了调试任务以及电路板测试。

此构建选项显示如下。若要选择特定的构建选项，请将 *BridgelessPFC-Settings.h* 文件中的宏 INCR_BUILD 设置为相应的构建选项，如下所示。一旦选择了构建选项，通过选择 rebuild-all 编译器选项来编译整个项目。下一章将提供运行每个构建选项的更多详细信息。

表 2-2. PFC 的增量构建选项

INCR_BUILD = 1	对升压操作和 ADC 反馈进行开环检查 (检查感应电路)
INCR_BUILD = 2	BL-PFC 的电压开环和电流闭环控制
INCR_BUILD = 3	BL PFC 的完整电压环和电流环控制

3 运行增量构建的流程

与这个受 C2x 控制的 BL PFC 系统相关的所有软件文件，即，主源文件、ISR 汇编文件和 C 框架的项目文件都位于目录 `...\\controlSUITE\\development_kits\\BLPFC_v1.0\\BLPFC` 中。此软件中包含的项目用于 CCSv4。

CAUTION

在电路板上会有高压出现。它应该只在实验室环境中由经验丰富的电源专业人员处理。要安全地评估此电路板，应使用隔离式交流电源为器件供电。在向电路板施加交流电源之前，应将电压表和**适当的电阻负载（仅适当的电阻负载）**连接到输出端。当交流电源关闭时，这将使总线电容器迅速放电。电路板未经电子负载测试，因此不应与此类负载一起使用。由于未在电路板上采取输出过流保护措施，所以，用户应想方设法防止发生任何输出短路的情况。**BLPFC 电路板应始终以 110Vac (60Hz) 启动。**电路板启动并运行后，可以将输入电压更改为规格范围内的任何其他电压。

请按照以下各节所列步骤来构建和运行 PFC 软件中包含的示例。

3.1 构建 1：通过 ADC 测量进行开环升压

3.1.1 构建 1 目标

这个构建的目标在于：

1. 评估 BL PFC PWM 和 ADC 软件驱动程序模块
2. 验证 MOSFET 栅极驱动电路、电压和电流感应电路
3. 熟悉 Code Composer Studio (CCS) 的操作。

在此构建下，系统以开环模式运行，因此测得的 ADC 值仅用于电路验证和仪表目的。接下来将解释构建和运行项目所需的步骤。

3.1.2 构建 1 概述

构建 1 中的软件已配置，用户可以通过在示波器上查看相关波形和观察占空比变化对 PFC 输出电压的影响来快速评估 PWM 驱动程序模块。用户可以通过 CCS 监视窗口调整 PWM 占空比。用户还可以通过查看监视窗口中的 ADC 采样数据来评估 ADC 驱动程序模块。

PWM 和 ADC 驱动程序宏实例在 `_DPL_ISR` 内部执行。图 3-1 显示了这个构建中所用的软件区块。两个 PFC 开关的两个 PWM 信号来自 ePWM 模块 1 和 2。ePWM1A 驱动一个 PFC 开关，而 ePWM2A 驱动另一个开关。

感应并反馈给 MCU 的量包括：(1) 线电压和中性点电压 (V_{L_fb} 、 V_{N_fb})，(2) 两个 PFC 开关电流 (I_{sw1} 、 I_{sw2})，在适当的线路半周期完成对每一种的采样，然后保存为 I_{pfc} ，以及 (3) 直流总线电压 (V_{pfc})。这些量由 ADC 驱动器模块读取并在图 3-1 中显示。ADC 驱动器模块将 12 位 ADC 结果转换为一个 32 位 Q24 值。ISR 中的几行代码实现了对输入交流线路半周期（正半周期和负半周期）的检测，并选择合适的 PFC 开关电流作为 I_{pfc} 。这些代码行还配置适当的 PWM 输出（PWM1A、PWM2A），使其在 PWM 模式或强制开启模式下运行，具体取决于输入交流线路半周期。

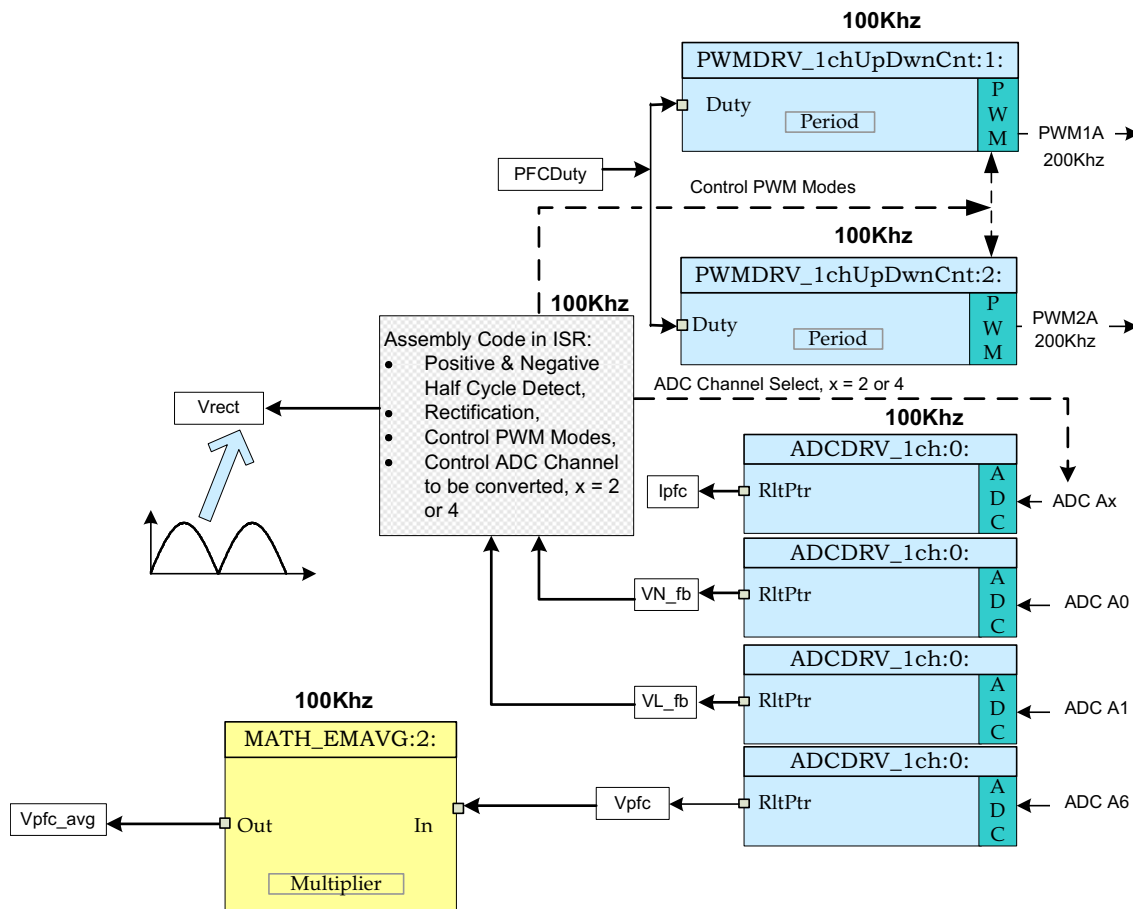


图 3-1. 构建 1 软件区块

PWM 信号以 200kHz 的频率产生，周期为 $5\mu\text{s}$ 。当控制器运行频率为 60MHz 时，ePWM1 时基计数器的一个计数对应的值为 16.6667ns。这说明， $5\mu\text{s}$ 的 PWM 周期相当于 300 个时基计数器计数 (TBCNT1、TBCNT2)。ePWM1 和 ePWM2 模块被配置为在向上计数/向下计数模式下运行，如图 3-2 中所示。这意味着时基周期值 150 (周期寄存器值) 将给出 300 个计数 ($5\mu\text{s}$) 的总 PWM 周期值。

采样值代表 CCM (连续接通模式) 条件下的平均电感器电流，因此 PFC 开关电流在 PWM 工作脉冲的中点进行采样。在 DCM 条件下，这一采样的开关电流值也代表一小部分平均电感器电流。通过分析 DCM 和 CCM 状态下的相关波形，得出了采样开关电流与平均电感器电流之间的关系。

所有其他电压信号转换过程也在此时启动。图 3-2 中显示了这一操作。C2000 器件上 ADC 和 PWM 模块的灵活性可实现精准且灵活的 ADC 转换启动。在这种情况下，ePWM1 作为一个时基使用，在 TBCNT1 达到零值时生成一个转换启动 (SOC) 触发信号。为了确保 ADC 结果的完整性，在这个点上执行一个假 ADC 转换。

图 3-2 还显示输入交流电压处于正半周期时的 PWM 输出。相应的开关电流 I_{sw1} 也显示为转换后的电流，并保存为用于 PFC 电流环路控制的 I_{pfc} 。

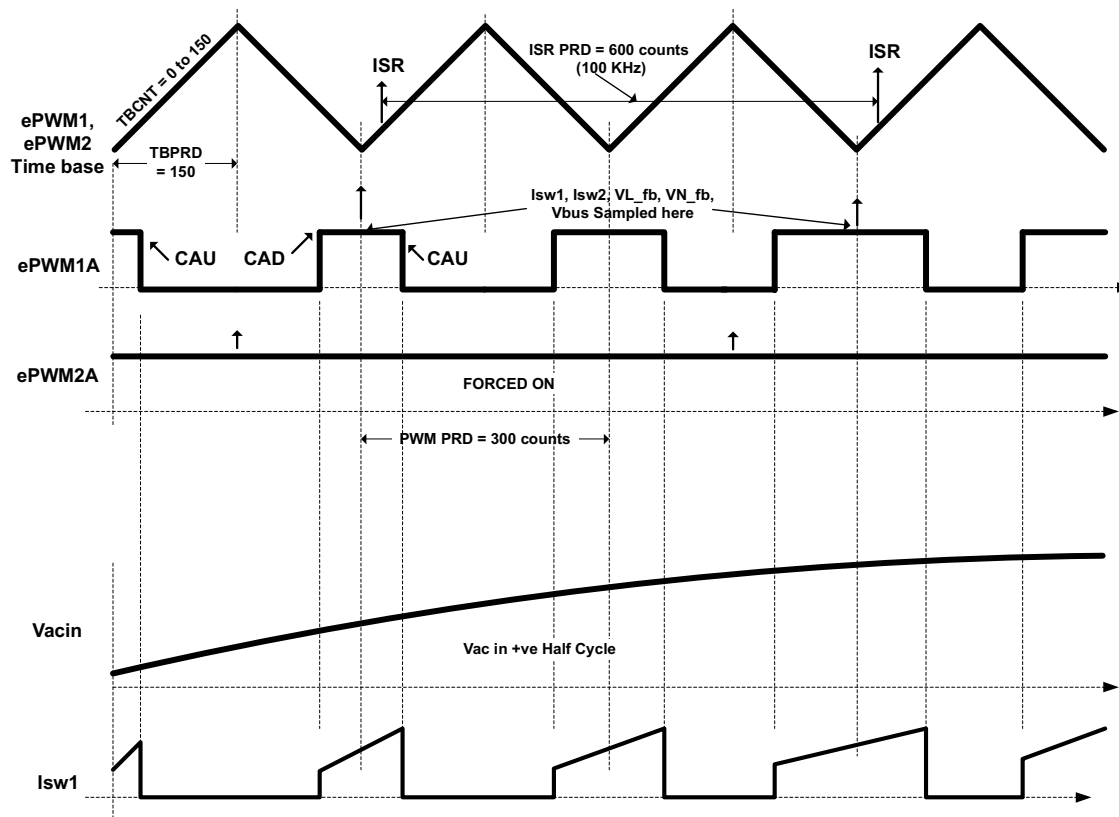


图 3-2. PWM 生成和 ADC 采样

在 CAU 事件上 (TBCNT1 = CMPA 并向上计数)，ePWM1A 输出被重置，而在 CAD 事件上 (TBCNT1 = CMPA 并向下计数)，设置 ePWM1A 输出。将 ePWM2A 设置为强制开启模式。ePWM1A 和 ePWM2A 的这种配置发生在输入电压的正半周期内。在输入电压的负半周期，两个 PWM 输出配置被切换，ePWM1A 进入强制开启模式，而 ePWM2A 在 PWM 模式下运行。

CMPA 值来自输入 “PFCDuty” (Q24 变量) 命令。

表 3-1 给出的示例为针对 150 的 TBPRD 值计算出的 CMPA 值。

表 3-1. 占空比参考值

PFCDuty (Q24 中为十六进制值)	$CMPA = (PFCDuty / (224 - 1)) * TBPRD$	% 占空比
0x00200000	18	12.5
0x00800000	75	50
0x00FFFFFF	150	100

将 ADC 模块配置为使用 ePWM1 的 SOCA，如此一来，SOCA 在 TBCNT1 = ZERO 事件发生时被触发。利用这个被触发的 SOCA 来完成所有转换。这 5 个 ADC 结果是通过执行 ADC 驱动程序模块从标记为 _DPL_ISR 的 100kHz ISR 中读取的。

在向上计数的 CMPB 匹配事件发生时，汇编中的 ISR (_DPL_ISR) 由 EPWM1 触发。CMPB 被设定为 80，如此一来，ISR 只在 ADC 转换完成后被触发。这是执行 PWMDRV_1ch_UpDwnCnt 宏，以及更新 PWM 比较阴影寄存器的地方。当下一个 TBCNT=ZERO 事件发生时，它们被载入到活动寄存器中。请注意，ISR 触发频率为图 3-2 中所示 PWM 开关频率的一半。

3.1.3 构建 1 保护

在软件中实现了针对此 BL PFC EVM 的过电压保护机制。将从 ADC 输入端感测到的直流总线输出电压与用户设置的过压保护阈值进行比较。默认的过电压阈值设置为 440V。该阈值参数在文件 *BridgelessPFC-Settings.h* 中标记为 `VBUS_OVP_THRSHLD`。在过电压条件下，使用 TZ（跳闸区）寄存器关闭 PWM 输出。C2000 器件触发机制的灵活性使根据不同触发事件采取不同操作成为可能。在这个项目中，两个 PWM 输出都将在发生跳闸事件时被驱动为低电平。两个输出都保持此状态，直到执行器件复位。

3.1.4 构建 1 流程

3.1.4.1 第 1.1 步：启动 CCS 和打开项目

请按照以下步骤执行此构建流程：

1. 将 USB 连接器连接至 F280xx controlCARD 以进行仿真。在 JP1 上为 12V 偏置电源加电。默认情况下，F280xx controlCARD 跳线配置为让器件从闪存引导（请参阅 F280xx controlCARD 文档）。更改这些跳线设置以允许在 CCS 控制下从 RAM 执行代码。
2. 启动 Code Composer Studio (CCS)。在 CCS 中，项目包含生成可在 MCU 硬件上运行的可执行输出文件 (.out) 所需的所有文件和构建选项。在菜单栏上，点击：Project Import Existing CCS/CCE Eclipse Project 并在 “Select root directory” 下，导航到并选择 `..\controlSUITE\development_kits\BLPFC_v1.0\BLPFC` 目录。确保选中 “Projects” 选项卡下的 BLPFC。点击 “Finish”。

此项目将调用所有必要的工具（编译器、汇编程序和链接器）来构建项目。

3. 在左侧的项目窗口中，点击项目左侧的加号符号 (+)。您的项目窗口将类似于 图 3-3：

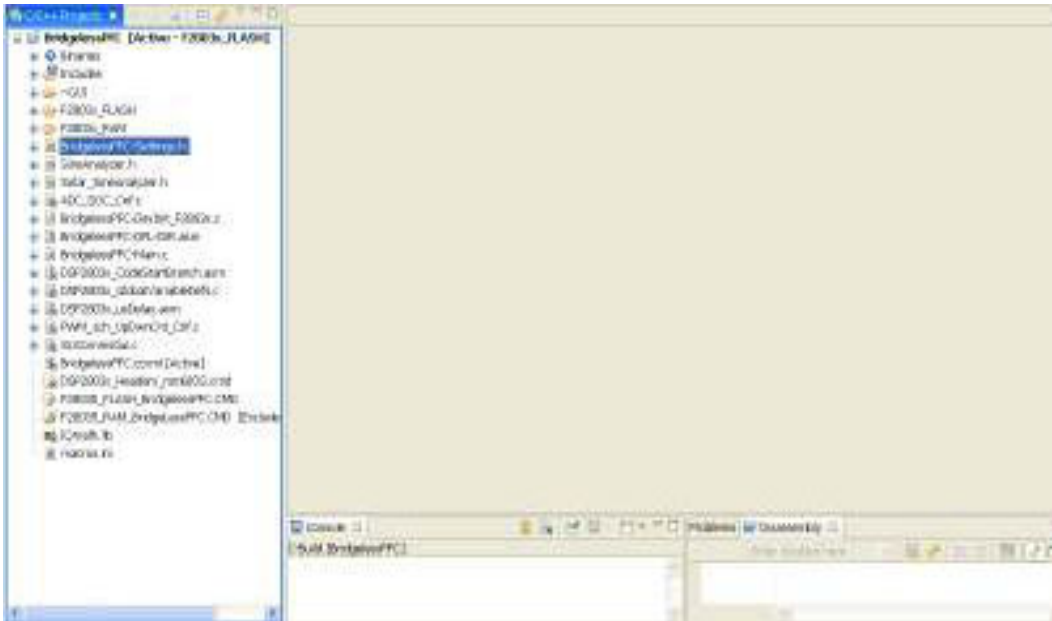


图 3-3. CCS 项目窗口

3.1.4.2 第 1.2 步：器件初始化、主菜单和 ISR 文件

NOTE

请勿对源文件进行任何更改 - 仅检查即可

1. 双击项目窗口中的文件名，打开并检查 *BridgelessPFC-DevInit_F2803x.c*。请注意，系统时钟、外设时钟预分频和外设时钟使能已经被设置。接下来，还请注意共用 GPIO 引脚也已经被配置。
2. 打开并检查 *BridgelessPFC-Main.c*。请注意对 *DeviceInit()* 函数的调用和其它变量的初始化。此外，请注意针对不同增量构建选项，ISR 初始化以及 (;) 环路所用背景的代码。
3. 找到并检查主文件内初始化代码下面专门用于构建 1 的以下代码。这是控制流程内连接 *PWMDRV_1ch_UpDwnCnt* 和 *ADCDRV_1CH* 区块的地方。

```
//-----
#if (INCR_BUILD == 1)    // Open Loop Debug only
//-----

// Lib Module connection to "nets"
//-----
// Connect the PWM Driver input to an input variable, Open Loop System
ADCDRV_1ch_Rlt1 = &Ipf;
ADCDRV_1ch_Rlt2 = &Vbus;
ADCDRV_1ch_Rlt3 = &VL_fb;
ADCDRV_1ch_Rlt4 = &VN_fb;

PWMDRV_1ch_UpDwnCnt_Duty1 = &DutyA;
PWMDRV_1ch_UpDwnCnt_Duty2 = &DutyA;

PWMDRV_1ch_UpDwnCnt_Duty4 = &Duty4A;

// Initialize the net variables
DutyA = _IQ24(0.1); // Variable initialized for open loop test
```

4. 找到并检查主文件内初始化代码下面的以下代码。这是 *PWMDRV_1ch_UpDwnCnt* 区块被配置和初始化的地方。这对于所有增量构建而言是通用的。这个 PWM 驱动器模块负责输入达 300 的总 PWM 周期值，并在内部计算出达 150 的周期寄存器值。

```
#define period 300 //300 cycles -> 200KHz @60MHz CPU
#define period_instr_pwm 120 //512 cycles -> 117k @60MHz CPU

// Configure PWM1 for 200KHz switching Frequency
PWM_1ch_UpDwnCnt_CNF(1, period, 1, 0);
// Configure PWM2 for 200KHz switching Frequency
PWM_1ch_UpDwnCnt_CNF(2, period, 0, 0);

// Configure PWM4 for Instrumentation; 1Mhz switching Frequency
PWM_1ch_UpDwnCnt_CNF(4, period_instr_pwm, 0, 0);
```

5. 还要找到并检查主文件内初始化代码下面的以下代码。这是 ADCDRV_1CH 区块被配置和初始化的地方。这对于所有增量构建而言也是通用的。

```
#define IpfcR   AdcResult.ADCRESULT1      //Q12
#define VbusR   AdcResult.ADCRESULT2      //Q12
#define VL_fbR   AdcResult.ADCRESULT3      //Q12
#define VN_fbR   AdcResult.ADCRESULT4      //Q12

// ADC Channel Selection for C2000EVM
ChSel[0] = 2;      // Dummy read for first
ChSel[1] = 2;      // A2 - IpfcA
ChSel[2] = 6;      // A6 - Vbus
ChSel[3] = 1;      // A1 - VL_fb
ChSel[4] = 0;      // A0 - VN_fb

// ADC Trigger Selection
TrigSel[0] = ADCTRIG_EPWM1_SOCa;      // ePWM1, ADCSOCA
TrigSel[1] = ADCTRIG_EPWM1_SOCa;      // ePWM1, ADCSOCA
TrigSel[2] = ADCTRIG_EPWM1_SOCa;      // ePWM1, ADCSOCA
TrigSel[3] = ADCTRIG_EPWM1_SOCa;      // ePWM1, ADCSOCA
TrigSel[4] = ADCTRIG_EPWM1_SOCa;      // ePWM1, ADCSOCA

// Configure ADC
ADC_SOC_CNF(ChSel, TrigSel, ACQPS, 17, 0);

// Configure ePWMs to generate ADC SOC pulses
EPwm1Regs.ETSEL.bit.SOCaEN = 1;      // Enable ePWM1 SOCA pulse
EPwm1Regs.ETSEL.bit.SOCaSEL = ET_CTR_ZERO;      // SOCA from ePWM1 Zero event
EPwm1Regs.ETPS.bit.SOCAPRD = ET_1ST;      // Trigger ePWM1 SOCA on every event
```

6. 打开并检查 *BridgelessPFC-DPL-ISR.asm*。请注意构建 1 下的 _DPL_Init and _DPL_ISR 段。这分别是 PWM 和 ADC 驱动器宏示例完成初始化和运行时间结束的地方。

3.1.4.3 第 1.3 步：构建和加载项目

- 在 *BridgelessPFC-Settings.h* 文件中，选择 1 级增量构建选项。

NOTE

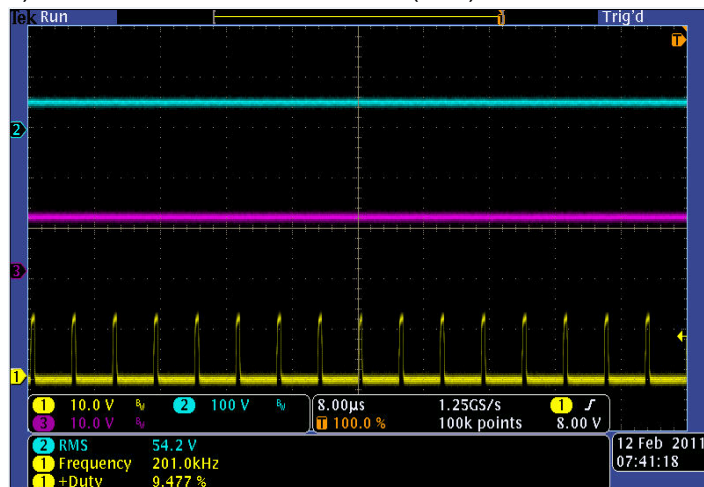
每次在 BridgelessPFCSettings.h 中更改增量构建选项后，务必选择“Rebuild All”。

- 点击“Project”->“Rebuild All”按钮并观察构建窗口中工具的运行情况。
- 点击“Target”->“Debug Active Project”。如果尚未选择，CCS 将要求您打开一个新的目标配置文件。如果已经为这个连接创建了一个有效的目标配置文件，您可以转到第 14 步。在 New target Configuration 窗口内为您将操作的目标键入以 .ccxml 为后缀的文件名（例如：xds100-F28035.ccxml）。选中“Use shared location”并点击“Finish”。
- 在打开的 .ccxml 文件中，选择 Connection 作为“Texas Instruments XDS100v2 USB Debug Probe”，然后在器件下，向下滚动并选择“TMS320F28035”。点击“Save”。
- 点击“Target”->“Debug Active Project”。将项目配置选为 F2803x_FLASH。此程序将被载入闪存。您现在应该处于 Main() 的起始位置。

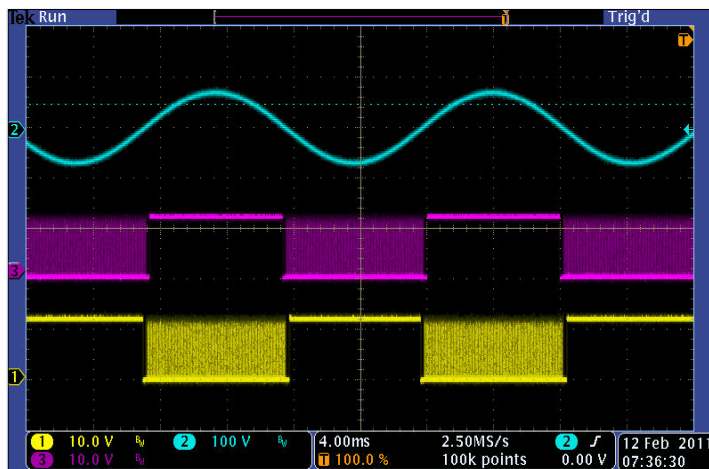
NOTE

仔细观察输出电压，该值不应超过电路板的最大额定电压。

- 在监视视图中添加诸如 V_{bus} 、 V_{L_fb} 和 V_{N_fb} 等其他变量，并验证不同的 ADC 结果。对于交流电压输入，感测到的线电压和中性点电压 (V_{L_fb} 、 V_{N_fb}) 将在监视窗口中持续变化。因此，为了验证 ADC 读数以及线电压和中性点电压感测电路，用户可以施加直流输入电压 (20~200V) 而不是交流电压 (如上面第 20 步中所述)。在这种情况下，PFC 级将暂时在纯直流/直流升压模式下运行。
- 当输出直流总线负载为 8 千欧且将占空比设置为约 10% 时，以下示波器捕获为两个 PWM 输出 (Ch1 和 Ch3) 和直流输入电压 (Ch2)。在这种情况下，输入电压始终为正 (在这种情况下为 +50V)，因此 PWM1A 将处于 PWM 模式 (Ch1)，PWM2A 将处于强制开启模式 (Ch3)。测得的 PWM 频率也为 200kHz。



- 当输出直流总线负载为 8 千欧且将占空比设置为约 10% 时，以下示波器捕获为两个 PWM 输出 (Ch1 和 Ch3) 和约 50V 的交流输入电压 (Ch2)。交流输入电压 PWM1A 和 PWM2A 在 PWM 模式和强制开启模式之间交替切换。



- 尝试不同的占空比值，并观察相应的 ADC 结果。以小步长增加占空比值。要一直小心地观察输出电压，这个值不应超过电路板的处理能力。诸如 PWM 栅极驱动信号、输入电压和电流以及输出电压等的不同波形也可使用一个示波器进行探测和验证。探测这些高压信号时，请采取适当的安全措施。

10. 在实时模式下，完全停止 MCU 需要执行两个步骤。在交流输入关闭的情况下，等待直流总线电容器完全放电。首先，通过使用工具栏上的“Halt”按钮，或者使用“Target”->“Halt”来暂停处理器。然后使 MCU 退出实时模式。最后复位 MCU。
11. 您可以选择让 Code Composer Studio 继续运行以用于下一个练习，或者选择将其关闭。
12. 练习结束

3.2 构建 2：具有闭合电流环路的 BL PFC

3.2.1 构建 2 目标

此构建的目的是验证 BL PFC 在闭合电流环路模式下的运行情况。

3.2.2 构建 2 概述

图 3-5 所示为在这个构建中使用的软件区块。请注意，构建 1 图 (图 3-1) 中添加了 4 个附加软件区块，以实现此闭合电流环路系统。正弦分析器区块负责计算输入电压的 RMS 电压和频率。PFC InvRmsSqr 区块负责计算 RMS 输入电压平方的倒数。此计算值与整流电压 (Vrect)、感应直流总线电压 (Vpfc) 和 PFC PWM 占空比一起用于第 3 个软件区块 PFC_BL_ICMD，以生成 PFC 电流控制环路的参考电流指令 PfcIcmd。PFC_BL_ICMD 区块使用第 5 个输入 VpfcVcmd 来控制参考电流幅度指令。此软件构建仅实现 PFC 电流环路 (开路电压环路)，因此需要从 CCS 窗口更改此参数 VpfcVcmd，以便调整参考电流的大小，从而调整 PFC 总线电压。采用双极点-双零点 (2p2z) 控制器实现电流控制环路。这是图 3-5 中显示为 CNTL_2P2Z:1 的第四个软件区块。根据控制环路的要求，还可以使用其他控制块，如 PI 或 3p3z 控制器。

如图 3-5 中所示，电流环路区块以 100kHz 的频率运行。CNTL_2P2Z 是一种二阶补偿器，由 IIR 滤波器结构实现。这个函数与任何外设无关，因此就不需要调用 CNF 函数。

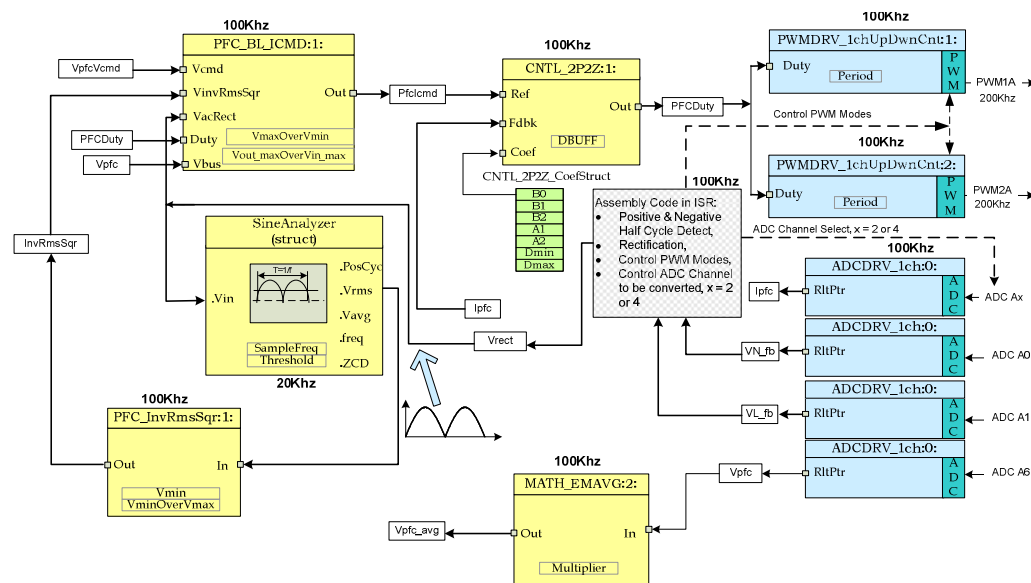


图 3-5. 构建 2 软件区块

这个 2p2z 控制器需要 5 个控制系数。这些系数和控制器的钳位输出被存储为一个名为 CNTL_2P2Z_CoefStruct1 的结构元素。如果系统需要多个环路，CNTL_2P2Z 区块可多次被实例化。每个实例可具有单独的系数集合。当前环路的 CNTL_2P2Z 实例使用被存储为结构 CNTL_2P2Z_CoefStruct1 的元素的系数。这样，具有不同结构 CNTL_2P2Z_CoefStruct2 的 CNTL_2P2Z 的第二个实例可用于 PFC 电压环路控制，我们将在构建 3 节 3.3 中介绍相关内容。

可通过修改结构 **CNTL_2P2Z_CoefStruct1** 内的 **B0** , **B1** , **B2** , **A1** 和 **A2** 值来直接更改控制器系数。或者, **2p2z** 控制器可表示为 **PID** 形式, 并且可通过改变 **PID** 系数来改变此系数。下文给出了五个控制器系数与三个 **PID** 增益之间的关系式。对于电流环路, 这些 **P**、**I** 和 **D** 系数分别被命名为: **Pgain_I**、**Igain_I** 和 **Dgain_I**。对于构建 3 中使用的电压环路, 这些系数分别被命名为: **Pgain_V**、**Igain_V** 和 **Dgain_V**。这些系数采用 **Q26** 格式。为了从 **GUI** 环境 (或从 **CCS** 监视视图) 更改这些系数, 它们被进一步调整为 0 到 9999 之间的值。

此补偿器区块 (**CNTL_2P2Z**) 有一个基准输入和一个反馈输入。这个被标记为 **Fdbk** 的反馈输入来自 **ADC**。被标记为 **Ref** 的参考输入来自前文提及的 **PFC_BL_ICMD** 区块。**CNTL_2P2Z** 的 **z** 域传递函数由以下内容给出 :

$$\frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (1)$$

PID 控制器的递归形式由不同的等式给出 :

$$u(k) = u(k-1) + b_0 e(k) + b_1 e(k-1) + b_2 e(k-2) \quad (2)$$

其中

$$b_0 = K_p + K_i + K_d$$

$$b_1 = -K_p + K_i - 2K_d$$

$$\bullet \quad b_2 = K_d$$

这个 **PID** 的 **z** 域传递函数为 :

$$\frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - z^{-1}} \quad (4)$$

与常规形式相比, 我们可以看到 **PID** 是 **CNTL_2P2Z** 控制的特例, 其中 :

$$a_1 = -1 \quad \text{and} \quad a_2 = 0 \quad (5)$$

MATH_EMAVG (指数平均数指标) 区块负责计算输出直流总线电压的平均值。此区块的输出用于检测过电压状态, 然后进行 **PWM** 关断。

3.2.3 构建 2 流程

3.2.3.1 第 2.1 步：构建和加载项目

请按照以下步骤执行此构建流程：

1. 完全按照构建 1 (节 3.1.4) 中的第 1 步到第 7 步进行操作，不过在第 6 步中应选择构建 2 选项而不是构建 1。然后按照下文所示完成第 6 步：
2. 找到并检查主文件内初始化代码下面专门用于构建 2 的以下代码。这是控制流程中与构建 2 相关的所有软件区块的连接位置。

```
#if (INCR_BUILD == 2) // Closed Current Loop Bridgeless PFC, Open Volt Loop
//-----
// Lib Module connection to "nets"
ADCDRV_1ch_Rlt1 = &Ipfc;
ADCDRV_1ch_Rlt2 = &Vbus;
ADCDRV_1ch_Rlt3 = &VL_fb;
ADCDRV_1ch_Rlt4 = &VN_fb;
//connect the 2P2Z connections, for the inner Current Loop, Loop1
CNTL_2P2Z_Ref1 = &PFCIcmd;
CNTL_2P2Z_Out1 = &DutyA; //Comment to open the curr loop.Then specify open loop duty to boost bus
CNTL_2P2Z_Fdbk1= &Ipfc;
CNTL_2P2Z_Coeff1 = &CNTL_2P2Z_CoeffStruct1.b2;
// Math_avg block connections - Instance 2
MATH_EMAVG_In2=&Vbus;
MATH_EMAVG_Out2=&VbusAvg;
MATH_EMAVG_Multiplier2=_IQ30(0.00025);
// INV_RMS_SQR block connections
VrectRMS = (sine_mainsV.Vrms)<< 9;//Q15 --> Q24, (sine_mainsV.Vrms) is in Q15
PFC_InvRmsSqr_In1=&VrectRMS;
PFC_InvRmsSqr_Out1=&VinvSqr;
PFC_InvRmsSqr_VminOverVmax1=_IQ30(0.1956); // 80V/409V
PFC_InvRmsSqr_Vmin1=_IQ24(0.1956); // 80V/409V
// PFC_BL_ICMD block connections
PFC_BL_ICMD_Vcmd1 = &VbusVcmd;
PFC_BL_ICMD_VinvSqr1=&VinvSqr;
PFC_BL_ICMD_VacRect1=&Vrect;
PFC_BL_ICMD_Out1=&PFCIcmd;
PFC_BL_ICMD_VmaxOverVmin1=_IQ24(2.00); // 3.5625 <=> 285V/80V
PFC_BL_ICMD_Vpfc1 = &Vbus;
PFC_BL_ICMD_Duty1 = &DutyA;
PFC_BL_ICMD_VoutMaxOverVinMax1 = _IQ24(1.266);//(1.272);//VoutMax=521.4V, VinMax=409.8V(peak)= 289

PWMDRV_1ch_UpDwnCnt_Duty1 = &DutyA;
PWMDRV_1ch_UpDwnCnt_Duty2 = &DutyA;
```

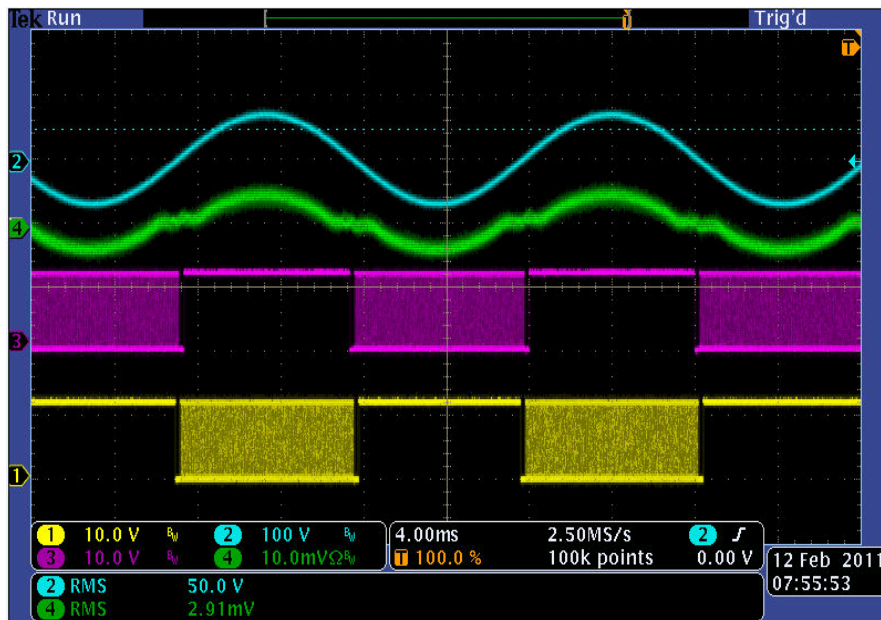
3. 打开并检查 *BridgelessPFC-DPL-ISR.asm*。请注意构建 2 下的 `_DPL_Init` and `_DPL_ISR` 段。这是构建 2 下的全部宏实例分别完成初始化和运行时间的位置。
4. 在 *BridgelessPFC-Settings.h* 文件中，选择 2 级增量构建选项。然后，为了运行此代码，请按照构建 1 中的第 10 步到第 17 步进行操作。当所有这些步骤完成时，您现在应该在 `Main()` 的起始位置。

NOTE

每次在 *BridgelessPFCSettings.h* 中更改增量构建选项后，务必选择“Rebuild All”。

5. 使用 <F8> 键来运行代码，或者使用工具栏上的“Run”按钮，或者使用菜单栏上的“Target”->“Run”。
6. 在监视视图中，添加变量 *VpfcVcmd*，并将其设置为 0.05 (= Q24 中的 838861)。这个变量设定针对电流控制环路的基准电流指令的大小。
7. 在 PFC 系统的直流输出上施加一个适当的电阻负载。例如，可以使用一个额定功率为 40W 的 8.0K Ω 电阻。这将在 200V 总线电压上提供一个 5W 的负载。

8. 从隔离式交流电源缓慢地向电路板提供交流电源。当输入电压缓慢升高至 50V rms 时，监测直流总线电压。总线电压当前应该在 200V 左右。调整 *VpfcVcmd* 的值以将总线电压设置为大约 200V。使用带有电压和电流探针的示波器观察输入电压、输入电流和 PWM 输出。使用 50V rms 输入、8 千欧电阻负载，并将总线电压设置为 200V 时，您会看到以下波形。在此处，Ch1 和 Ch3 显示了 PWM 输出。Ch2 是输入电压，而 Ch4 是输入电流。当电流环路闭合时，输入电流应具有与输入电压相同的形状和良好的功率因数。



9. 略微增加 *VpfcVcmd* 的值（步长为 0.01），并观察总线电压稳定至一个较高值。增大 *VpfcVcmd* 会增加电流参考信号的幅度，而 PFC 电压环路是开路，因此总线电压将升高。因而，请采取防护措施并将过压保护阈值设定为一个少于 350V 的值。此阈值参数在文件 *BridgelessPFC-Settings.h* 中标记为 *VBUS_OVP_THRSHLD*。现在，改变输入电压或负载电阻，观察 PFC 在电流控制环路下的运行情况。
10. 略微增加 *VpfcVcmd* 的值（步长为 0.01），并观察总线电压稳定至一个较高值。增大 *VpfcVcmd* 会增加电流参考信号的幅度，而 PFC 电压环路是开路，因此总线电压将升高。因而，请采取防护措施并将过压保护阈值设定为一个少于 350V 的值。此阈值参数在文件 *BridgelessPFC-Settings.h* 中标记为 *VBUS_OVP_THRSHLD*。现在，改变输入电压或负载电阻，观察 PFC 在电流控制环路下的运行情况。
11. 练习结束

3.3.3 构建 3 流程

3.3.3.1 第 3.1 步：构建和加载项目

请按照以下步骤执行此构建流程：

1. 完全按照构建 1 (节 3.1) 中的第 1 步到第 7 步进行操作，不过在第 6 步中应选择构建 3 选项而不是构建 1。然后按照以下步骤完成第 6 步
2. 找到并检查主文件内初始化代码下面专门用于构建 3 的下列代码。这是控制流程中与构建 3 相关的所有软件区块的连接位置。

```
#if (INCR_BUILD == 3)    // Closed Current Loop & closed volt loop Bridgeless PFC
//-----
// Lib Module connection to "nets"
ADCDRV_1ch_Rlt1 = &Ipfc;
ADCDRV_1ch_Rlt2 = &Vbus;
ADCDRV_1ch_Rlt3 = &VL_fb;
ADCDRV_1ch_Rlt4 = &VN_fb;
//connect the 2P2Z connections, for the inner Current Loop, Loop1
CNTL_2P2Z_Ref1 = &PFCicmd;
CNTL_2P2Z_Out1 = &DutyA;
CNTL_2P2Z_Fdbk1= &Ipfc;
CNTL_2P2Z_Coef1 = &CNTL_2P2Z_CoefStruct1.b2;
//connect the 2P2Z connections, for the outer Voltage Loop, Loop2
CNTL_2P2Z_Ref2 = &VbusTargetSlewed;
CNTL_2P2Z_Out2 = &VbusVcmd;
CNTL_2P2Z_Fdbk2= &Vbus;
CNTL_2P2Z_Coef2 = &CNTL_2P2Z_CoefStruct2.b2;
// Math_avg block connections - Instance 2
MATH_EMAVG_In2=&Vbus;
MATH_EMAVG_Out2=&VbusAvg; //Average PFC bus volt calculated for OV protection
MATH_EMAVG_Multiplier2=_IQ30(0.00025);
// INV_RMS_SQR block connections
VrectRMS = (sine_mainsV.Vrms)<< 9; //Q15 --> Q24, (sine_mainsV.Vrms) is in Q15
PFC_InvRmsSqr_In1=&VrectRMS;
PFC_InvRmsSqr_Out1=&VinvSqr;
PFC_InvRmsSqr_VminOverVmax1= _IQ30(0.1956);    // 80V/409V
PFC_InvRmsSqr_Vmin1=_IQ24(0.1956);
// PFC_BL_ICMD block connections
PFC_BL_ICMD_Vcmd1 = &VbusVcmd;
PFC_BL_ICMD_VinvSqr1=&VinvSqr;
PFC_BL_ICMD_VacRect1=&Vrect;
PFC_BL_ICMD_Out1=&PFCicmd;
PFC_BL_ICMD_VmaxOverVmin1=_IQ24(2.00);    // 3.6225 <=> 289.8V/80V,
PFC_BL_ICMD_Vpfc1 = &Vbus;
```

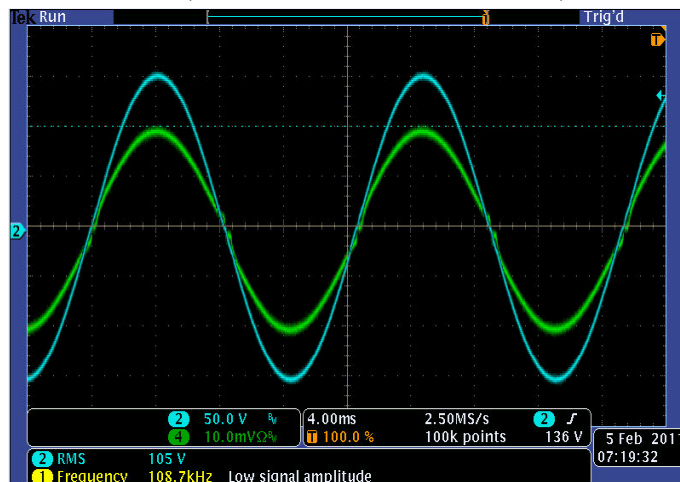
3. 打开并检查 *BridgelessPFC-DPL-ISR.asm*。请注意构建 3 下的 `_DPL_Init` and `_DPL_ISR` 段。这是构建 3 下的全部宏实例分别完成初始化和运行时间的位置。
4. 在 *BridgelessPFCSettings.h* 文件中，选择 3 级增量构建选项。然后，为了运行此代码，请按照构建 1 中的第 10 步到第 17 步进行操作。当所有这些步骤完成时，您现在应该在 `Main()` 的起始位置。

NOTE

每次在 *BridgelessPFC-Settings.h* 中更改增量构建选项后，务必选择“Rebuild All”

5. 使用 <F8> 键来运行代码，或者使用工具栏上的“Run”按钮，或者使用菜单栏上的“Target”->“Run”。
6. 在监视视图中，添加变量 `VbusTargetSlewed` 和 `Vbus`，并将 Q-format 设置为 Q24。这些变量分别代表参考总线电压和反馈总线电压。当应用交流电源时，PFC 启动，这些变量将缓慢增加到设定值。
7. 在 PFC 系统的直流输出上施加一个适当的电阻负载。例如，可以使用一个额定功率为 40W 的 8 千欧电阻。这将在 400V 总线电压上提供一个 20W 的负载。

8. 配置隔离式交流电源以输出 110V, 60Hz 的交流电压。使用电压表来监测直流总线电压。开启 110Vrms 的交流电源输出。当直流总线电压达到 100V 时, 浪涌继电器将启动, 总线电压将缓慢增加至 400V。请注意, 当 Q-Format 设置为 Q24 时, 监视窗口上的 VbusTargetSlewed 和 Vbus 变量显示的值约为 0.7707 (=400/519)。Vbus 感应电阻器设置的最大总线电压约为 519V, 对应于 3.3V 的最大 ADC 输入。因此, 当实际总线电压为 400Vdc 时, 标准化或归一化值约为 0.7707。将 VbusTargetSlewed 调整为 0.732 (=380/519), 以将总线电压设置为约 380V。使用带有电压和电流探针的示波器观察输入电压、输入电流和 PWM 输出。使用 110V rms 输入、540 欧姆电阻负载, 并将总线电压设置为 400V 时, 您会看到以下波形。



- A. Ch2 = 输入电压
- B. Ch4 = 输入电流

图 3-7. 300W 时的 BL PFC 输入电压和电流波形

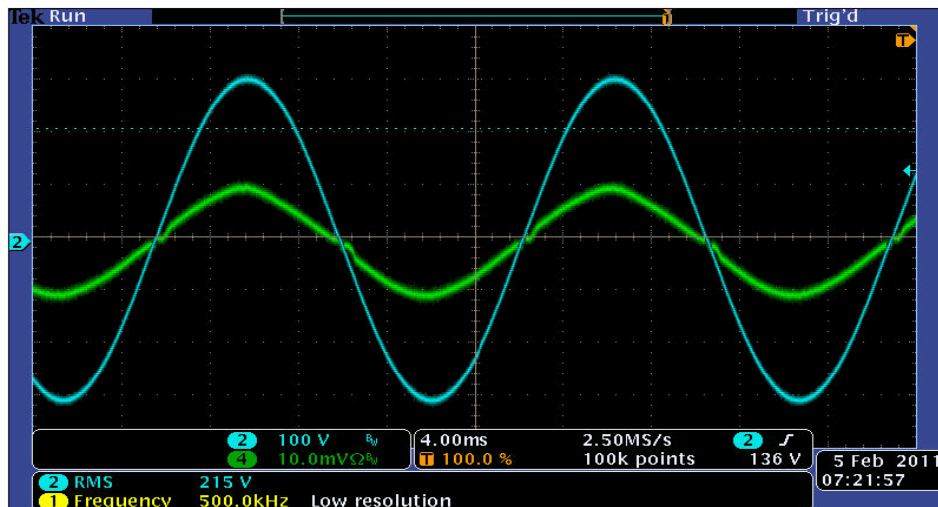
使用 110V rms 输入、1080 Ω 电阻负载, 并将总线电压设置为 400V 时, 您会看到以下波形。



图 3-8. 150W 时的 BL PFC 输入电压和电流波形

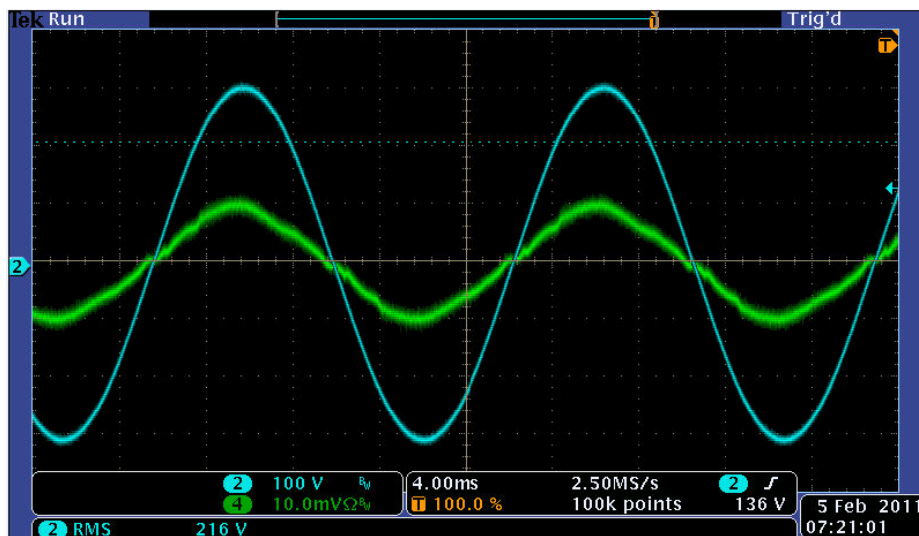
9. 改变输入电压 (90Vrms~240Vrms) 或负载电阻 (0~300W), 观察 PFC 在闭合电流和电压控制环路下的运行情况。
10. 按照 节 3.1 中的第 26 步和第 27 步进行操作, 以关闭电源并重置 MCU。
11. 练习结束。测试结果参见 节 4。

4 测试结果



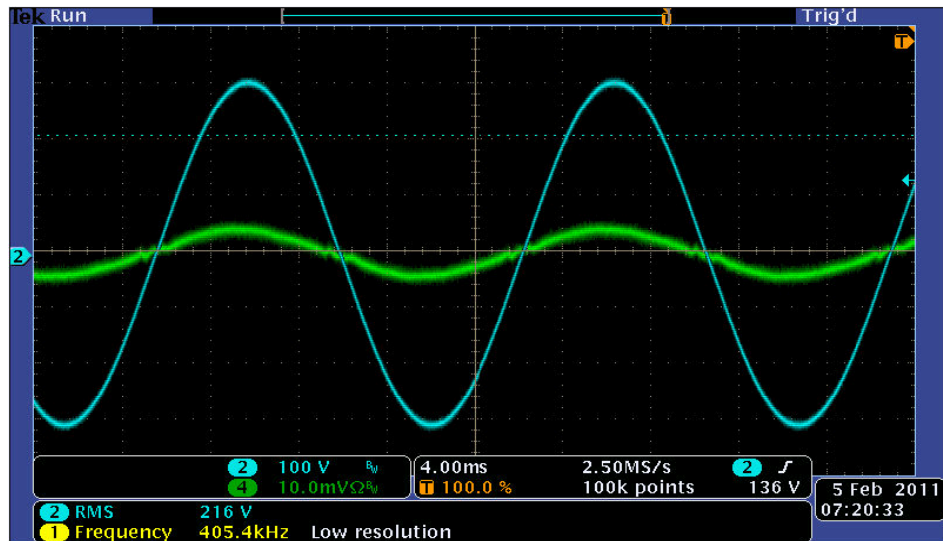
- A. Ch2 - V_{in}
- B. Ch4 - I_{in}
- C. $V_{rms} = 220V$
- D. $V_{bus} = 400V$

图 4-1. BL PFC 输入电压和电流波形 ($P_{out} = 300W$)



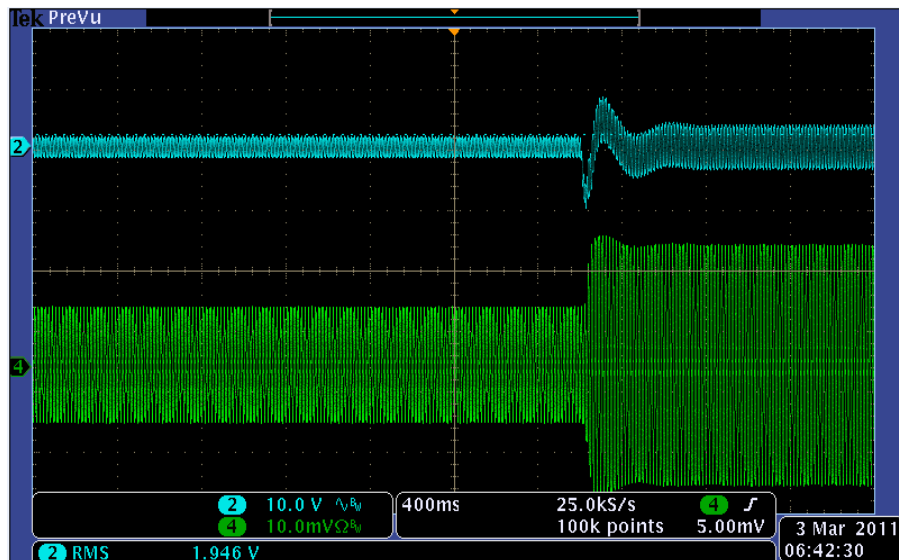
- A. Ch2 - V_{in}
- B. Ch4 - I_{in}
- C. $V_{rms} = 220V$
- D. $V_{bus} = 400V$

图 4-2. BL PFC 输入电压和电流波形 ($P_{out} = 150W$)



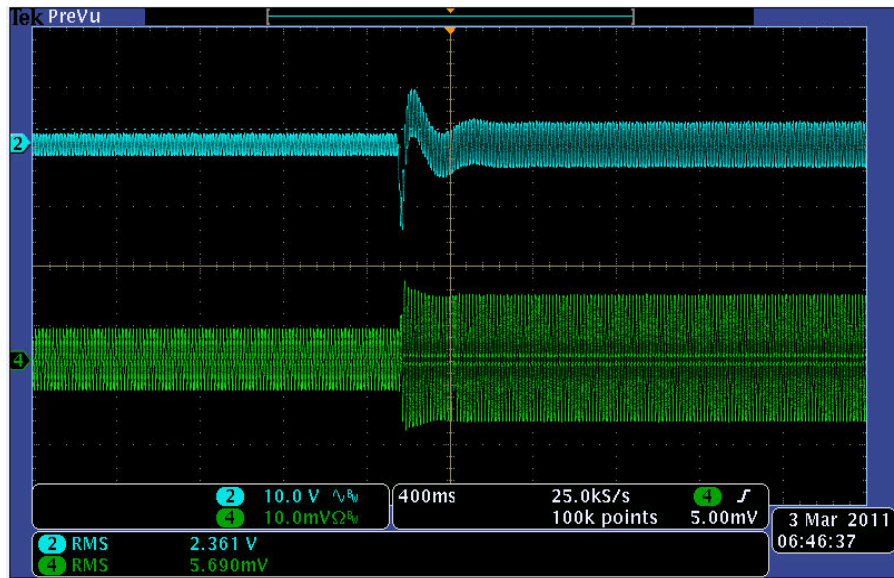
- A. Ch2 - V_{in}
- B. Ch4 - I_{in}
- C. $V_{rms} = 220V$
- D. $V_{bus} = 400V$

图 4-3. BL PFC 输入电压和电流波形 ($P_{out} = 60W$)



- A. Ch2 - V_{bus}
- B. Ch4 - I_{in}
- C. $V_{bus} = 400V$
- D. 负载阶跃 150W ~ 300W

图 4-4. BL PFC 直流总线负载瞬态响应 ($V_{rms} = 110V$)



- A. Ch2 - Vbus
- B. Ch4 - Iin
- C. Vbus = 400V
- D. 负载阶跃 150W ~ 300W

图 4-5. BL PFC 直流总线负载瞬态响应 (Vrms = 220V)

5 参考文献

如需更多信息，请参阅以下指南：

BLPFC-GUI-QSG 使用 GUI 界面快速演示 BLPFC EVM 的快速入门指南

..\controlSUITE\development_kits\BLPFC \~Docs\BLPFC-GUI-QSG.pdf

BLPFC_Rel-1.0-HWdevPkg 包含与 F28035 controlCARD 原理图和 BL PFC 原理图相关的各种文件的文件夹。

..\controlSUITE\development_kits\BLPFC\BLPFC_HWDevPkg

NOTE

此 PFC EVM 使用 F28035 controlCARD 而非 F28027 controlCARD，其原因是 F28027 controlCARD 中没有 RC 滤波器。在 F28035 controlCARD 上可感应两个 BL PFC 开关电流的两个 ADC 通道中都提供这种 RC 滤波器。然而，在 F28027 controlCARD 上，两个 ADC 通道中只有其中一个具有此 RC 滤波器。此 F28027 controlCARD 的修改版可轻松实现对此 BL PFC EVM 的完全控制。

6 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (May 2014) to Revision A (November 2020)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式。.....	3
• 整个文档都进行了更新，删除了对 Piccolo 的引用。.....	3

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司