

# 使用 AM263 的 10kW 三相三级 (T 型) 逆变器



Sri Vidya Gunturi, Salil Chellappan

## 摘要

本用户指南重点介绍了 AM263x 微控制器如何用于控制 TIDA-01606 基于 SiC 的双向三级、三相逆变器和 PFC 功率级参考设计。TIDA-01606 参考设计可帮助客户在光伏逆变器、电动汽车充电器、电池储能系统、UPS 和电动汽车车载充电器 (OBC) 中实现单向或双向交流/直流和直流/交流功率级。AM263x 可有效地提供经过优化的快速控制环路性能，以便收集数据并处理所收集的数据，还可在定义的时间范围内使用紧密集成的高分辨率 PWM (用于驱动) 更新系统，从而生成精确的占空比。

## 内容

1 引言.....	4
2 硬件要求.....	4
2.1 所需的硬件和测试仪器.....	4
2.2 设计中使用的微控制器资源.....	4
2.3 TIDA-01606 REV-6 的硬件更改.....	5
2.4 TMDSCNCD263 controlCARD™ 更改.....	6
3 软件.....	9
3.1 固件入门.....	9
3.2 SysConfig 设置.....	13
3.3 中断和实验结构.....	24
3.4 保护方案.....	29
3.5 CPU 负载.....	31
3.6 构建、加载和调试固件.....	31
4 实施优化.....	32
5 测试和结果.....	34
5.1 实验 1.....	34
5.2 测试逆变器运行情况.....	35
5.3 测试 PFC 运行情况.....	37
6 参考文献.....	46

## 插图清单

图 2-1. TIDA-01606 中的 R196.....	5
图 2-2. 用于 HSEC 引脚 92 配置的 I2C2 IO 扩展器.....	6
图 2-3. 用于 HSEC 引脚 92 配置的 I2C 多路复用器和多路信号分离器.....	6
图 2-4. AM263x controlCARD 中的 I2C0 MUX SEL.....	7
图 2-5. IO 扩展器的 I2C SysConfig 配置.....	7
图 2-6. L18 连接到 EPW21_B.....	8
图 3-1. 无引导模式.....	9
图 3-2. 软件方框图.....	10
图 3-3. Rampgen 模型.....	11
图 3-4. 控制 ISR 中的 SPLL 使用流程.....	11
图 3-5. $I_d$ 电流环路模型.....	12
图 3-6. 演示的文件夹结构.....	13
图 3-7. EPWM 时基子模块配置.....	14

图 3-8. EPWM 计数器比较子模块配置.....	14
图 3-9. 较高的 CMPA 值会产生高占空比.....	14
图 3-10. CMPA 值可改变 EPWM0 波形的占空比.....	15
图 3-11. EPWM 动作限定符子模块配置.....	15
图 3-12. EPWM 死区子模块配置.....	16
图 3-13. EPWM 事件触发器子模块配置.....	16
图 3-14. EPWM1 中断的 INT XBAR 配置.....	16
图 3-15. ISR2 的计时器配置.....	17
图 3-16. SDFM 配置.....	17
图 3-17. TIDA-01606 的电网侧和逆变器侧.....	18
图 3-18. ADC 配置.....	19
图 3-19. AM263x controlCARD 中的 ADC 和 DAC 基准开关.....	20
图 3-20. 选择 CMPSS 实例并启用模块.....	21
图 3-21. CMPSS 高电平比较器配置和阈值设置.....	21
图 3-22. EPWM X-BAR 配置.....	21
图 3-23. EPWM 数字比较子模块配置.....	22
图 3-24. EPWM 跳闸区域子模块配置.....	22
图 3-25. ECAP 配置.....	23
图 3-26. 输出 X-Bar 配置.....	23
图 3-27. 输入 X-Bar 配置.....	24
图 3-28. ISR1 中的实验 1.....	26
图 3-29. ISR1 中的实验 3.....	26
图 3-30. ISR1 软件流程方框图.....	27
图 3-31. 通过 ISR1 进行 PWM 更新.....	28
图 3-32. ISR2 软件流程方框图.....	28
图 3-33. 具有栅极驱动器故障检测功能的 TIDA-01606 方框图.....	29
图 3-34. 板保护方框图.....	29
图 3-35. 针对栅极驱动器故障检测的跳闸使能.....	30
图 4-1. 启用实验 3 时的 TCMA 和 TCMB 存储器消耗.....	32
图 4-2. 编译器优化设置.....	33
图 5-1. 实验 1 EPWM 输出波形.....	34
图 5-2. 使用 DAC 在示波器上绘制的 SDFM 通道 A 检测到的信号.....	35
图 5-3. TIDA-01606 在实验中的测试设置.....	36
图 5-4. 实验 3 的输出波形 A、B、C (从 TIDA-01606 电网侧).....	36
图 5-5. 实验 3 的控制环路测试.....	37
图 5-6. PFC 模式测试设置.....	38
图 5-7. 实验 5 流程图.....	38
图 5-8. 显示 AM263x 中的所有内核.....	39
图 5-9. 将 CS_DAP_0 固定到 Memory Browser.....	39
图 5-10. 已固定 CS_DAP_0 用于实时调试的 Memory Browser.....	40
图 5-11. “Expressions”窗口中的 Gi 环路和功率级状态.....	40
图 5-12. Memory Browser 中的 Gi 环路和功率级状态.....	40
图 5-13. 实验 6 流程图.....	41
图 5-14. PWM 死区软启动.....	42
图 5-15. 实验 7 流程图.....	42
图 5-16. PFC 模式下 TIDA 硬件电网侧 200V 交流输入的 $I_a$ 、 $I_b$ 、 $I_c$ 波形.....	43
图 5-17. PFC 硬件在 200V 交流输入下的输出总线电压.....	43
图 5-18. PFC 模式下 TIDA 硬件电网侧 130V 交流输入的 $I_a$ 、 $I_b$ 、 $I_c$ 波形.....	44
图 5-19. PFC 硬件在 130V 交流输入下的输出总线电压.....	44
图 5-20. 200V 交流输入和恒定总线电压基准值下的 ITHD 和 PF 值.....	45
图 5-21. 130V 交流输入和恒定总线电压基准值下的 ITHD 和 PF 值.....	45

## 表格清单

表 2-1. 电路板上用于控制功率级的关键控制器外设.....	4
表 3-1. E1 controlCARD 的 ADC 至 PWM SOC 映射.....	18
表 3-2. 带 HSEC 板的 E1 和 E2 controlCARD 的 ADC 映射.....	18
表 3-3. E2 controlCARD 的 ADC 至 PWM SOC 映射.....	19
表 3-4. E1 controlCARD 的 ADC 至 CMPSS 映射.....	20
表 3-5. E2 controlCARD 的 ADC 至 CMPSS 映射.....	20
表 3-6. 所有 ISR1 实验.....	25
表 3-7. PWM XBAR 与跳闸的对应关系.....	30
表 3-8. 软件中的故障 API.....	30
表 3-9. 不同系统故障的 PWM XBAR 位位置.....	31
表 5-1. 中断操作中的 THD 值.....	37
表 5-2. ISR1 最长和最短 CPU 执行时间计算.....	37
表 5-3. 实验 7 与 AM263x controlCARD™ 配合使用 PFC 时的运行性能.....	44
表 5-4. PFC 运行的中断基准测试.....	46

## 商标

controlCARD™, C2000™, and Code Composer Studio™, and E2E™ are trademarks of Texas Instruments.  
Arm® and Cortex® are registered trademarks of Arm Limited.  
所有商标均为其各自所有者的财产。

## 1 引言

本文档介绍了如何使用 AM263x MCU 来控制 TIDA-01606 基于 SiC 的双向三级、三相逆变器参考设计。该硬件与采用 HSEC controlCARD™ 格式的 MCU 兼容。虽然最初为 C2000™ MCU 产品系列而设计，但该参考设计也接受了 AM263x controlCARD (TMDSCNCD263)，只需进行少量修改即可。该参考设计可用于在光伏逆变器、电动汽车充电器、电池储能系统以及 UPS 和电动汽车车载充电器 (OBC) 中实现单向或双向交流/直流和直流/交流功率级。此设计中使用的 T 型三级功率级是提高这些系统的功率密度和效率的理想选择。

TIDA-01606 参考设计由四个独立的板组成，这些板协同工作以形成该三相逆变器参考设计：

- 一块电源板，包括开关器件、LCL 滤波器、传感电子器件和板电源
- TMDSCNCD263 通用 HSEC controlCARD，用于连接到基于 AM2634 Arm® 的 MCU
- 三块栅极驱动器卡，每块卡带有两个 ISO5852S 和两个 UCC5320 栅极驱动器
- 一个直流母线电压测量板 (TIDA-01606 ISOHV CARD)

有关其他硬件详细信息，请参阅 [TIDA-01606 : 10kW 双向三相三级 \(T 型\) 逆变器和 PFC 参考设计](#) 设计指南。

## 2 硬件要求

### 2.1 所需的硬件和测试仪器

该参考设计包含以下元素：

- 一块 TIDA-01606 电源板

以下列表显示了用于控制 MCU 上的功率级的重要资源：

- 三块 TIDA-01606 栅极驱动器卡
- 一块 TIDA-01606 ISOHV CARD
- 一块 TMDSCNCD263 controlCARD
- Mini USB 电缆
- 用于将软件加载到 AM263x MCU 的计算机

对本设计进行供电和评估需要以下测试设备列表：

- 用作电路板初级电源的 15V、4A 台式电源

对于 PFC 模式：

- 支持 400V L-L 的三相交流电源
  - 要连接到直流输出端的 10kW 电阻负载

对于逆变器模式：

- 10kW 星形连接式电阻负载网络
- 用于直流链路输入的 800V、12A 电源
- 四通道功率分析仪
- 带有电流和高压探头的示波器

### 2.2 设计中使用的微控制器资源

表 2-1 列出了 MCU 上用于控制功率级的重要资源：

表 2-1. 电路板上用于控制功率级的关键控制器外设

引脚编号	说明	软件名称
15、31、28	电网电压感测 A、B、C 相	TINV_VGRID_A、B、C
21、33、30	逆变器侧电压 A、B、C 相	TINV_VINV_A、B、C
25、37、34	逆变器侧电流 A 相	TINV_IINV_A、B、C
42	总线电压感测	TINV_VBUS
40	总线电压中点检测	TINV_VBUS_MID
12、14、18、20	A、B、C 相和环境温度	TINV_TEMP_A、B、C、AMB

表 2-1. 电路板上用于控制功率级的关键控制器外设 (continued)

引脚编号	说明	软件名称
49、50、58	Q1 PWM A、B、C 相	TINV_Q1_A、B、C
51、52、60	Q3 PWM A、B、C 相	TINV_Q3_A、B、C
53、54、62	Q2 PWM A、B、C 相	TINV_Q2_A、B、C
99、103、107	SDFM 数据 IG A、B、C	TINV_IGRID_A、B、C
101、105、109	SDFM 时钟 IG A、B、C	
57、75	SDFM 时钟源	
89、87、85	SiC 故障信号 A、B、C (低电平有效)	TINV_FAULT_A、B、C
86、88、90、92	A、B、C、N 上的继电器	TINV_RELAY_A、B、C、N
61、63	栅极驱动器电源 PWM	TINV_GATE_DRIVE
59	风扇的控制 GPIO	TINV_FAN
108、110	用于在启动固件调试时查看扩展坞上的 ISR 嵌套等	TINV_PROFILING1、2
95	栅极驱动器使能	TINV_PWM_EN
81	栅极驱动器复位	TINV_R

### 2.3 TIDA-01606 REV-6 的硬件更改

由于 TIDA-01606 电源板最初是使用 C2000™ MCU 产品系列为 controlCARD 打造的，因此电源板需要进行一些微小更改才能适应 AM263x controlCARD：

1. 用跳线短接 HSEC 连接器引脚 88 和 94，为继电器 B 提供输出。
2. R196 已从 TIDA-01606 板上移除，因为 R196 连接到 HSEC 板的引脚 32，该引脚路由到 AM263 的 GND。

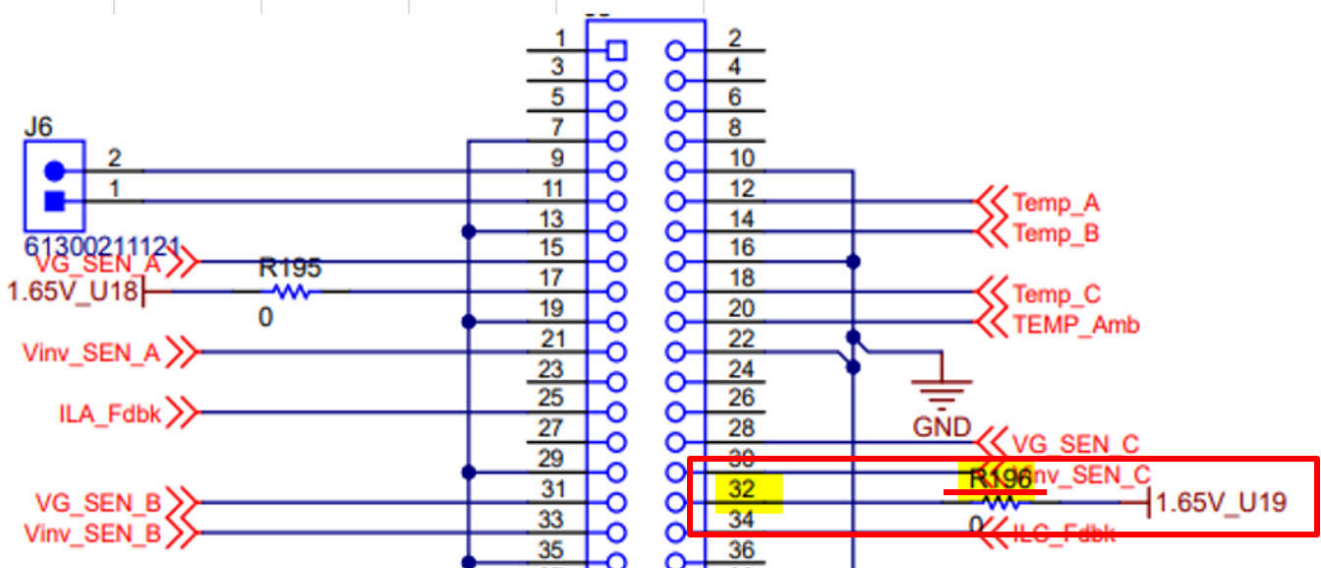


图 2-1. TIDA-01606 中的 R196

## 2.4 TMDSCNCD263 controlCARD™ 更改

需要对 TMDSCNCD263 controlCARD 进行细微更改，以控制电源硬件。所需的第一项更改允许访问 HSEC 板的引脚 92 (GPIO 134)，需要该引脚才能连接到逆变器板的继电器 N 引脚。

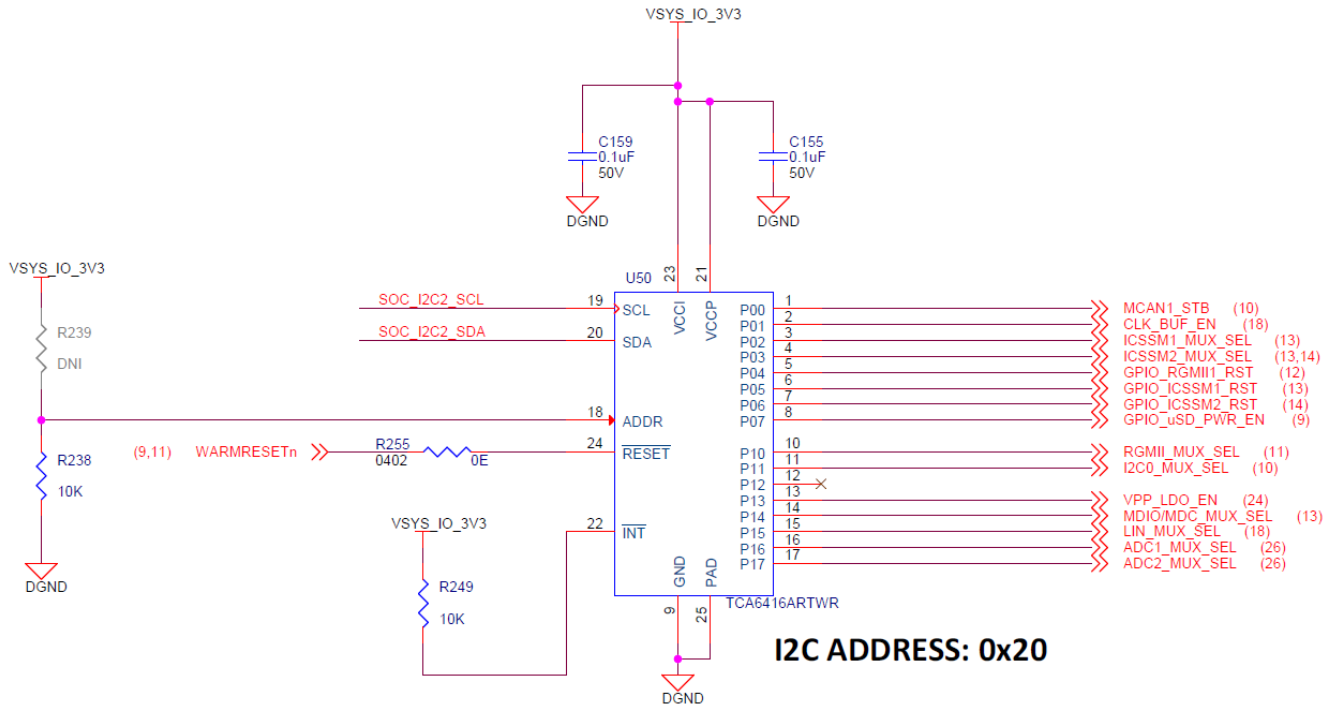
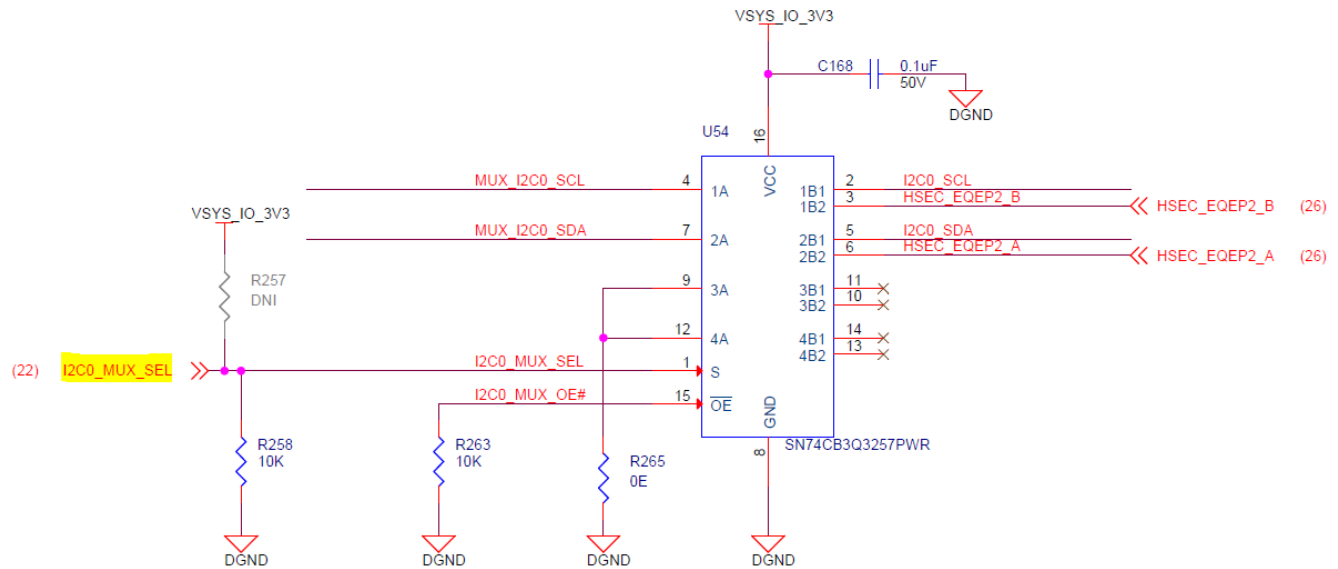


图 2-2. 用于 HSEC 引脚 92 配置的 I2C IO 扩展器



I2C0 - 1:2 MUX

SEL	CONDITION	FUNCTION
LOW	I2C0 SELECTED	A-->B1 port
HIGH	HSEC EQEP selected	A-->B2 port

图 2-3. 用于 HSEC 引脚 92 配置的 I2C 多路复用器和多路信号分离器

拆下 R258 电阻器并使用相同值的电阻器 (10kΩ) 填充 R257。或者，使用 I2C2 实例将 I2C0\_MUX\_SEL 引脚设置为高电平以访问 HSEC\_EQEP2\_A 引脚。图 2-5 提供了 SysConfig 设置，图后紧跟的是该设置的示例 C 代码。

GPIO 134 配置为 I2C FS 开漏电压缓冲器，因此需要一个外部上拉电阻器来实现高电平有效信号。

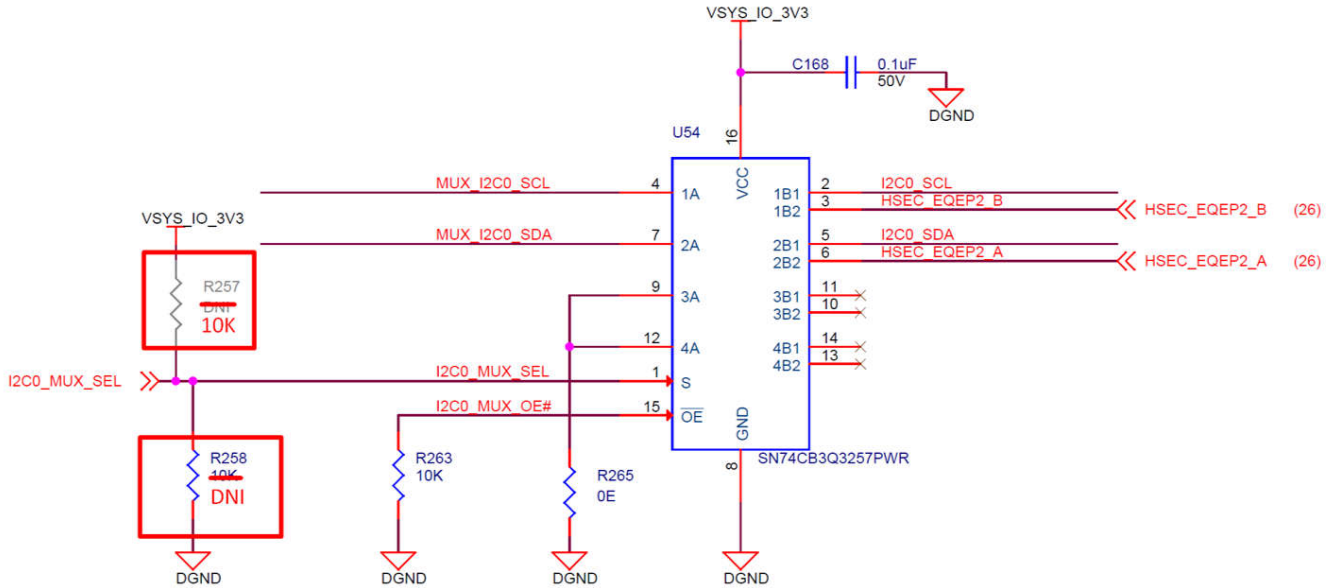


图 2-4. AM263x controlICARD 中的 I2C0 MUX SEL

I2C (1 Added) + ADD REMOVE ALL

✓ CONFIG\_I2C2 🗑️

Name: CONFIG\_I2C2

Bit Rate: 400 KHZ

Enable Interrupt:

Show Advanced Config:

I2C Instance: I2C2 🔒

IO Set: I2C2\_IOSet\_2

✓ Signals	Pins	Pull Up/Down	Slew Rate
✓ I2C Clock Pin(I2C2_SCL)	UART0_RTSn/C7 <span style="float: right;">🔒</span>	No Pull <span style="float: right;">▼</span>	Low <span style="float: right;">▼</span>
✓ I2C Data Pin(I2C2_SDA)	UART0_CTSn/B7 <span style="float: right;">🔒</span>	No Pull <span style="float: right;">▼</span>	Low <span style="float: right;">▼</span>

图 2-5. IO 扩展器的 I2C SysConfig 配置

以下代码块演示了如何将 I2C0 MUX SEL 设置为高电平。

```

I2C_Transaction i2cTransaction;
uint8_t buffer[2U];
int32_t status = SystemP_SUCCESS;

I2C_Transaction_init(&i2cTransaction);
i2cTransaction.writeBuf = buffer;
i2cTransaction.writeCount = 2U;
i2cTransaction.slaveAddress = 0x20;

buffer[0] = 0x03U;
buffer[1] = (0x01 << 1);
status += I2C_transfer(I2C_getHandle(CONFIG_I2C2), &i2cTransaction);

buffer[0] = 0x07U;
buffer[1] = ~(0x01 << 1);
status += I2C_transfer(I2C_getHandle(CONFIG_I2C2), &i2cTransaction);

```

需要进行另一项更改才能访问 HSEC 板的引脚 90 (GPIO 86)，以便连接继电器 C。拆下 R139 电阻器并使用相同值的电阻器 (0E) 填充 R146。

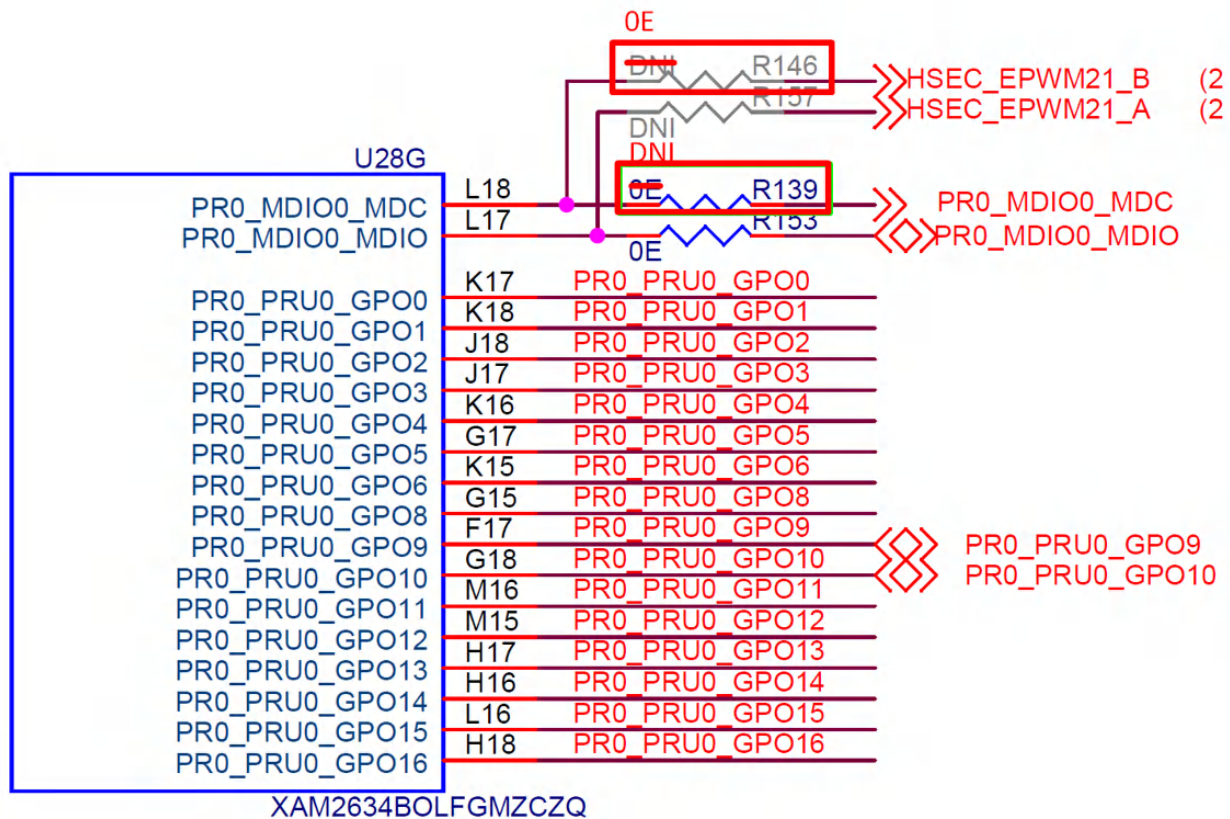


图 2-6. L18 连接到 EPW21\_B



## 3 软件

### 3.1 固件入门

#### 3.1.1 打开 Code Composer Studio™ 工程

该设计的软件可在 MCU+ Academy 内获取。若要打开工程：

1. 如有必要，请按照 [AM263x MCU+ SDK 入门指南](#) 中的说明安装 Code Composer Studio™ (CCS) 版本 12 或更高版本。
2. 支持的最低 SysConfig 版本为 1.14.0 或更高版本。
3. 本演示支持的最低 MCU\_PLUS\_SDK 版本为 mcu\_plus\_sdk\_am263x\_08\_05\_00\_24。
  - 如果演示不支持最新的 MCU SDK，请在 E2E™ 论坛上报告。
4. 如有必要，每个软件包的安装步骤将作为 [SDK 和工具设置](#) 的一部分提供。
5. 从 [Resource Explorer](#) 中的 AM26x 设计库下载 [AM263x 的串式逆变器演示](#)。
6. 打开 CCS 并创建一个新的工作区。
7. 在 CCS 中，转到 *Project* → *Import CCS Project*。导航到固件的位置，然后点击“import project”。
8. 要使用 CCS 启动工程，请将 EVM 设置为 NOBOOT 模式。按照 [入门指南](#) 的“[EVM 设置](#)”部分中所述的步骤设置 EVM。

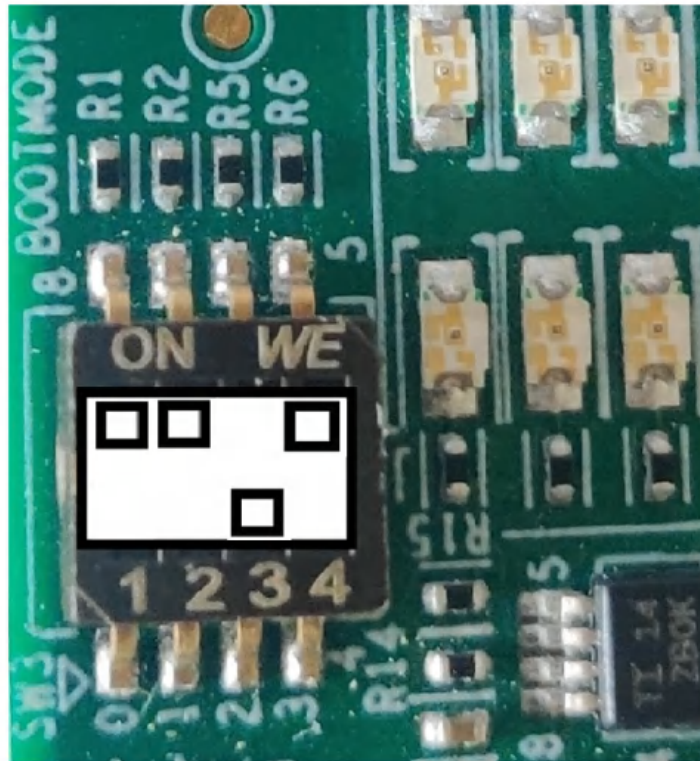


图 3-1. 无引导模式

9. 要将逆变器演示代码加载到 EVM 中并运行这些代码，请按照 [入门指南](#) 中 [加载和运行示例](#) 部分所述的步骤操作。

#### 3.1.2 软件架构

工程流程包含使用 AM263x SysConfig 工具和 MCU\_PLUS\_SDK\_AM263x 对片上系统 (SOC) 的所有外设时钟和子模块进行初始化的过程。节 3.2 讨论了如何初始化该设计中使用的 [外设 IP](#)，如 EPWM、ADC、CMPSS、GPIO、SDFM 等。

此软件包含 2 个 ISR 和 1 个后台任务。图 3-2 中介绍了代码流程。ISR1 计划每 20  $\mu$ s 运行一次，ISR2 计划每 1ms 运行一次。ISR1 包含节 3.3.1 中所述的主控制环路。ISR2 包含辅助控制功能，如计算用于后台监视的相电流和电压的 RMS 值，节 3.3.3 对此进行了说明。

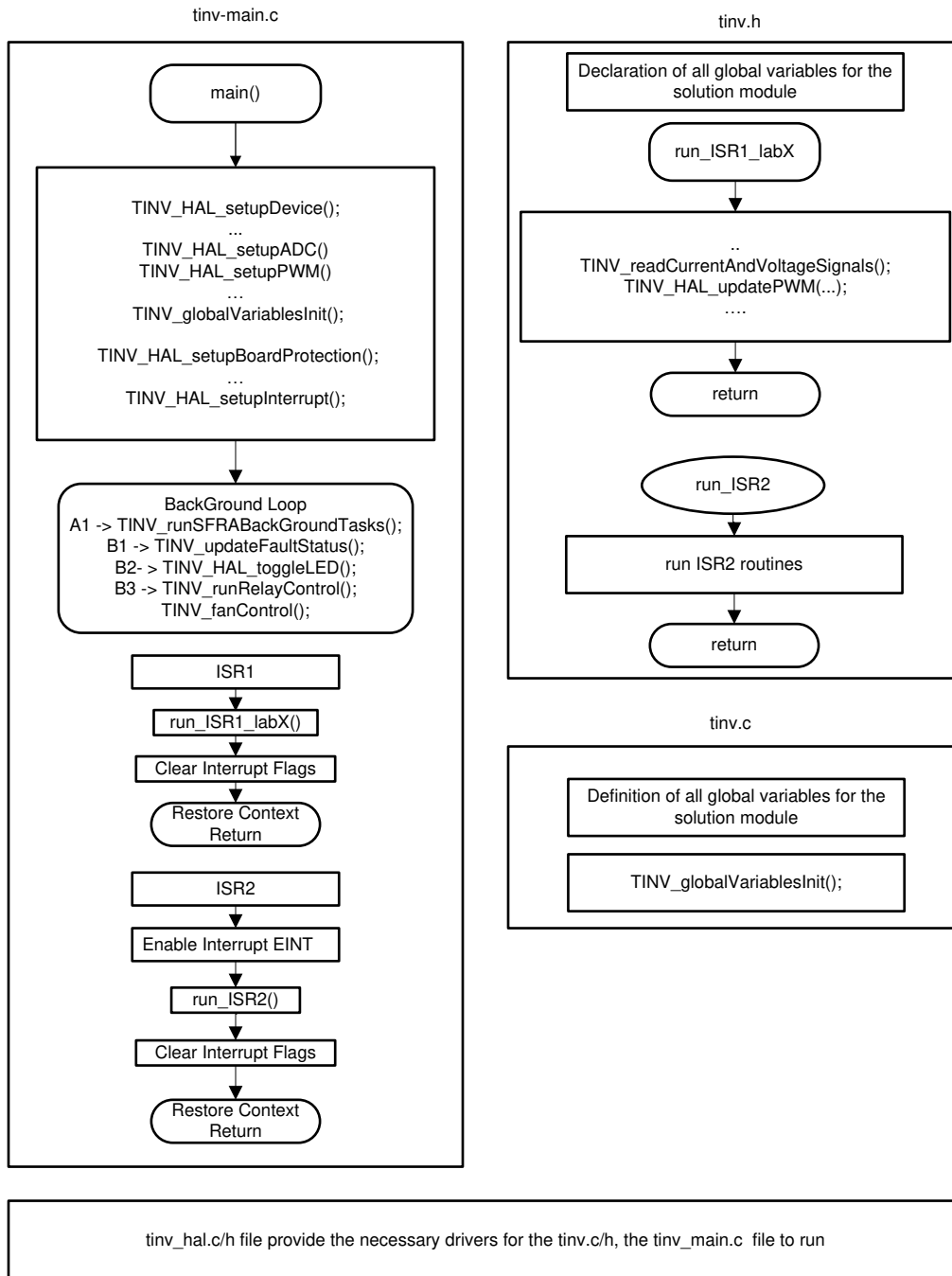


图 3-2. 软件方框图

该工程有两个版本的 ADC 检测数据，具体取决于 controlCARD 的版本 ( E1 和 E2 )。如果正在使用的电路板是 controlCARD 的 E1 版本，则启用宏 AM263x\_CC\_E1。或者，如果正在使用的电路板是 controlCARD 的 E2 版本，则启用宏 AM263x\_CC\_E2。这些宏设置在 Source/Includes 文件夹中的 tinv\_user\_settings.h 文件中定义。根据使用的电路板版本选择 .syscfg 文件，并从构建中排除另一个文件：对于 E1 controlCARD，此文件为 tinv\_AM263x\_E1.syscfg；对于 E2 controlCARD，此文件为 tinv\_AM263x\_E2.syscfg。

后台任务分为四个部分 - A1、B1、B2 和 B3。每次所连接的计时器溢出时，即延迟为由 RTI 计时器控制的 1ms，都会执行 A1。之前在 C2000 架构中使用 A1 来运行 SFRA 环路，目前 A1 保持开路，因为不支持 SFRA。

B1、B2 和 B3 通过带有 RTI 计时器的 1ms 延迟控制器依次执行。节 3.2.3 对此配置进行了说明。B2 任务通过捕获 ISR 上的 GPIO 高波形和低波形，使用 ECAP 计算 ISR1 和 ISR2 的 CPU 负载时间和平均 CPU 负载时间。B3 任务监视连接到 TIDA-01606 的风扇和继电器控制。B1 任务更新系统的故障状态。通过将宏

TINV\_PROTECTION 设置为 TINV\_PROTECTION\_ENABLED，可启用此设置。默认情况下，实验 1 禁用此步骤，而其他实验则启用此步骤。有关这方面的更多信息，请参阅节 3.4。

以下库取自 C2000 Digital Power SDK 工具套件，并集成了逆变器演示工程。

- **Transforms 库**：此库用于执行 Clarke 和 Park 变换函数。三相交流波形具有三种不同的量，因此很难控制输出波形并在全部三个波形之间实现同步。因此，必须将波形转换为直流域，以应用与控制增益相关的计算。
  - ABC 至 DQ → Clarke + Park 变换。这会将三相量转换为旋转坐标系中的直流量，以实现正序列和负序列。
  - DQ 至 ABC → 逆向 Park + 逆向 Clarke 变换。这会将旋转坐标系中的直流量转换为时域中的三相量。这两种变换完全通过软件计算实现。
- **Rampgen 库**：此库提供一组可生成所需频率的斜坡信号的函数。在逆变器运行模式下，需要使用基准斜坡信号来计算 50Hz 正弦波输出所需的 PLL 角度。

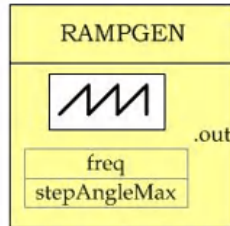


图 3-3. Rampgen 模型

- **SPLL - 软件锁相环 (PLL)**：这款基于软件的 PLL 代码基于三相电网连接的同步坐标系。首先，检测并记录三个相电压 A、B 和 C。然后将这些电压转换为直流分量 D 和 Q。要确保三相系统是同步的；Q 值必须接近零。SPLL 使用该设定生成下一个  $\theta$ ，以提供有效输出。[适用于三相并网应用并采用 C2000™ 微控制器的软件锁相环设计](#) 应用手册中对这些信息进行了深入介绍。

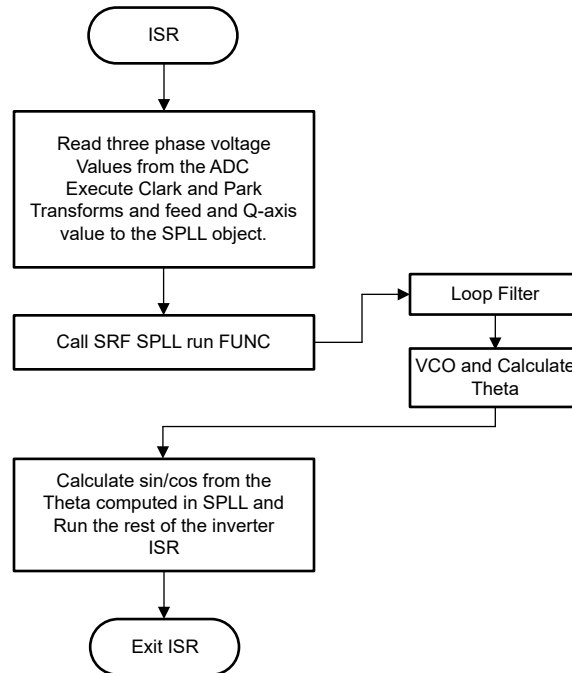
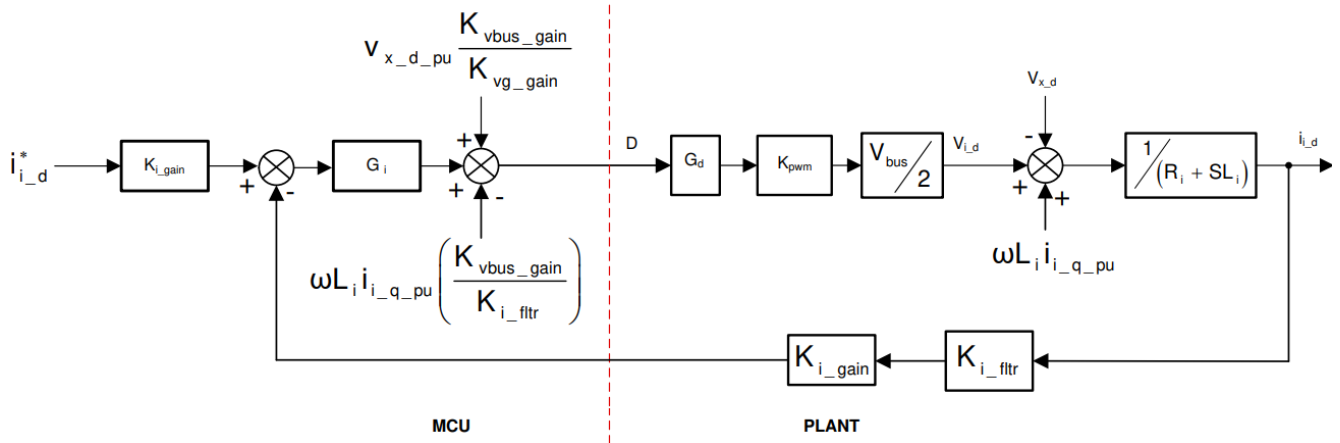


图 3-4. 控制 ISR 中的 SPLL 使用流程

- **PIC 控制器库**：本演示使用比例积分控制环路 (PI) 来维持输出电流。用户为系统选择基准电流值。无论输入如何，PI 控制器都必须在系统中保持该电流。可根据  $I_a$ 、 $I_b$  和  $I_c$  计算  $I_d$  和  $I_q$  值。现在将生成的  $I_d$  与基准  $I_d$  进行比较，并计算出误差。接下来，确定 PWM 的脉冲宽度，从而调节输出电流。图 3-5 显示了电流控制环路的方框图。[10kW 双向三相三级 \(T 型\) 逆变器和 PFC 参考设计](#) 设计指南的控制设计部分介绍了 PIC 控制器的计算。



其中：

- $i_{i\_d}^*$  = 电流基准
- $K_{i\_gain}$  = 电流检测标量，是最大电流检测值的倒数
- $K_{i\_fltr}$  是在电流检测路径上连接的滤波器。电流检测标量，是最大电流检测值的倒数
- $K_{vbus\_gain}$  = 总线的电压检测标量，是最大电压检测值的倒数
- $K_{vg\_gain}$  = 电网电压的电压检测标量，是最大电压检测值的倒数

图 3-5.  $I_d$  电流环路模型

### 3.1.3 工程文件夹结构

图 3-6 显示了项目的总体结构。导入工程之后，CCS 内将显示 *Project Explorer*。

特定于设计并且独立于器件的文件位于 **Source** 文件夹中，其中含有核心算法代码。与数字电源应用相关的库和控制算法位于 **libraries** 文件夹中。特定于电路板和特定于驱动程序的更改在 **SysConfig** 中配置，此代码由 **Release/SysCfg** 文件夹中的 **SysConfig** 自动生成。**main.c** 文件包含工程的主要框架。该文件包括对电路板和文件的调用（有助于构建系统框架），以及中断服务例程 (ISR) 和慢速后台任务。图 3-6 介绍了详细的文件夹结构。

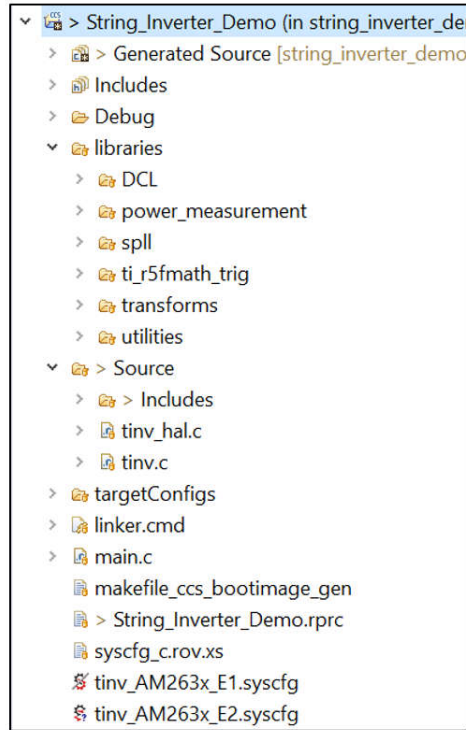


图 3-6. 演示的文件夹结构

所有变量和函数调用都以 *TINV* 名称开头（例如，*TINV\_vSecSensed\_pu*）。这种命名规则让用户能够将不同的设计组合在一起，同时避免命名冲突。

## 3.2 SysConfig 设置

AM263x controlCARD 有两个已发布版本：E1 和 E2。ADC 至 CMPSS 的映射中的版本有所不同。根据使用的电路板版本选择 .syscfg 文件，并从构建中排除另一个文件。

*tin\_v\_AM263x\_E1.syscfg* 或 *tin\_v\_AM263x\_E2.syscfg* 用于配置串式逆变器演示中的所有模块。要运行控制环路，需要使用 ADC、SDFM 和 EPWM 模块。对于电路板保护相关的应用，需要 CMPSS 配置。

最后，要将这些模块从一个元件链接到另一个元件，需要配置一些 XBAR。下一节将单独介绍这些模块中的每一个。

### 3.2.1 EPWM 配置

PWM 是逆变器软件配置的核心部分之一。AM263x 的 PWM 模块继承了 TI 经典 C28 控制器的特性。总共需要 12 个 PWM 通道来控制逆变器输出。

创建了六个 EPWM 实例，用于控制逆变器的三个臂：*TINV\_Q1\_Q3\_A\_PWM\_BASE*、*TINV\_Q2\_Q4\_A\_PWM\_BASE*、*TINV\_Q1\_Q3\_B\_PWM\_BASE*、*TINV\_Q2\_Q4\_B\_PWM\_BASE*、*TINV\_Q1\_Q3\_C\_PWM\_BASE*、*TINV\_Q2\_Q4\_C\_PWM\_BASE*。EPWM 外设时钟以 200MHz 的频率运行。在向上/向下计数模式下，每个 EPWM 都配置为 50kHz 频率。

$$Time\ Base\ Period\ for\ Up - DoWn\ Count\ Mode = \frac{\left(\frac{EPWMCLK}{HSPCLK \times CLKDIV}\right)}{2 \times ReqFreq} = \frac{200\ MHz}{100\ kHz} = 2000 \quad (1)$$

从方程式 1 可以看出，当高速时钟和时基时钟分频器为 1 时，每个 EPWM 时基周期为 2000。本节仅介绍了 TINV\_Q1\_Q3\_A\_PWM\_BASE 的初始化。不过，对于其他 5 个 PWM 实例，也需要重复相同的初始化过程。图 3-7 显示了 TINV\_Q1\_Q3\_A\_PWM\_BASE 的完整时基配置。

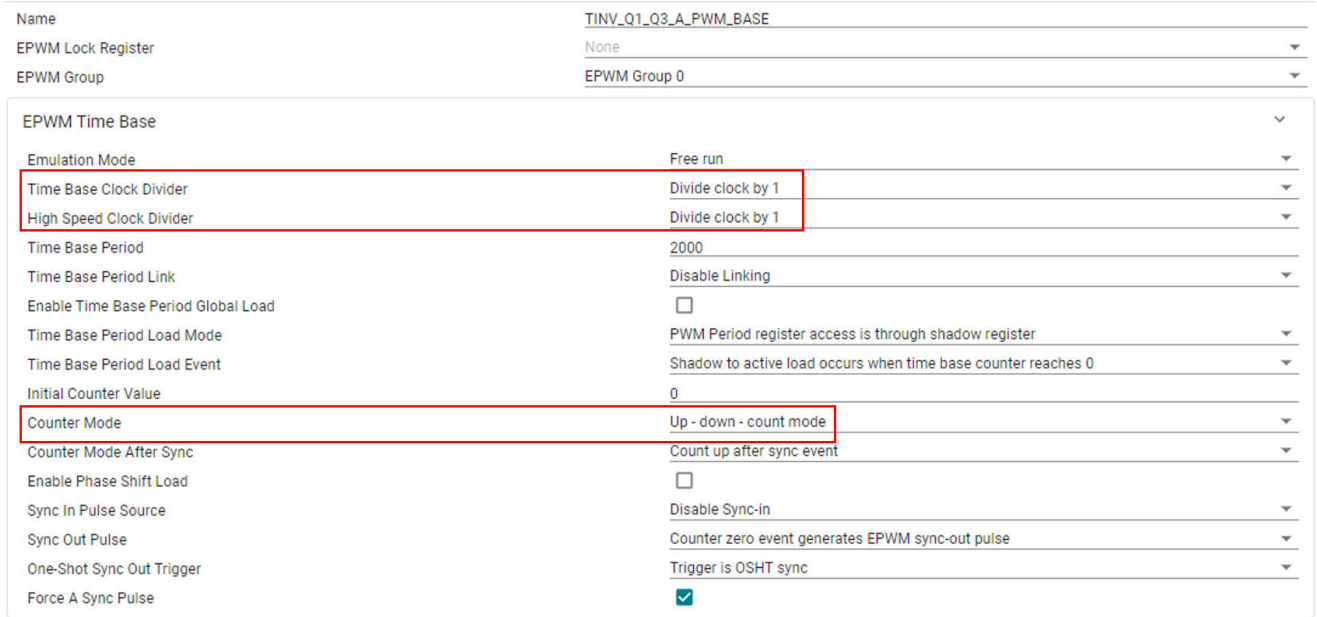


图 3-7. EPWM 时基子模块配置

通过修改计数器比较 A，可在 ISR 中每 20 μs 改变一次每个 EPWM 的脉冲宽度。放置在 CMPA 寄存器中的值是通过 PLL 环路计算的，更多相关信息将在以后的章节中讨论。目前，CMPA 初始化为 0。

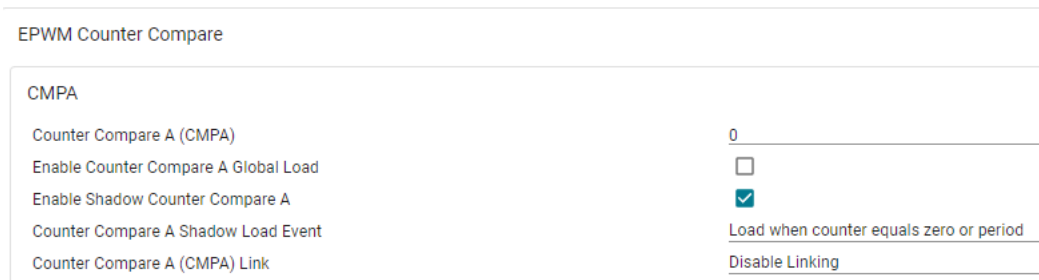


图 3-8. EPWM 计数器比较子模块配置

EPWM 脉冲配置为变为高电平和低电平，具体取决于计数器达到 CMPA 值时的事件。如果计数器在递增计数时达到 CMPA，EPWM 脉冲变为低电平。类似地，如果计数器在递减计数时达到 CMPA 值，EPWM 将设置为高电平。这样就可以为高 CMPA 值提供较高的占空比，并为低 CMPA 值提供较低的占空比。

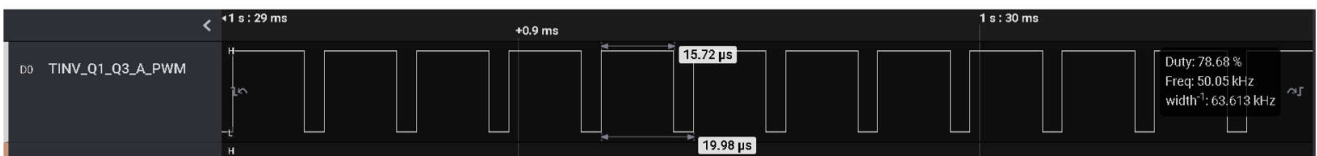


图 3-9. 较高的 CMPA 值会产生高占空比



图 3-10. CMPA 值可改变 EPWM0 波形的占空比

EPWM Action Qualifier	
Continuous SW Force Global Load	<input checked="" type="checkbox"/>
Continuous SW Force Shadow Mode	Shadow mode load when counter equals zero or period
EPWMXA Output Configuration	
EPWMXA Global Load Enable	<input checked="" type="checkbox"/>
EPWMXA Shadow Mode Enable	<input checked="" type="checkbox"/>
EPWMXA Shadow Load Event	Load when counter equals zero
EPWMXA T1 Trigger Source	Digital compare event A 1
EPWMXA T2 Trigger Source	Digital compare event A 1
EPWMXA One-Time SW Force Action	No change in the output pins
EPWMXA Continuous SW Force Action	Set output pins to low
Events Configured For EPWMXA Output	None
EPWMXA OutputEvent Output Configuration	
EPWMXA TBCTR Equals Zero	No change in the output pins
EPWMXA TBCTR Equals Period	No change in the output pins
EPWMXA TBCTR Up Equals COMPA	Set output pins to low
EPWMXA TBCTR Down Equals COMPA	Set output pins to High
EPWMXA TBCTR Up Equals COMPB	No change in the output pins
EPWMXA TBCTR Down Equals COMPB	No change in the output pins
EPWMXA T1 Event On Count Up	No change in the output pins
EPWMXA T1 Event On Count Down	No change in the output pins
EPWMXA T2 Event On Count Up	No change in the output pins
EPWMXA T2 Event On Count Down	No change in the output pins

图 3-11. EPWM 动作限定符子模块配置

这些动作会在发生下一个 CTR = 0 事件时加载到所有 EPWM 中。当动作限定符设置中的 CTR = 0 时，这是“影子到活动”加载选项。将加载模式选为全局模式，以确保同时加载到所有 PWM。有关“影子到活动”加载的详细信息，请参阅 *AM263x Sitara™ 微控制器技术参考手册* 的 *AQCTLA* 和 *AQCTLB* 影子模式操作一节。

EPWMB 通过死区进行配置，上升沿和下降沿延迟为 20 (  $20 \times 5 = 100\text{ns}$  延迟 )。

EPWM Dead-Band	
Rising Edge Delay Input	Input signal is ePWMA
Falling Edge Delay Input	Input signal is ePWMA
Rising Edge Delay Polarity	DB polarity is not inverted
Falling Edge Delay Polarity	DB polarity is inverted
Enable Rising Edge Delay	<input checked="" type="checkbox"/>
Rising Edge Delay Value	20
Enable Falling Edge Delay	<input checked="" type="checkbox"/>
Falling Edge Delay Value	20
Swap Output for EPWMxA	<input type="checkbox"/>
Swap Output for EPWMxB	<input type="checkbox"/>
Enable Deadband Control Global Load	<input type="checkbox"/>
Enable Deadband Control Shadow Mode	<input checked="" type="checkbox"/>
Deadband Control Shadow Load Event	Load when counter equals zero
Enable RED Global Load	<input type="checkbox"/>
Enable RED Shadow Mode	<input checked="" type="checkbox"/>
RED Shadow Load Event	Load when counter equals period
Enable FED Global Load	<input type="checkbox"/>
Enable FED Shadow Mode	<input checked="" type="checkbox"/>
FED Shadow Load Event	Load when counter equals zero
Dead Band Counter Clock Rate	Dead band counter runs at TBCLK rate

图 3-12. EPWM 死区子模块配置

用于过流和过压保护的数字比较和跳闸区 EPWM 模块在文档的 CMPSS 部分中进行了配置。

### 3.2.2 EPWM 事件触发中断

在递增计数时，ISR1 由 *TINV\_Q1\_Q3\_A\_PWM* 的计数器 = CMPC 事件触发。对于 CMPC = 100 事件，它是在 EPWM 的事件触发器子模块中进行配置，如图 3-13 所示。因此，ISR1 每 20  $\mu$ s 触发一次。

EPWM Event-Trigger 	
Enable EPWM Interrupt	<input checked="" type="checkbox"/>
Interrupt Event Sources	Time-base counter equal to zero
Interrupt Event Count	Disabled
Interrupt Event Count Initial Value	<input type="checkbox"/>
Force Interrupt Event Count Value	<input type="checkbox"/>

图 3-13. EPWM 事件触发器子模块配置

这些设置可根据设计要求进行调整。AM263 中共有 32 个不同的中断 XBAR。此 EPWM 事件将路由到中断 XBAR0。稍后会构建中断，并将 ISR 例程添加到同一 XBAR0 中。


INT XBAR (1 of 32 Added) 	
<input checked="" type="checkbox"/> CONFIG_INT_XBAR0	
Name	CONFIG_INT_XBAR0
XBAR Output	EPWM1_INT
Instance	INT_XBAR_0

图 3-14. EPWM1 中断的 INT XBAR 配置

### 3.2.3 计时器配置

ISR2 计时器用于实现以下功能：

- 计算检测到的电流和电压的 RMS 值，然后计算功率的 RMS 值。
- 根据计算出的  $I_d$ 、 $I_q$ 、 $V_d$  和  $V_q$  值计算有功功率和无功功率
- 使用 ADC 检测和计算所有通道的温度。



使用 RTI 计时器中断每 1ms 触发一次此 ISR2。图 3-15 显示了使用名为 ISR2 的计时器回调例程配置计时器中断所需的 SysConfig 选项。

ISR2_TIMER	
Name	ISR2_TIMER
Input Clock Frequency (Hz)	25000000
Input Clock Source	WUCPUCLK
Tick Period (usecs)	1000
Actual Tick Period (usecs)	1000
DMA Trigger Enable	<input type="checkbox"/>
Start Timer After Setup	<input checked="" type="checkbox"/>
Interrupt Mode Enable	<input checked="" type="checkbox"/>
Interrupt Priority	6
Timer Callback	ISR2
Timer Instance	Any(RTI1)
<input checked="" type="checkbox"/> Signals $t_1$	Pins

图 3-15. ISR2 的计时器配置

同样，对于 TASK\_A 和 TASK\_B 操作，为 TASK\_A\_TIMER 和 TASK\_B\_TIMER 配置了 1ms 的时钟周期。这些任务监测用于电路板保护的过流事件并计算 ISR1 和 ISR2 的负载平均值。

### 3.2.4 SDFM 配置

在全部三个相位 A、B 和 C 中配置三个 SDFM 通道进行电流检测，如图 3-16 所示。

CONFIG_SDFM0	
Name	CONFIG_SDFM0
SDFM Instance	SDFM0
IO Set	
<input checked="" type="checkbox"/> Signals $t_1$	Pins
<input type="checkbox"/> SDFM CLK Pin	Any
<input checked="" type="checkbox"/> SDFM CLK Pin(SDFM0_CLK1)	Any(SDFM0_CLK1/A16)
<input checked="" type="checkbox"/> SDFM CLK Pin(SDFM0_CLK2)	Any(SDFM0_CLK2/B15)
<input checked="" type="checkbox"/> SDFM CLK Pin(SDFM0_CLK3)	Any(SDFM0_CLK3/A15)
<input type="checkbox"/> SDFM Data Pin	Any
<input checked="" type="checkbox"/> SDFM Data Pin(SDFM0_D1)	Any(SDFM0_D1/D13)
<input checked="" type="checkbox"/> SDFM Data Pin(SDFM0_D2)	Any(SDFM0_D2/C13)
<input checked="" type="checkbox"/> SDFM Data Pin(SDFM0_D3)	Any(SDFM0_D3/C14)

图 3-16. SDFM 配置

### 3.2.5 ADC 配置

此演示中使用 ADC 来检测电网侧、逆变器侧、总线侧等的所有电压、电流和温度。如果使用 ADC 而不是 SDFM，则需总共配置 11 个 ADC 以及 3 个用于电流检测的可选 ADC。表 3-1 汇总了 ADC SysConfig 设置。使用 *Inv* 定义在电路的逆变器侧检测到的信号，并使用 *Grid* 定义在电路的电网侧检测到的信号。逆变器侧和电网侧的电流值相同。

对于 AM263x controlCARD 的 E1 和 E2 版本，ADC HSEC 板引脚排列不同，如表 3-2 中所述。

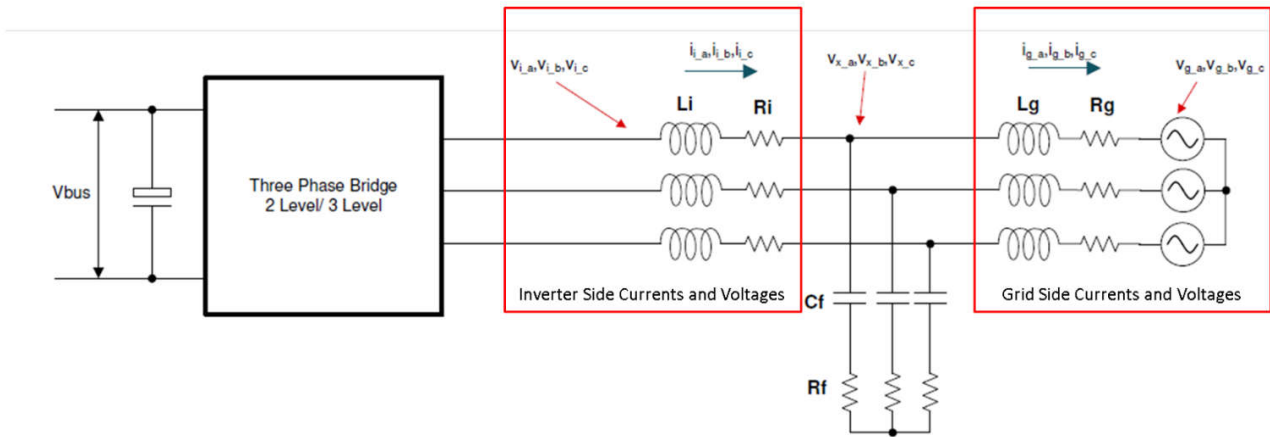


图 3-17. TIDA-01606 的电网侧和逆变器侧

表 3-1. E1 controlCARD 的 ADC 至 PWM SOC 映射

ADC 类型	ADC 基址	ADC SOC 型号	ADC 引脚	触发条件
TINV_TEMP_A_ADC_BASE	CONFIG_ADC0_BASE_ADDR	ADC_SOC_NUMBER0	ADC0_AIN0	ePWM0、ADCSOCB
TINV_TEMP_B_ADC_BASE	CONFIG_ADC0_BASE_ADDR	ADC_SOC_NUMBER1	ADC0_AIN1	ePWM0、ADCSOCB
TINV_VGRID_A_ADC_BASE	CONFIG_ADC0_BASE_ADDR	ADC_SOC_NUMBER2	ADC0_AIN2	ePWM0、ADCSOCA
TINV_TEMP_C_ADC_BASE	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER0	ADC1_AIN0	ePWM0、ADCSOCB
TINV_TEMP_AMB_ADC_BASE	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER1	ADC1_AIN1	ePWM0、ADCSOCB
TINV_VINV_A_ADC_BASE	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER2	ADC1_AIN2	ePWM0、ADCSOCA
TINV_IINV_A_ADC_BASE (或 SDFM)	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER3	ADC1_AIN3	ePWM0、ADCSOCA
TINV_VGRID_C_ADC_BASE	CONFIG_ADC2_BASE_ADDR	ADC_SOC_NUMBER0	ADC2_AIN2	ePWM0、ADCSOCA
TINV_VINV_C_ADC_BASE	CONFIG_ADC2_BASE_ADDR	ADC_SOC_NUMBER1	ADC2_AIN3	ePWM0、ADCSOCA
TINV_VGRID_B_ADC_BASE	CONFIG_ADC3_BASE_ADDR	ADC_SOC_NUMBER0	ADC3_AIN0	ePWM0、ADCSOCA
TINV_VINV_B_ADC_BASE	CONFIG_ADC3_BASE_ADDR	ADC_SOC_NUMBER1	ADC3_AIN1	ePWM0、ADCSOCA
TINV_IINV_C_ADC_BASE (或 SDFM)	CONFIG_ADC3_BASE_ADDR	ADC_SOC_NUMBER2	ADC3_AIN2	ePWM0、ADCSOCA
TINV_IINV_B_ADC_BASE (或 SDFM)	CONFIG_ADC4_BASE_ADDR	ADC_SOC_NUMBER0	ADC4_AIN0	ePWM0、ADCSOCA
TINV_VBUS_ADC_BASE	CONFIG_ADC4_BASE_ADDR	ADC_SOC_NUMBER1	ADC4_AIN3	ePWM0、ADCSOCA

表 3-2. 带 HSEC 板的 E1 和 E2 controlCARD 的 ADC 映射

HSEC 板	E1	E2
12	ADC0_AIN0	ADC1_AIN0
14	ADC0_AIN1	ADC1_AIN1
15	ADC0_AIN2	ADC0_AIN2
18	ADC1_AIN0	ADC1_AIN2
20	ADC1_AIN1	ADC1_AIN3
21	ADC1_AIN2	ADC0_AIN4
23	ADC1_AIN3	ADC0_AIN5
28	ADC2_AIN2	ADC3_AIN0
30	ADC2_AIN3	ADC3_AIN1
31	ADC3_AIN0	ADC2_AIN0
33	ADC3_AIN1	ADC2_AIN1
34	ADC3_AIN2	ADC3_AIN2
37	ADC4_AIN0	ADC2_AIN2
39	ADC4_AIN3	ADC2_AIN3

表 3-3. E2 controlCARD 的 ADC 至 PWM SOC 映射

ADC 类型	ADC 基址	ADC SOC 型号	ADC 引脚	触发条件
TINV_VGRID_A_ADC_BASE	CONFIG_ADC0_BASE_ADDR	ADC_SOC_NUMBER0	ADC0_AIN2	ePWM0、ADCSOCA
TINV_VINV_A_ADC_BASE	CONFIG_ADC0_BASE_ADDR	ADC_SOC_NUMBER1	ADC0_AIN4	ePWM0、ADCSOCA
TINV_IINV_A_ADC_BASE (或 SDFM)	CONFIG_ADC0_BASE_ADDR	ADC_SOC_NUMBER2	ADC0_AIN5	ePWM0、ADCSOCA
TINV_TEMP_A_ADC_BASE	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER0	ADC1_AIN0	ePWM0、ADCSOCB
TINV_TEMP_B_ADC_BASE	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER1	ADC1_AIN1	ePWM0、ADCSOCB
TINV_TEMP_C_ADC_BASE	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER2	ADC1_AIN2	ePWM0、ADCSOCB
TINV_TEMP_AMB_ADC_BASE	CONFIG_ADC1_BASE_ADDR	ADC_SOC_NUMBER3	ADC1_AIN3	ePWM0、ADCSOCB
TINV_VGRID_B_ADC_BASE	CONFIG_ADC2_BASE_ADDR	ADC_SOC_NUMBER0	ADC2_AIN0	ePWM0、ADCSOCA
TINV_VINV_B_ADC_BASE	CONFIG_ADC2_BASE_ADDR	ADC_SOC_NUMBER1	ADC2_AIN1	ePWM0、ADCSOCA
TINV_IINV_B_ADC_BASE (或 SDFM)	CONFIG_ADC2_BASE_ADDR	ADC_SOC_NUMBER2	ADC2_AIN2	ePWM0、ADCSOCA
TINV_VBUS_ADC_BASE	CONFIG_ADC2_BASE_ADDR	ADC_SOC_NUMBER3	ADC2_AIN3	ePWM0、ADCSOCA
TINV_VGRID_C_ADC_BASE	CONFIG_ADC3_BASE_ADDR	ADC_SOC_NUMBER0	ADC3_AIN0	ePWM0、ADCSOCA
TINV_VINV_C_ADC_BASE	CONFIG_ADC3_BASE_ADDR	ADC_SOC_NUMBER1	ADC3_AIN1	ePWM0、ADCSOCA
TINV_IINV_C_ADC_BASE (或 SDFM)	CONFIG_ADC3_BASE_ADDR	ADC_SOC_NUMBER2	ADC3_AIN2	ePWM0、ADCSOCA

图 3-18 显示了一个完全配置的 ADC0 实例。

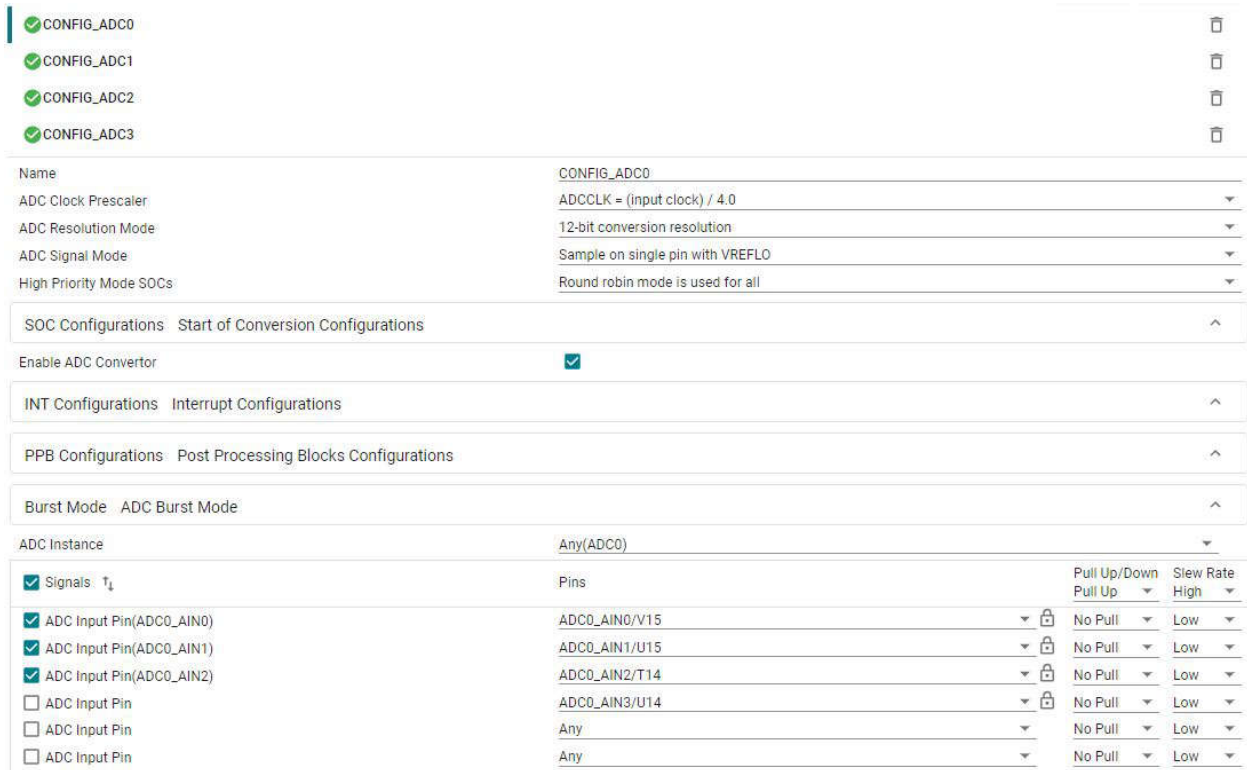


图 3-18. ADC 配置

查看“ADC 和 DAC 接口”一节中相应的电路板原理图文档，根据应用要求设置 ADC DAC 基准电压开关。

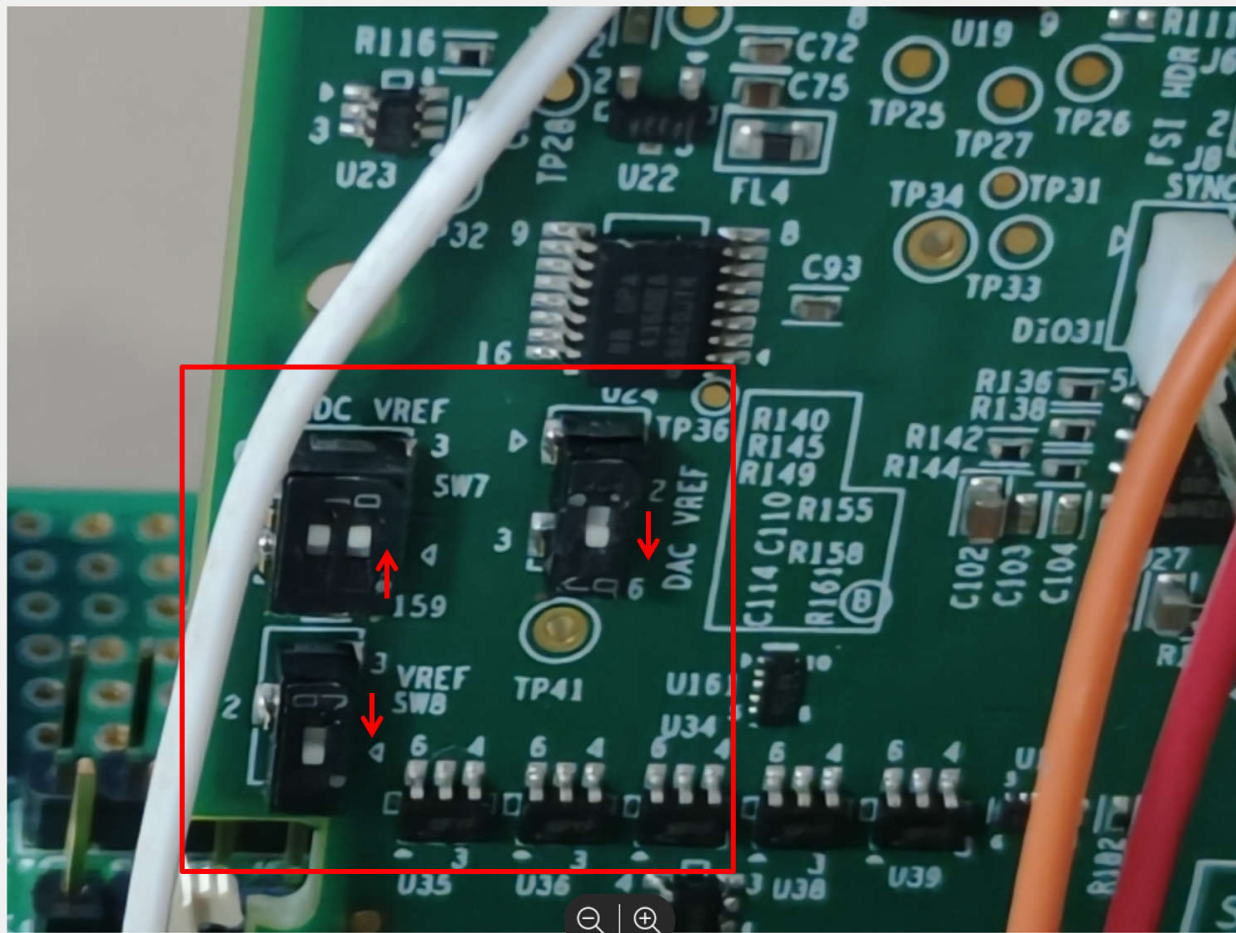


图 3-19. AM263x controlCARD 中的 ADC 和 DAC 基准开关

### 3.2.6 CMPSS 配置

比较器子系统 (CMPSS) 由模拟比较器和支持电路组成，这些电路用于电压和电流跳闸监控。电流演示使用 ADC 监控 6 个电压单元 ( 电网侧和逆变器侧组合 )。因此，配置了 6 个 CMPSS 比较器并将其连接到这些各自的 ADC。每个 CMPSS 都有一个连接到负极端子的基准 DAC。SysConfig 提供了一个数据字段来修改此 DAC 值，从而设置跳闸的阈值电压。表 3-4 介绍了使用的 6 个 CMPSS 模块。由于 E1 和 E2 之间的 ADC 引脚排列不同，因此连接到 HSEC 板的 E1 和 E2 版 controlCARD 的 CMPSS 和 ADC 映射不同。表 3-5 中列出了这种引脚排列的差异。

表 3-4. E1 controlCARD 的 ADC 至 CMPSS 映射

ADC	CMPSS	DAC 参考	跳闸事件
TINV_VGRID_A_ADC_BASE (ADC0_AIN2)	CMPSSA1:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VINV_A_ADC_BASE (ADC1_AIN2)	CMPSSA3:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VGRID_C_ADC_BASE (ADC2_AIN2)	CMPSSA5:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VINV_C_ADC_BASE (ADC2_AIN3)	CMPSSA5:inL	3600	无法实现一次性触发
TINV_VGRID_B_ADC_BASE (ADC3_AIN0)	CMPSSA6:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VINV_B_ADC_BASE (ADC3_AIN1)	CMPSSA6:inL	3600	无法实现一次性触发

表 3-5. E2 controlCARD 的 ADC 至 CMPSS 映射

ADC 类型名称	ADC 引脚	CMPSS	DAC 参考	跳闸事件
TINV_VGRID_A_ADC_BASE	ADC0_AIN2	CMPSSA1:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VINV_A_ADC_BASE	ADC0_AIN4	CMPSSB0:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VGRID_B_ADC_BASE	ADC2_AIN0	CMPSSA4:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VINV_B_ADC_BASE	ADC2_AIN1	CMPSSA4:inL	3600	无法实现一次性触发

表 3-5. E2 controlCARD 的 ADC 至 CMPSS 映射 (continued)

ADC 类型名称	ADC 引脚	CMPSS	DAC 参考	跳闸事件
TINV_VGRID_C_ADC_BASE	ADC3_AIN0	CMPSSA6:inH	3600	+ve 端子的输入高于 DCH 中的阈值
TINV_VINV_C_ADC_BASE	ADC3_AIN1	CMPSSA6:inL	3600	无法实现一次性触发

Name: CONFIG\_CMPSS1  
 CMPSS Instance: CMPSSA1  
 Enable Module:

图 3-20. 选择 CMPSS 实例并启用模块

图 3-21 说明了连接到 A 相电网侧电压 (TINV\_VGRID\_A\_ADC\_BASE) 的 ADC1\_AIN2 的高电平比较器配置。

Name: CONFIG\_CMPSS1  
 CMPSS Instance: CMPSSA1  
 Enable Module:

High Comparator Configuration

Negative Input Source: Input driven by internal DAC

Output Is Inverted:

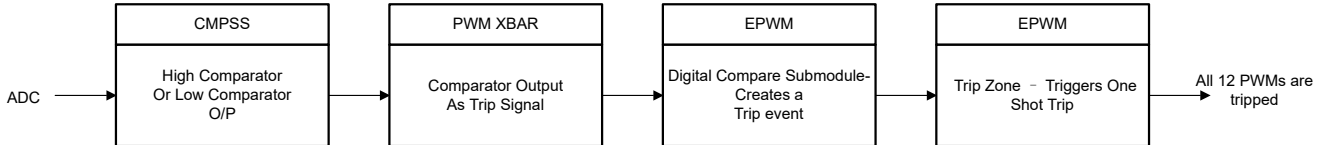
Asynch OR Latch:

Signal Driving CTRIPOUTH: Asynchronous comparator output drives CTRIPOUTH

Signal Driving CTRIPH: Asynchronous comparator output drives CTRIP

Set High Comparator DAC Value: 2500

图 3-21. CMPSS 高电平比较器配置和阈值设置



当超过阈值时，比较器会发出 CTRIPH、CTRIPL 信号（高电平）。该信号需要在 PWM XBAR 系统的帮助下传递到 EPWM 的跳闸区域模块。为高电平比较器或低电平比较器的每个 CTRIP 配置 6 个 PWM XBAR。

### 3.2.7 EPWM XBAR 配置

EPWM XBAR (12 of 30 Added)

- CONFIG\_EPWM\_XBAR3
- CONFIG\_EPWM\_XBAR4
- CONFIG\_EPWM\_XBAR5
- CONFIG\_EPWM\_XBAR6
- CONFIG\_EPWM\_XBAR7
- CONFIG\_EPWM\_XBAR8

Name: CONFIG\_EPWM\_XBAR8

XBAR Output: CMPSSA1\_CTRIPH

Invert Output Before Latch:

Instance: EPWM\_XBAR\_8

图 3-22. EPWM X-BAR 配置

对于高电平比较器事件，当 DCAEVT1 和 DCBEVT1 事件各自的跳闸设置为高电平时，将生成这两个事件。这里，EPWM\_XBARx 携带跳闸信号。图 3-23 中显示了同一配置的对对应关系。

对于低电平比较器，会生成 DCAEVT2 和 DCBEVT2。但这不会生成一次性跳闸（EVT2 仅适用于 CBC 跳闸）。

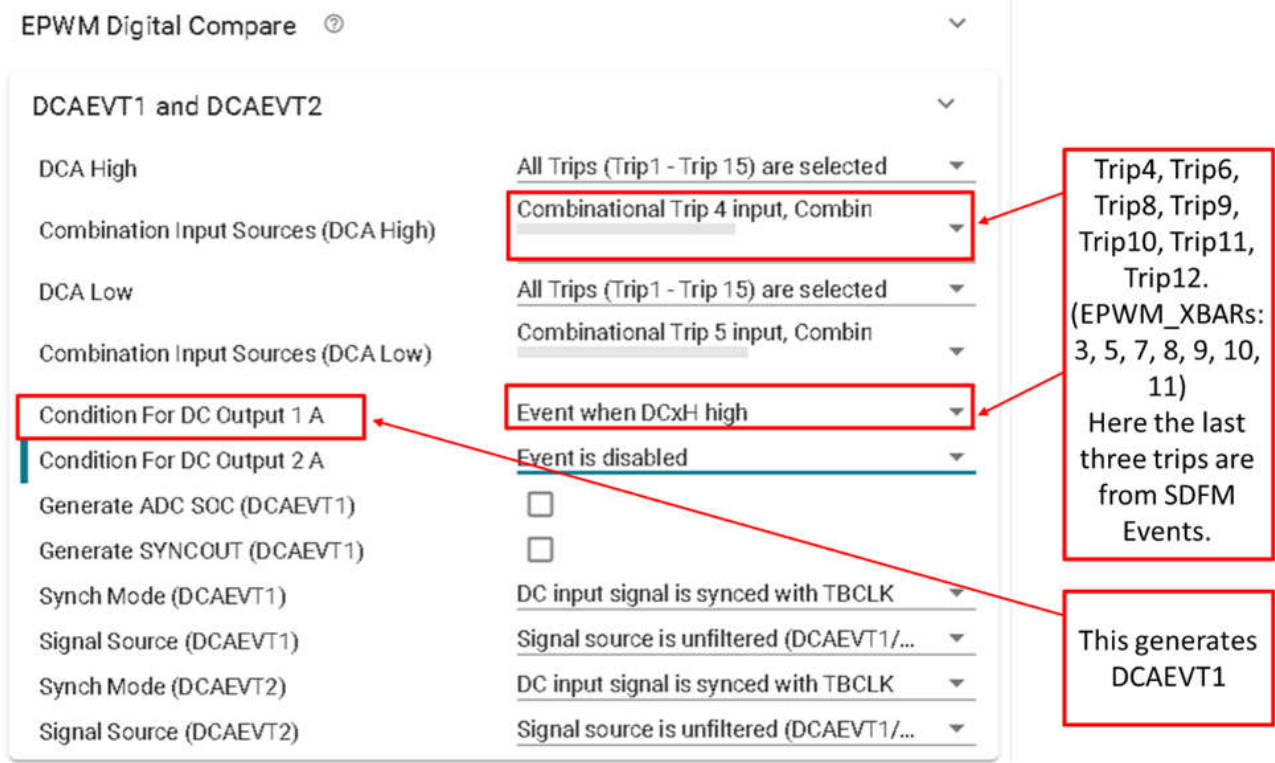


图 3-23. EPWM 数字比较子模块配置

生成事件后，将事件链接到一次性跳闸配置。对所有 6 个 PWM 实例重复此数字比较和跳闸区域配置。配置完成后，任何超过指定阈值的相电压或电流都将使系统中的所有 PWM 发生跳变。



图 3-24. EPWM 跳闸区域子模块配置

### 3.2.8 ECAP 配置

IC AMC3306 用作检测到的电流的  $\Sigma$ - $\Delta$  调制器。该 IC 的输出作为 SDFM 模块的输入。此 IC 需要输入时钟信号来创建同步调制输出。根据 IC 数据表，最大输入时钟频率可以为 21MHz。因此，若要生成最多数据样本，应创建周期为 50ns 的 APWM 波形。

$$A) \text{PWM Period} = \left( \frac{\text{ClkFreq}}{\text{PWMPFreq}} \right) - 1 = \left( \frac{200 \text{ MHz}}{20 \text{ MHz}} \right) - 1 = 9 \quad (2)$$

当发生 CTR = 0 事件时，APWM 变为高电平。ACMP ( APWM 比较 ) 是 PWM 变为低电平时 CTR 的值。ACMP 决定 APWM 的占空比。

<ul style="list-style-type: none"> <li>✓ CONFIG_ECAP1</li> <li>✓ CONFIG_ECAP_ISR1</li> <li>✓ CONFIG_ECAP_ISR2</li> </ul>	
Name	CONFIG_ECAP1
ECAP Instance	ECAP1
Emulation Mode	TSCTR is not affected by emulation suspension
ECAP Mode	APWM
Use Interrupt	<input type="checkbox"/>
Phase Shift Count	0
Enable Load Counter	<input type="checkbox"/>
Load Counter	<input type="checkbox"/>
Sync Out Mode	sync out on counter equals period
APWM Polarity	APWM is active high
APWM Period	9
APWM Compare	5
Sync-In Pulse Source	Disable Sync-in

图 3-25. ECAP 配置

### 3.2.9 输出 XBAR 配置

ECAP 配置为为  $\Sigma$ - $\Delta$  调制器提供时钟，并使用输出 XBAR 系统路由到 GPIO 引脚。AM263 共有 16 个输出 XBAR 引脚。根据 TIDA-01606 原理图，时钟输出必须从 HSEC-DOC 引脚 75 传递，该引脚映射到 OUTPUTXBAR3 ( AM263 的 B10 )。

<ul style="list-style-type: none"> <li>✓ CONFIG_OUTPUT_XBAR0</li> </ul>	
Name	CONFIG_OUTPUT_XBAR0
XBAR Output	ECAP1_OUT
Invert Output Before Latch	<input type="checkbox"/>
Select Latch Singal Source	Select Non Latched Output
Select Stretched Pulse Source	Select Non Stretched Output
Select Stretched Pulse Length	Output Signal Stretched Pulse Length is 16 SYSCLK
Invert XBAR Output	<input type="checkbox"/>
OUTPUTXBAR Instance	OUTPUTXBAR3
<input checked="" type="checkbox"/> Signals $f_d$	Pins
<input checked="" type="checkbox"/> Outputxbar Pin(OUTPUTXBAR3)	SPI1_D0/B10

图 3-26. 输出 X-Bar 配置

### 3.2.10 输入 XBAR 配置

来自三个栅极驱动器中任何一个的故障输入配置为输入 `GPIO FLT_A`、`FLT_B`、`FLT_C`。变为高电平时，这些信号表示栅极驱动器硬件中的故障条件。此输入作为跳闸输入通过输入 XBAR → EPWM XBAR 路由到 EPWM 的一次性跳闸区域。

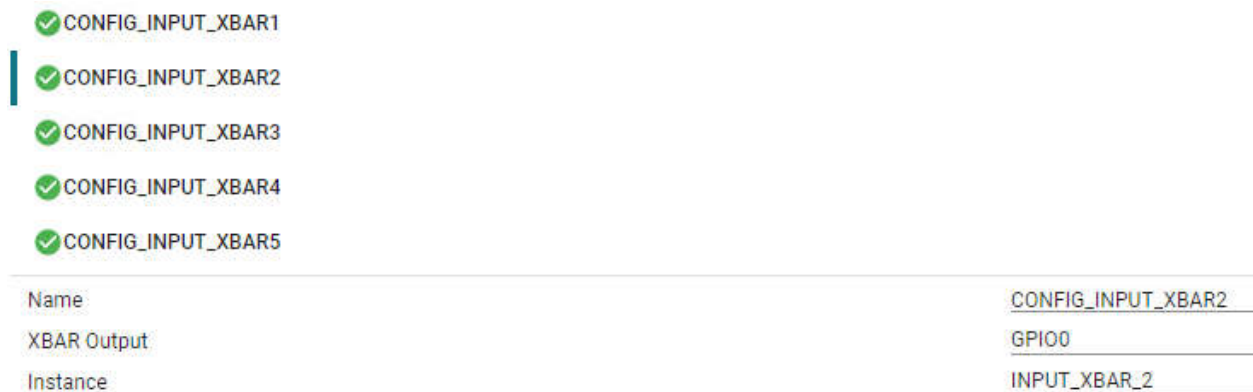


图 3-27. 输入 X-Bar 配置

从 EPWM XBAR，为数字比较子模块提供跳闸输入以创建 `DCAEVT1` 和 `DCBEVT1`。这些事件会再次配置为跳闸区域子模块中一次性跳闸的输入。节 3.2.6 对此过程进行了说明。

### 3.3 中断和实验结构

测试高电压参考设计通常在不同的阶段执行，首先是在禁用电路板保护的电压下进行开环测试。这个阶段测试的是基本功能，例如，中断和任务切换、ADC 检测、PWM 输出等，不深入研究控制环路功能。基本功能建立后，会引入电路板保护逻辑以验证高压功能。随着电压升高，系统中的噪声会增加，因此即使在高噪声条件下，软件也需要保留该功能。同时，如果系统中任何不必要的过压或过流导致设计降级或破坏，则软件需要使 PWM 跳闸。下一个阶段涉及闭合环路，以测试设计的完整功能。

软件旨在这些阶段之间切换，以便于测试和验证固件。系统初始化和后台任务在所有阶段都很常见。ISR1 中的源代码使用宏 `lab_number` 区分各阶段。该设计有七个通过实验进行测试的阶段，其中，实验 1 和实验 2 是逆变器工作模式的开环测试。实验 3 使用电阻负载在逆变器模式下实现了闭合电流环路。实验 4 通过并网操作在逆变器模式下实现了闭合电流环路。实验 5 和实验 6 分别用于 PFC 运行模式的开环和闭环执行。实验 7 通过并网操作在 PFC 模式下实现了闭合电流环路和闭合电压环路。有关这些实验的说明，请参阅节 3.3.1。

工程中包含两个 ISR (ISR1 和 ISR2)，其中 ISR1 是速度最快且不可嵌套的 ISR。节 3.3.1 和节 3.3.3 讨论了 ISR 的内容及其执行触发。



### 3.3.1 ISR1

ISR1 被保留用于控制环路和 PWM 更新。ISR1 由 `LEG1_PWM_BASE → EPWM_INT_TBCTR_U_CMPC` 事件触发。ISR1 总共有七个实验组件。一次只能选择运行一个实验。每个实验都启用了 API，具体取决于表 3-6 中所述的测试要求。

由于需要针对所使用的参数预先测试控制环路，因此对实验 1 和实验 2 进行了测试，以确保在启用闭环控制之前 PWM 更新和 ADC 检测在开环中保持稳定。实验 3 包含闭合电流环路实现，节 3.3.2 中对此进行了讨论。

本文针对本参考设计的软件进行了七个实验。表 3-6 列出了这些实验以及这些实验是如何测试的。

**表 3-6. 所有 ISR1 实验**

实验编号	说明	注释	测试环境
1	INV：PWM 和 ADC 检查	测试 PWM 驱动器、ISR 结构和执行速率，可在 controlCARD 上运行。单元测试保护机制。测试转换数据的 ADC 映射和读取。	controlCARD
2	INV：开环检查	PWM 检查、ADC 检查、保护检查、逆变模式直流总线连接和电阻星形网络作为负载	controlCARD + 功率级硬件
3	INV：闭合电流环路，电阻负载连接到交流电		controlCARD + 功率级硬件
4	INV：闭合电流环路，并网测试逆变器模式		controlCARD + “硬件在环”下的仿真功率级
5	PFC：三相交流电源，直流电阻负载，开环检查	1. 检查 vGridRms、iGridRms、vBus 测量值是否正确 <ul style="list-style-type: none"> <li>• 检查 PLL 是否已锁定</li> </ul>	controlCARD + 功率级硬件
6	PFC：闭合电流环路，电阻负载连接到直流电，三相交流电		controlCARD + 功率级硬件
7	PFC：闭合电压环路 + 电流环路，电阻负载连接到直流电，三相交流电		controlCARD + 功率级硬件

图 3-28 和图 3-29 介绍了实验 1 (开环测试) 和实验 3 (闭环测试) 之间的主要差异。

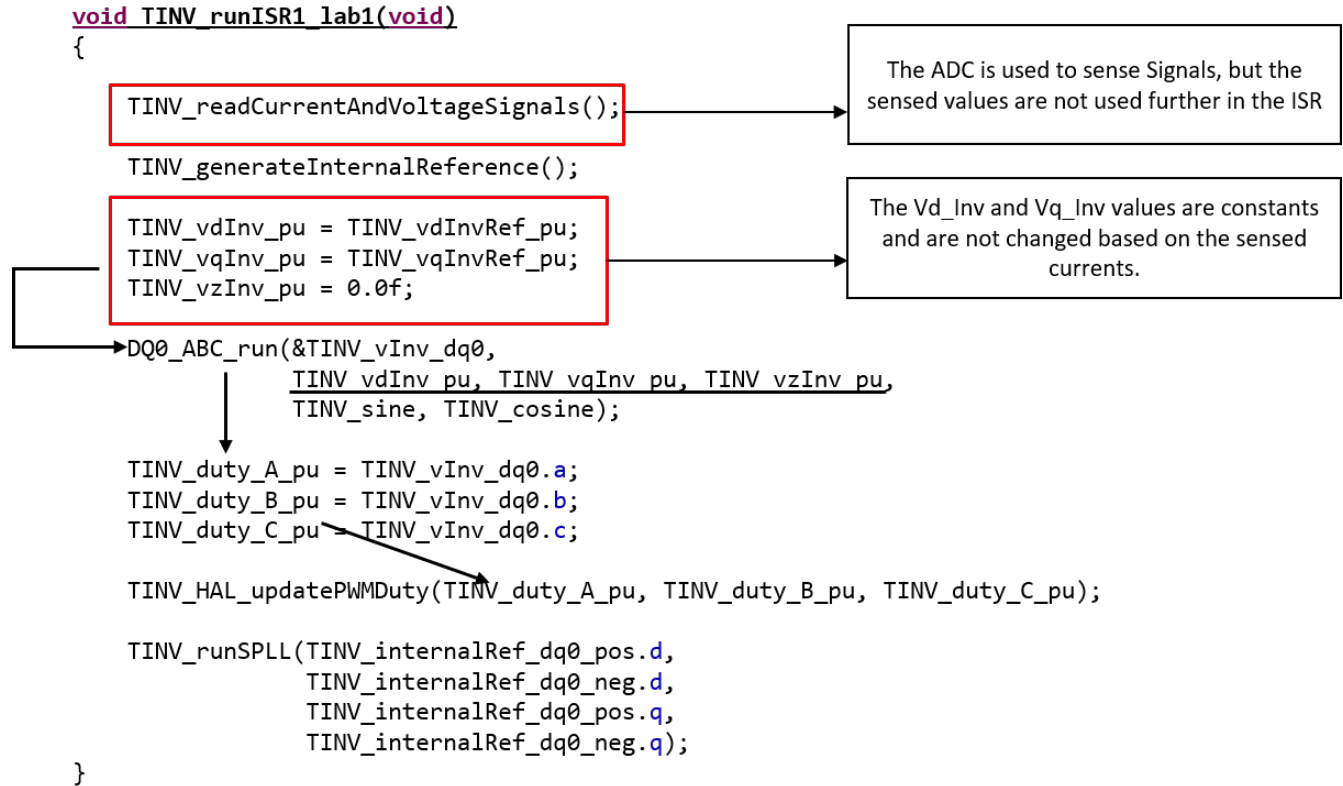


图 3-28. ISR1 中的实验 1

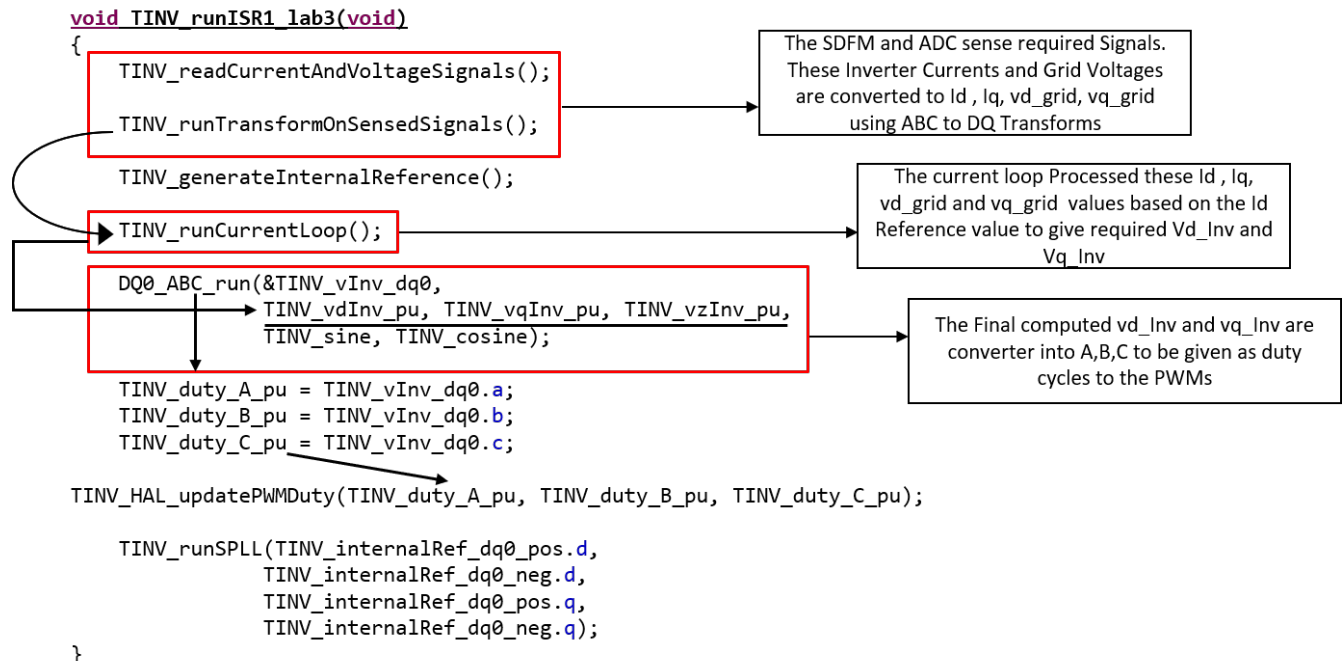


图 3-29. ISR1 中的实验 3

### 3.3.2 ISR1 - 实验 3

实验 3 - ISR1 的初始任务是检测 ADC 和 SDFM 信号。使用 *Inv* 定义在电路的逆变器侧检测到的信号，并使用 *Grid* 定义在电路的电网侧检测到的信号。使用 SDFM 模块检测三相电流  $I_a$ 、 $I_b$  和  $I_c$  并记录了值。电网侧和逆变器侧的电流值相同。因此，只有逆变器侧的值是通过 SDFM 检测，这些值等同于电网侧的值：

1.  $TINV\_ilnv\_A\_sensed\_pu = TINV\_iGrid\_A\_sensed\_pu;$
2.  $TINV\_ilnv\_B\_sensed\_pu = TINV\_iGrid\_B\_sensed\_pu;$
3.  $TINV\_ilnv\_C\_sensed\_pu = TINV\_iGrid\_C\_sensed\_pu;$

使用 ABC\_DQ 软件块将这些值转换为  $I_d$  和  $I_q$  (Clarke 和 Park 变换)。这些格式化的输入电网电流  $I_d$  和  $I_q$  以及参考  $I_d$  值用于计算所需的输出电网电压  $vd\_Inv\_out$  和  $vq\_Inv\_out$ ，如图 3-30 所示。

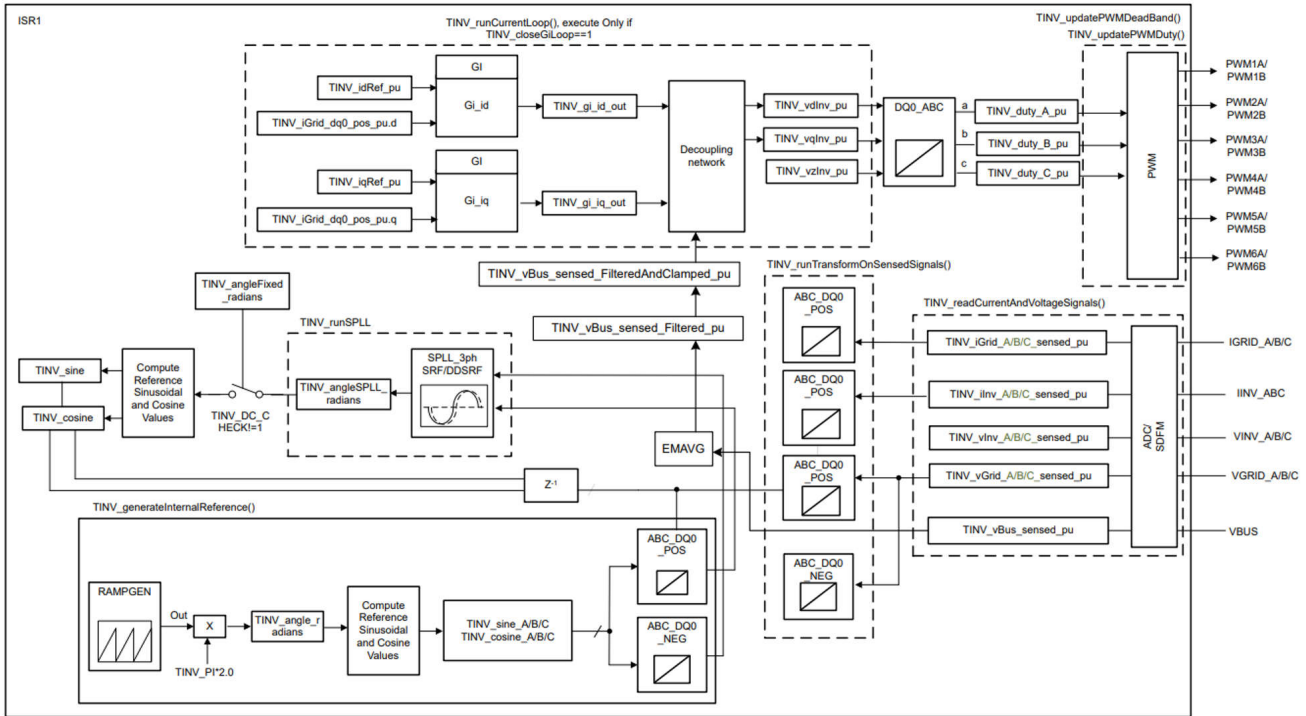


图 3-30. ISR1 软件流程方框图

默认情况下，电流  $I_d$  基准值为 0.005，对应于 500mA 的输出电流。要增加控制环路的总体电流输出，请增大  $I_d$  基准值。0.01 的  $I_d$  基准大致对应于 1A 的总输出电流。

通过应用 DQ\_ABC 软件块将  $vd\_out$  和  $vq\_out$  转换为三相值 (逆向 Park 和逆向 Clarke 变换)。获得的三个相位值被格式化为 PWM 波形的占空比。这些占空比值作为其各自 EPWM 的 CMPA 的输入，如以下列表所示。

1. Duty\_ticks\_A → TINV\_Q1\_Q3\_A\_PWM 和 TINV\_Q1\_Q4\_A\_PWM 的 CMPA
2. Duty\_ticks\_B → TINV\_Q1\_Q3\_B\_PWM 和 TINV\_Q1\_Q4\_B\_PWM 的 CMPA
3. Duty\_ticks\_C → TINV\_Q1\_Q3\_C\_PWM 和 TINV\_Q1\_Q4\_C\_PWM 的 CMPA

如果从 DQ\_ABC 变换中获得的 A 值为正值 (大于 0)，则表示正弦波的正半个周期。因此，启用 PWM 实例 TINV\_Q1\_Q3\_A\_PWM 并禁用实例 TINV\_Q1\_Q4\_A\_PWM 来执行正半周期的切换。

如果从 DQ\_ABC 变换中获得的 A 值为负 (小于 0)，则表示正弦波的负半个周期。因此，禁用 PWM 实例 TINV\_Q1\_Q3\_A\_PWM 并启用实例 TINV\_Q1\_Q4\_A\_PWM 来执行负半周期的切换。

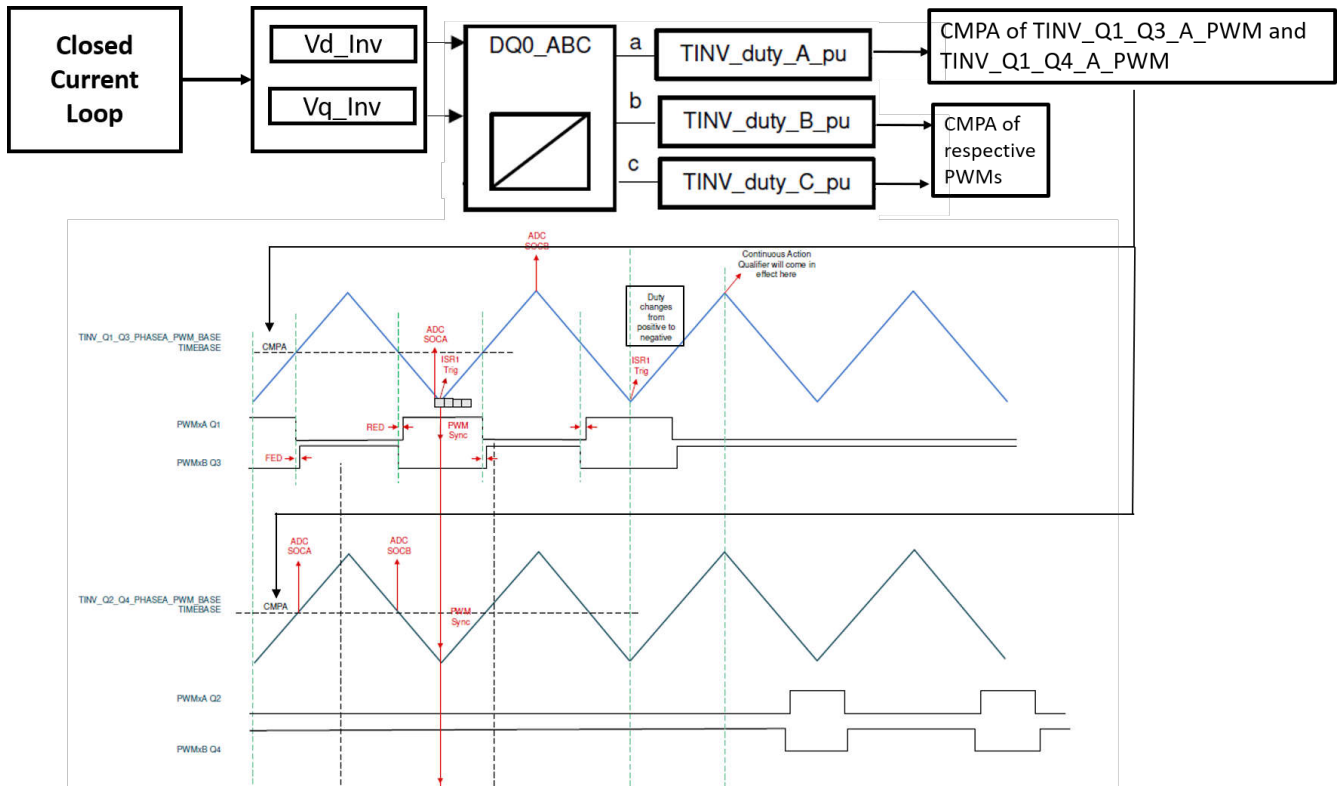


图 3-31. 通过 ISR1 进行 PWM 更新

在本演示中，使用 SRF 软件 PLL 锁定斜坡发生器的参考角。该 PLL 角度的正弦和余弦用于计算 ABC\_DQ 和 DQ\_ABC 变换。图 3-30 显示了方框图。

### 3.3.3 ISR2

ISR2 由 CPU 计时器 INT 触发，而 CPU 计时器 INT 由 CPU 计时器的溢出发起。ISR2 用于运行辅助控制功能，例如对电流和电压信号进行连续平均以消除噪声，以及为命令参考运行压摆率功能。图 3-32 描述了 ISR2 方框图。

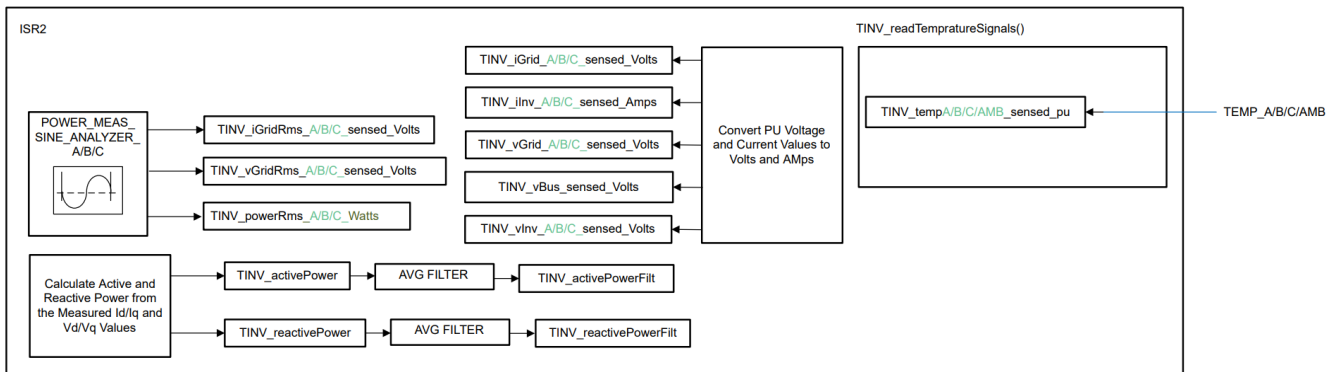


图 3-32. ISR2 软件流程方框图

### 3.4 保护方案

硬件电路中的敏感元件极少，尤其是栅极驱动器、MOSFET 等需要高压和电流保护的元件。为了以相同的功率驱动 MOSFET，使用了 TI 设计的栅极驱动器。硬件中有三块栅极驱动器卡，每块卡带有两个 ISO5852S 和两个 UCC5320 栅极驱动器。这些栅极驱动器在短路期间提供软关断 (STO) 功能，并在检测到 MOSFET 去饱和时在 FLT 上发出故障报警。有关栅极驱动器的更多信息，请参阅 [TIDA-01606 : 10kW 双向三相三级 \(T 型\) 逆变器和 PFC 参考设计](#) 设计指南。

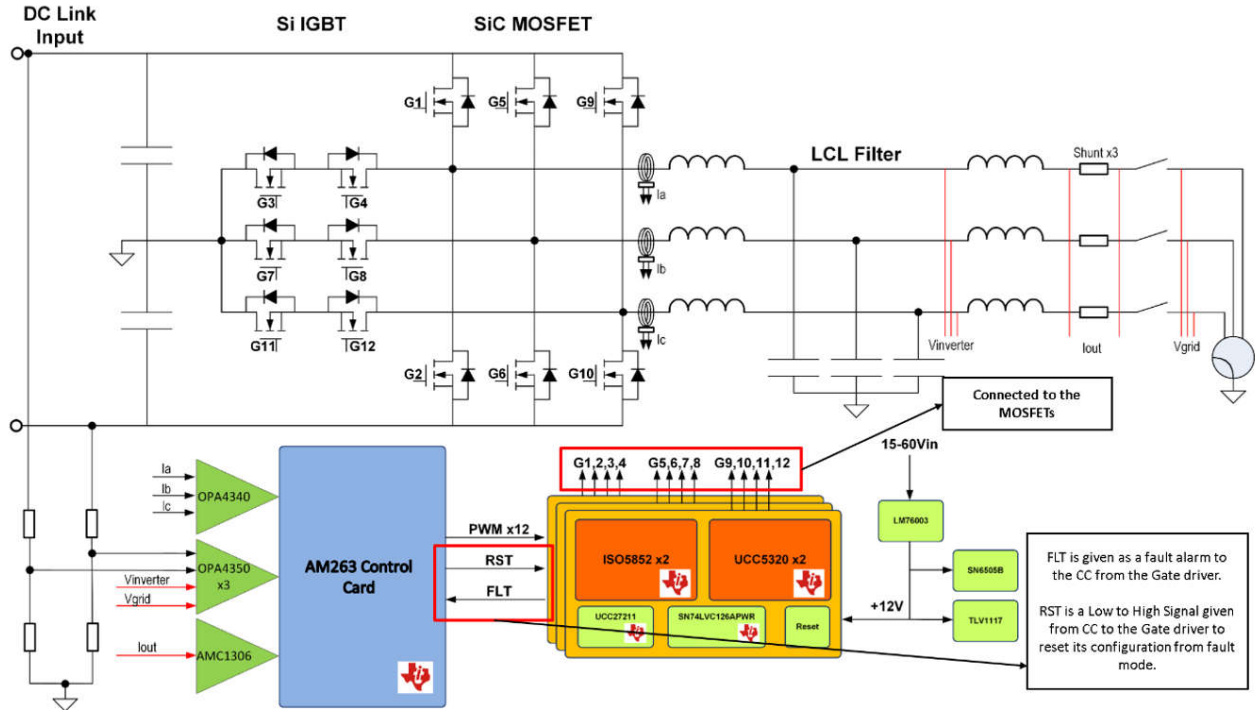


图 3-33. 具有栅极驱动器故障检测功能的 TIDA-01606 方框图

表 3-7 列出了用于过压保护、过流保护和栅极驱动器故障指示的跳闸配置，以便保护 MOSFET 免受任何过压、过流或浪涌事件的影响。除此之外，MOSFET 连接到散热器，可在高电压下运行时防止电路中出现高温。

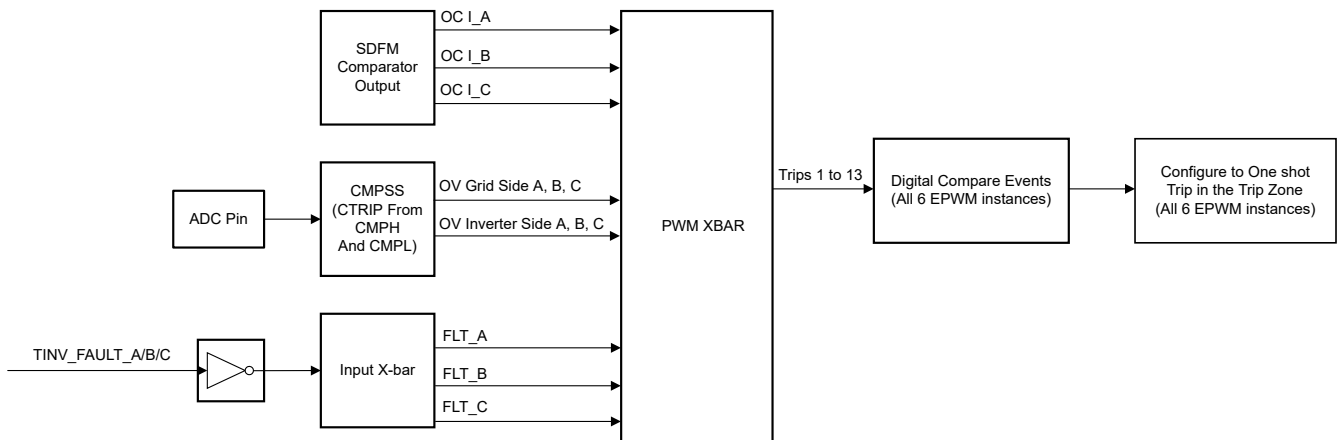


图 3-34. 板保护方框图

表 3-7. PWM XBAR 与跳闸的对应关系

跳闸原因	PWM_XBAR ( PWM 交叉开关 )	跳闸编号 ( 数字比较 - EPWM )
FLT_A (GateDriver_A)	PWM XBAR0	Trip1
FLT_B (GateDriver_B)	PWM XBAR1	Trip2
FLT_C (GateDriver_C)	PWM XBAR2	Trip3
InvSide_OverVoltage_A	PWM XBAR3	Trip4
InvSide_OverVoltage_C	PWM XBAR4	Trip5
GridSide_OverVoltage_C	PWM XBAR5	Trip6
InvSide_OverVoltage_B	PWM XBAR6	Trip7
GridSide_OverVoltage_B	PWM XBAR7	Trip8
GridSide_OverVoltage_A	PWM XBAR8	Trip9
OverCurrent_A	PWM XBAR10	Trip11
OverCurrent_B	PWM XBAR11	Trip12
OverCurrent_C	PWM XBAR12	Trip13

默认情况下，跳闸 4 至跳闸 13 连接到一次性跳闸区域，以实现立即跳闸。从跳闸 1 到跳闸 3 ( 栅极驱动器的故障检测 ) 在数字比较事件中被禁用。要将其启用，请检查 SysConfig 中 EPWM 数字比较子模块的组合跳闸输入。

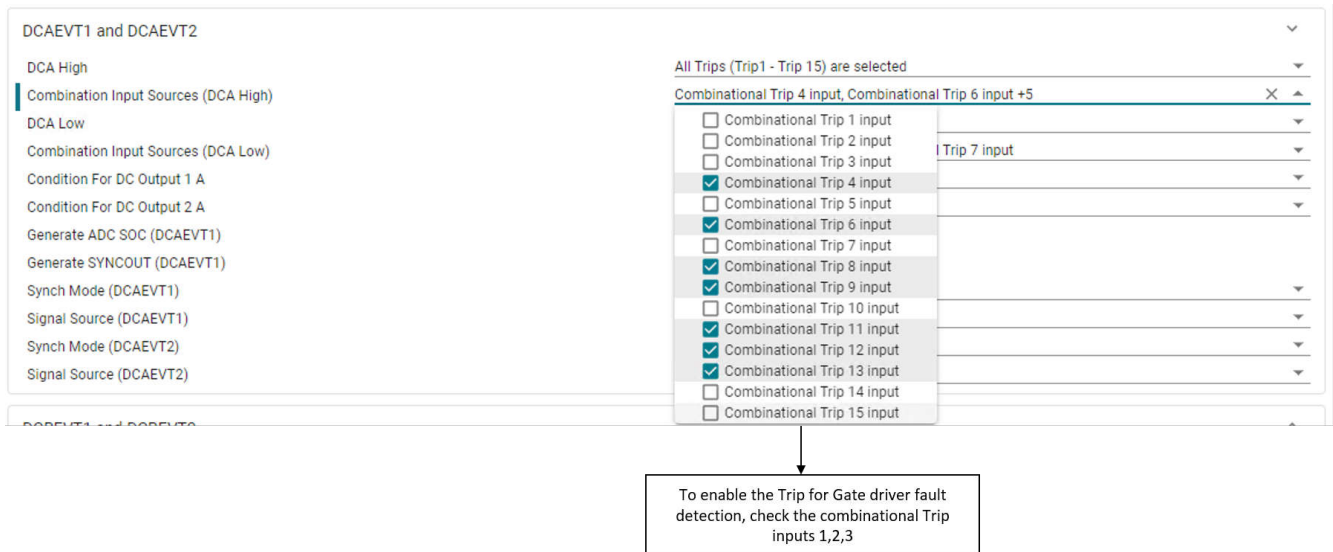


图 3-35. 针对栅极驱动器故障检测的跳闸使能

`TINV_updateFaultStaus()` 函数将在缓慢的后台任务中被定期调用以更新跳闸标志并在需要时复位锁存器。如果发生跳闸事件，PWM 需要单独清除跳闸标志。这部分通常在 ISR 中通过调用 `TINV_clearPWMTrips()` 来处理。

B1 任务会更新系统的故障状态。通过将宏 `TINV_PROTECTION` 设置为 `TINV_PROTECTION_ENABLED`，可启用此设置。默认情况下，实验 1 已禁用该功能，其他实验已启用该功能。有关这方面的更多信息，请参阅节 3.4。故障更新会持续监控连接到比较器跳闸输出、输入 XBAR 跳闸和 SDFM 比较器跳闸输出的 PWM XBAR。当在这些部分中记录任何跳闸时，会相应地更新 `TINV_boardFaultFlags` 结构。如果设计人员在输出端未看到任何 PWM 波形，此功能有助于进行调试。表 3-8 中的功能可指示过压、过流和栅极驱动器故障状态。

表 3-8. 软件中的故障 API

故障	API
过流故障 - 逆变器和电网侧	<code>TINV_HAL_get_iInvA_overCurrentFlag</code> <code>TINV_HAL_get_iInvB_overCurrentFlag</code> <code>TINV_HAL_get_iInvC_overCurrentFlag</code>

表 3-8. 软件中的故障 API (continued)

故障	API
过压故障 - 逆变器侧	TINV_HAL_get_VInvA_overVoltageFlag TINV_HAL_get_VInvB_overVoltageFlag TINV_HAL_get_VInvC_overVoltageFlag
过流故障 - 电网侧	TINV_HAL_get_VGridA_overVoltageFlag TINV_HAL_get_VGridB_overVoltageFlag TINV_HAL_get_VGridC_overVoltageFlag
栅极驱动器故障	TINV_HAL_get_faultAFlag TINV_HAL_get_faultBFlag TINV_HAL_get_faultCFlag

另一种调试跳闸状态的方法是在 Memory Browser 中观察寄存器 0x52000C18 PWMXBAR 状态。通过识别此寄存器的位来确定特定跳闸 (在表 3-9 中所示的位置设置了这些位)。

表 3-9. 不同系统故障的 PWM XBAR 位位置

PWM_XBAR	PWM XBAR 输出 状态寄存器位映射
PWMXBAR_FLT_A	0
PWMXBAR_FLT_B	1
PWMXBAR_FLT_C	2
PWMXBAR_Inv_OverVoltage_A	3
PWMXBAR_Inv_OverVoltage_B	6
PWMXBAR_Inv_OverVoltage_C	4
PWMXBAR_Grid_OverVoltage_A	8
PWMXBAR_Grid_OverVoltage_B	7
PWMXBAR_Grid_OverVoltage_C	5
PWMXBAR_OverCurrent_A	10
PWMXBAR_OverCurrent_B	11
PWMXBAR_OverCurrent_C	12

### 3.5 CPU 负载

当内核 R50\_0 以 400MHz 运行时，实验 3 的主控制 ISR 需要大约 2.5  $\mu$ s 的时间来执行。此实验 3 每 20  $\mu$ s 执行一次。这会占用大约 12.5% 的 CPU 时间。节 5 提及了 ISR1 在所有实验中的 CPU 负载。

### 3.6 构建、加载和调试固件

若要构建工程，右键点击工程名称，然后点击 *Rebuild Project* (重建工程)。随后将成功构建工程。

要加载工程，首先确保在 Project Explorer 中的 targetConfigs (\*.ccxml 文件) 下将正确的目标配置设置为“Active”。然后，点击 *Run* → *Debug* (运行 → 调试) 以启动调试会话。对于双 CPU 器件，可能会显示一个窗口，用户可以在其中选择要对哪个 CPU 执行调试。在本例中，选择“R50\_0”。然后工程加载到器件上，同时 CCS 调试视图变为活动状态。代码在主例程开始时停止。在不同的实验中，有时需要通过查看图形窗口中的数据来验证所测量的电流和电压或控制变量。要将图形导入 CCS 视图，请选择 *Tools* → *Graph*。

## 4 实施优化

在逆变器演示中，使用以下技术来优化 ISR 的执行时间：

- 构建设置更改为带有优化的释放模式。
  - 释放模式具有可根据要求进行配置的优化。
  - 可用的选项：`O[0 | 1 | 2 | 3 | fast | g | s | z]`
  - 有关更多信息，请参阅 [编译器工具用户手册](#) 的 [优化选项](#) 部分。
- 将代码添加至 TCM 以加快执行速度。TCM 具有低延迟，通过 TCM 访问的指令占用一个 CPU 周期。
  - 在 TCM 中执行 ISR 以加快代码创建速度。ISR 内执行的所有变量均为全局变量并添加到 TCM 中。
  - 自定义节添加到 linker.cmd 文件的 TCM 部分中。变量是添加到单独 TCM 部分的函数，可避免冲突。
    - `.controlfnc : {} palign(8) > R5F_TCMA`
    - `.controldata : {} palign(8) > R5F_TCMB`
  - 在函数原型中使用 `__section__("SECTION_NAME")` 属性，以将其添加到 TCM 中。
  - `always_inline` 属性用于强制 API 进入内联函数。
  - [图 4-1](#) 列出了实验 3 的 TCM 存储器消耗

MEMORY CONFIGURATION						
name	origin	length	used	unused	attr	fill
R5F_VECS	00000000	00000040	00000040	00000000	RWIX	
R5F_TCMA	00000040	00007fc0	00007d10	000002b0	RWIX	
R5F_TCMB	00080000	00008000	000014e8	00006b18	RWIX	
FLASH	60100000	00080000	00000000	00080000	RWIX	
OCRAM	70040000	00040000	0000e3d0	00031c30	RWIX	
USER_SHM_MEM	701d0000	00004000	00000000	00004000	RWIX	
LOG_SHM_MEM	701d4000	00004000	00000000	00004000	RWIX	
RTOS_NORTOS_IPC_SHM_M	72000000	00003e80	00000000	00003e80	RWIX	

图 4-1. 启用实验 3 时的 TCMA 和 TCMB 存储器消耗

- 三角计算被 `r5_math_Trig` 库函数取代。
  - MCU\_PLUS\_SDK 具有 `r5_math_Trig` 库，与标准 `mathlib` 相比，该库具有优化的三角运算。
  - `sinf(TINV_angleSPLL_radians) → ti_r5fmath_sin(TINV_angleSPLL_radians, ti_r5fmath_Plconst, ti_r5fmath_sinCoef)`
  - 一次 `sinf` 计算大约消耗  $0.45 \mu s$  的执行时间。这可以用高级三角函数库 API 取而代之，后者进行一次正弦运算消耗  $0.12 \mu s$ 。
- 如果相邻的 16 位 ADC 读操作转换为 32 位读操作且值被划分为 16 位，则进行两次 16 位 ADC 读操作。
  - 妥善做法是在一组中配置 ADC 检测功能。例如，`CSL_ADC_RESULT_ADCRESULT0` 和 `CSL_ADC_RESULT_ADCRESULT1` 可以同时配置和读取。
  - 寄存器上的读取操作成本高昂，该方法将读取操作减少了一次。

```
#define CSL_ADC_RESULT_ADCRESULT0 (0x00000000U)
#define CSL_ADC_RESULT_ADCRESULT1 (0x00000002U)
#define CSL_ADC_RESULT_ADCRESULT2 (0x00000004U)
#define CSL_ADC_RESULT_ADCRESULT3 (0x00000006U)
```

- 修改 ADC 和 PWM 寄存器空间的 MPU 配置以启用已发布的写入。这会配置 MPU 区域访问控制寄存器位分配。
  - 所有寄存器写入操作默认严格排序，也就是说，数据首先写入寄存器，并且 CPU 会等待 `Ack` 成功写入然后再进行下一次写入。
  - 删除严格排序配置将启用连续写入操作。根据要求，可以将配置设置为可缓存或不可缓存。通常，为加快执行速度，优选不可缓存，因为不会占用 CPU 周期来缓存数据。
  - 这可以与 DMA 相结合，以便及时发送多个寄存器数据（在本文档发布时，尚未实现此功能）。
  - 有关更多信息，请参阅 [Cortex-R4F 技术参考手册](#) 的 MPU 寄存器部分。
- 更多编译器优化：



- Arm® Cortex® - R5 将涉及浮点的算术和逻辑计算卸载到 FPU，从而提高速度和运算能力。选择浮点 *abi* 作为硬配置，以使用 FPU 硬件进行运算，而不是进行软件计算。

Select assumed floating-point ABI (-mfloat-abi)	hard	▼
Select floating-point hardware processor (-mfpu)	vfpv3-d16	▼

<input checked="" type="checkbox"/> Select Link-Time Optimization (LTO) (-flto)
---

图 4-2. 编译器优化设置

- 链接时优化：
  - 将编译器优化配置为 TIARM Clang 2.1 LTS 中支持的 `-flto` (链接时优化 - LTO)。
  - 要访问 `-flto` 功能，需要将编译器配置为 CCS 工程中受支持的版本。
  - 依次转到“Project Properties” → “Arm Compiler” → “Optimizations” → “Select Link-Time Optimization Option (LTO)”，如图 4-2 中所示。
  - 启用设置并重新构建工程后，请检查 `ItoN` 选项的映射文件，它表示正在使用 LTO 选项的位置。
  - 映射文件还表明，`.rodata` 输出段在支持 LTO 的 `-Oz` 构建中要小得多。
  - 有关更多详细信息，请参阅 [编译器工具用户手册](#) 中的 [链接时优化](#) 部分。

## 5 测试和结果

### 5.1 实验 1

实验 1 可在 AM263x controlCARD 和集线站上运行。若要选择实验 1，请更改以下代码中所示的宏。目前，所有其他选项都可以保留默认值。

```
#define TINV_LAB 1
```

将 controlCARD 集线站的 EPWM0、EPWM1、EPWM2、EPWM3、EPWM6、EPWM7 引脚连接到逻辑分析仪以查看 PWM 波形。

- EPWM0 - 引脚 49、51
- EPWM1 - 引脚 53、55
- EPWM2 - 引脚 50、52
- EPWM3 - 引脚 54、56
- EPWM6 - 引脚 58、60
- EPWM7 - 引脚 62、64

图 5-1 涵盖了全部六个 EPWM\_A 的 PWM 波形。



图 5-1. 实验 1 EPWM 输出波形

要测试 ADC 和 SDFM 检测，请将采集的 ADC 结果寄存器或 SDFM 滤波器寄存器输出值发送到 AM263x DAC，同时将 controlCARD 置于 TIDA-01606 设置中。在图 5-2 中，绿色线表示输出通道 A，粉色线表示通道 A 检测到的电流的 SDFM 滤波输出。

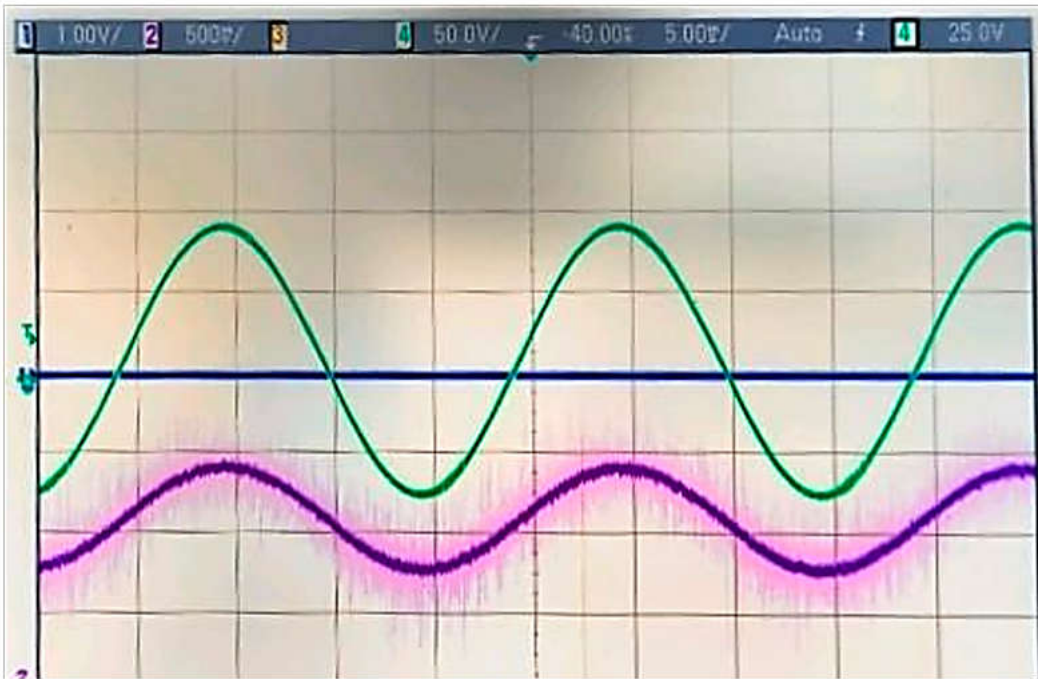


图 5-2. 使用 DAC 在示波器上绘制的 SDFM 通道 A 检测到的信号

## 5.2 测试逆变器运行情况

### 5.2.1 实验 2 和实验 3

实验 2 是开环逆变器运行模式，实验 3 是电流闭环逆变器运行模式。在端子 J1 和 J2 之间施加高压 (800VDC)。15V 辅助电源连接到端子 J33。在端子 J3、J4 和 J31 之间连接三相星形连接式电阻负载。J32 是未连接到负载的中性端子。

运行实验 3 时，请考虑以下步骤。要获取更多信息，请参阅 [TIDA-01606 : 10kW 双向三相三级 \(T 型\) 逆变器和 PFC 参考设计](#) 设计指南。

- 要选择实验 3，请更改如下所示的宏。目前，所有其他选项都可以保留默认值。

```
#define TINV_LAB 3
```

- SFRA 未集成在 AM263 中；因此，需要实时手动调整电流环路系数。用于运行控制环路的电流补偿器系数如以下代码中所示。修改这些系数以满足必要的环路带宽和相位裕度。用于电阻负载的理想系数与用于并网的系数略有不同，因为电网阻抗非常低。

```
//  
// PI Controller Settings from Compensation Designer  
//  
#define TINV_GI_PI_KP ((float32_t)0.14) //Inverter mode Kp for current loop  
#define TINV_GI_PI_KI ((float32_t)0.1120860479) //Inverter mode Ki for current loop  
  
//By default, Id reference:  
#define TINV_IREF_DEFAULT ((float32_t)0.005f)  
TINV_idRef_pu = TINV_IREF_DEFAULT
```

- 首先设置一个大约 200Ω 的适当电阻负载，但尽管也可以在空载条件下启动逆变器模式。将直流母线电压 Vbus 缓慢斜升至 800V。设置 `TINV_clearPWMTrip = 1` 以清除 PWM 跳闸信号。现在开关操作将开始，并且正弦电压开始出现在输出端。此时，辅助电源汲取的电流接近 500mA。
- `TINV_closeGiLoop` 变量已启用，闭合电流环路操作开始。`TINV_idRef_pu` 是电流命令参考，默认情况下，此参考在启动时填充为 0.005pu 的值。缓慢改变此值以提高输出交流电压，并观察测量电流跟踪指令值。

- 在继续闭合实验 3 中的电流环路之前，验证 watch window 中的  $TINV\_idRef\_pu$  和  $TINV\_ilnv\_dq0\_pos.d$  数据是否处于低电平。在 800V 输入电压下将  $id\_ref$  缓慢增加至 0.36pu，以提高输出功率。
- 图 5-3 显示了此测试设置。

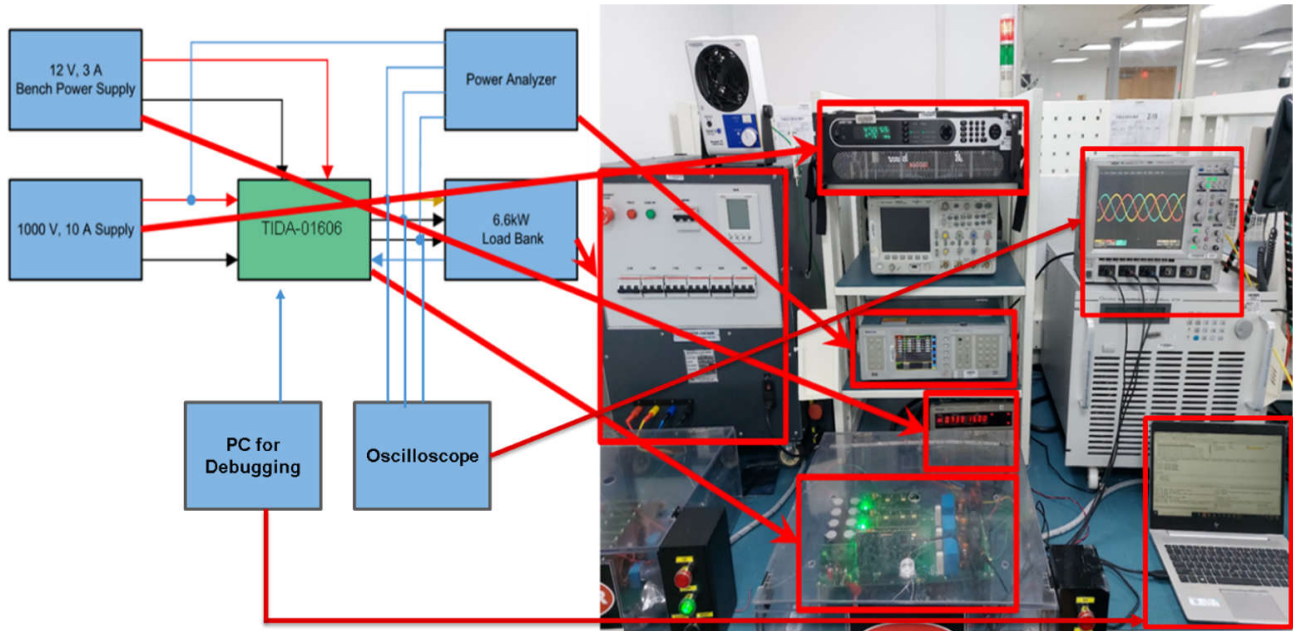


图 5-3. TIDA-01606 在实验中的测试设置

图 5-4 显示了示波器在三个阶段的输出。

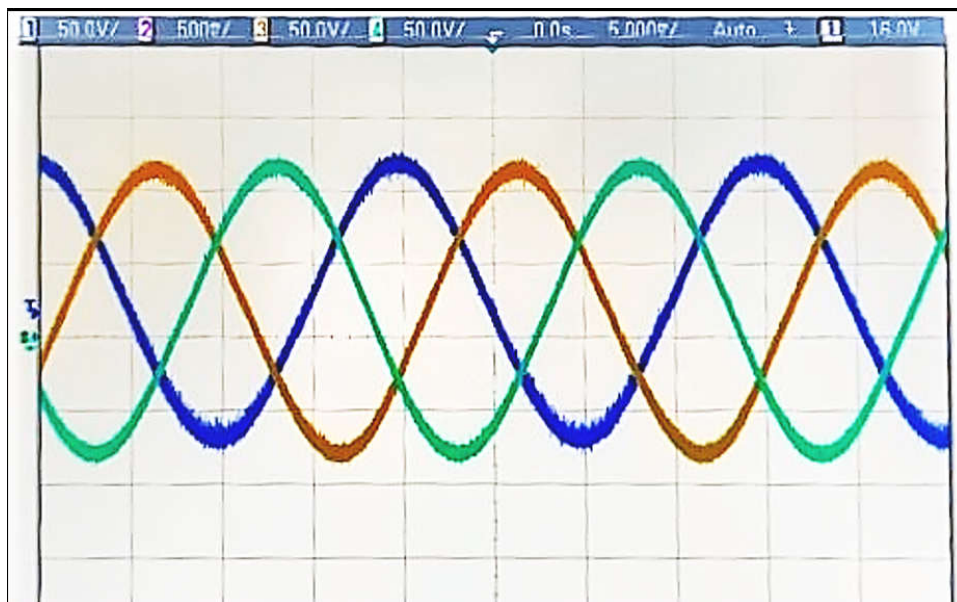


图 5-4. 实验 3 的输出波形 A、B、C (从 TIDA-01606 电网侧)

降低频率以验证电流环路的稳定性。输入电压现在为 400V，电压快速降至 100V。输出电流突然降低，但电流环路调整了输出波形，使输出电流恢复到基准值，如图 5-5 所示。这里，电流基准值  $TINV\_idRef\_pu$  为 0.01，对应于 1A 电流。

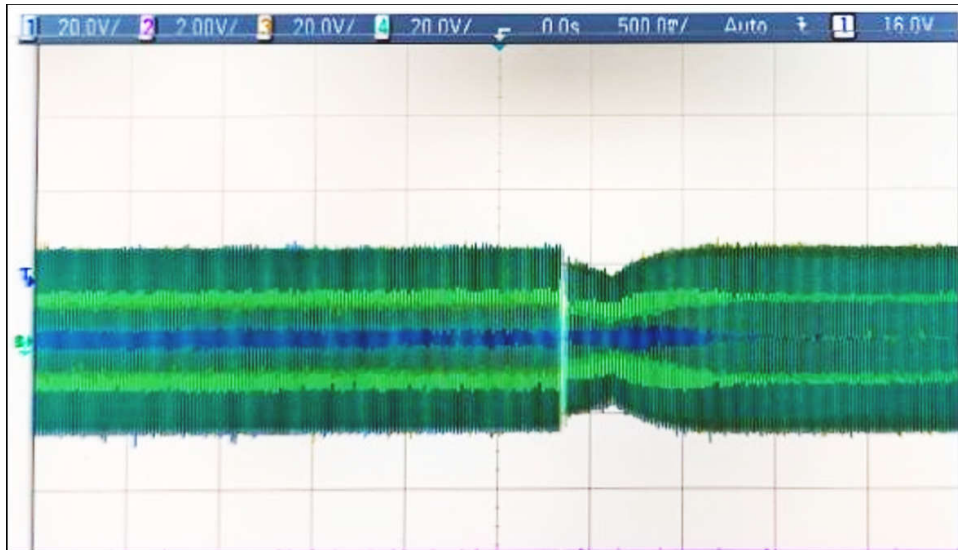


图 5-5. 实验 3 的控制环路测试

### 5.2.2 实验 3 性能

表 5-1 中显示了实验 3 串式逆变器的效率和总谐波失真 (THD) 值。

表 5-1. 中断操作中的 THD 值

输出功率 ( 瓦特 )	效率 (%)	电流 - THD ( A 相 )	电流 - THD ( B 相 )	电流 - THD ( C 相 )
982	96.6	1.78	1.63	1.58
1548	97.3	1.35	0.91	0.83
2576	97.6	1.02	0.82	0.75
3822	97.9	0.61	0.74	0.62

### 5.2.3 逆变器中断基准测试

使用 SDK 中的 CPU cycleP\_Counter 功能测试了实验 1、实验 2 和实验 3 中 ISR1 的执行时间。与使用 GPIO 切换来计算 ISR 执行时间相比，这种方法可以提供准确的结果，因为 GPIO 切换具有延迟。

表 5-2. ISR1 最长和最短 CPU 执行时间计算

实验编号	CPU 周期数 - 最大值	消耗的 CPU 时间 - 最长	CPU 周期数 - 最小值	消耗的 CPU 时间 - 最短
实验 1	860	2.15 $\mu$ s	829	2.072 $\mu$ s
实验 2	735	1.84 $\mu$ s	708	1.77 $\mu$ s
实验 3	936	2.34 $\mu$ s	890	2.225 $\mu$ s

### 5.2.4 实验 4

在实验 4 中，逆变器实时连接到电网。此实验未通过连接到电网进行测试。

### 5.3 测试 PFC 运行情况

实验 5 到实验 7 涉及开环和闭环 PFC 运行。实验 5 是开环 PFC 运行模式，实验 6 是电流闭环 PFC 运行模式。实验 7 是闭合电压和电流环路中的 PFC 运行模式。

前馈和去耦功能在 ISR1 内部实现，并为所有使用电流环路的 PFC 实验添加。因此，对于 PFC 模式，这是在实验 6 和实验 7 中完成的。对于此前馈和去耦功能，将滤波后的直流总线电压与用户定义的最小总线电压进行比较，以计算钳位滤波后的直流总线电压。这也是在 ISR1 中完成的。这个钳位滤波后的直流母线电压和电流控制器输出最终用于实现前馈和去耦功能。添加了过流和过压保护。

图 5-6 展示了硬件设置，其中的直流端子 J1 和 J2 连接到电阻负载。15V 辅助电源连接到端子 J33。在端子 J3、J4 和 J31 (A、B 和 C) 之间连接三相交流电源。J32 是中性端子。

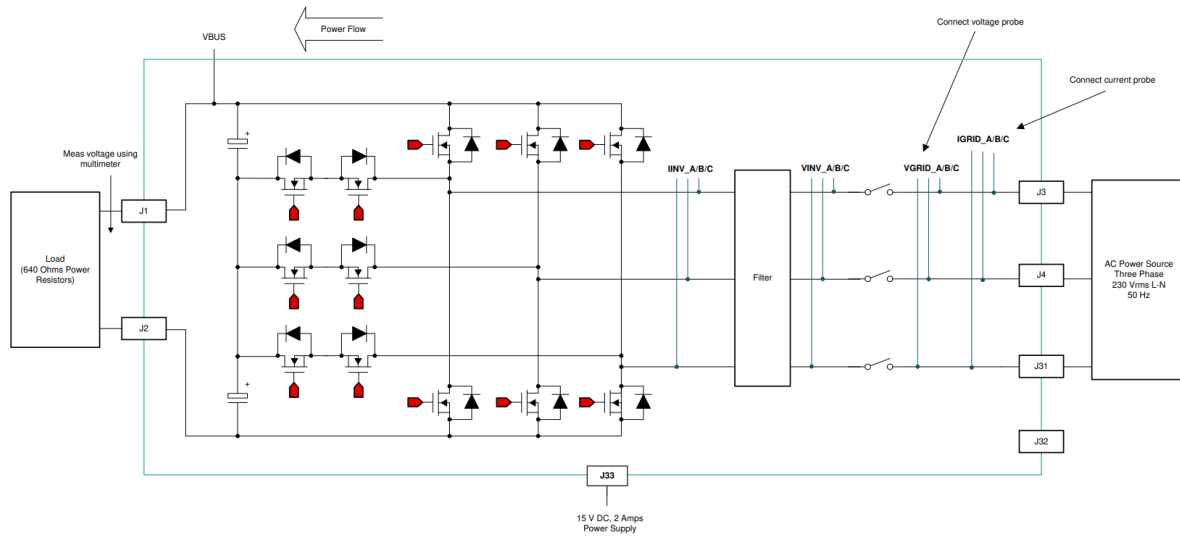


图 5-6. PFC 模式测试设置

### 5.3.1 实验 5

本实验会检查检测功能，并且在 `clearPWMTrip` 设置为 1 之前不会发生开关操作。通过更改用户设置文件中的 `LAB_NUMBER` 将工程设置为实验 5。在这种情况下，转换器作为整流器运行，可以观察到在无任何功率因数校正的情况下汲取的整流电流。还可以在此构建中安全地验证 SPLL 锁定。确保正确指定电网频率，可以使用 `TINV_AC_FREQ_HZ` 宏更改电网频率。继电器在初始化期间关闭。图 5-7 展示了实验 5 ISR1 的工作流程。ISR2 与前面的实验类似。

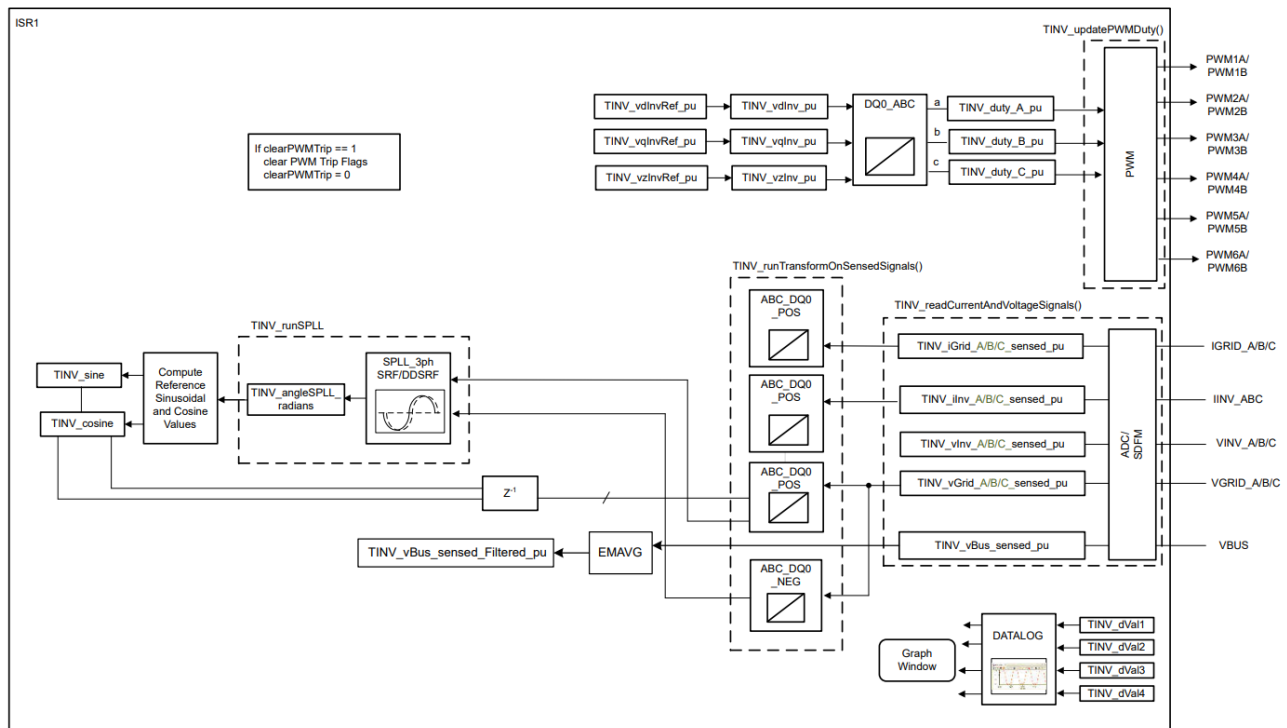


图 5-7. 实验 5 流程图

若要验证实验 5 中的升压操作，请按照提供的顺序执行以下步骤：

- 开启辅助电源并将其设置为 15V，然后调试并运行代码。
- 将负载连接到 J1 和 J2 端子。确保使用高负载电阻（约 2kΩ）。低电阻负载会导致高浪涌电流并触发过流保护。
- 向三相端子施加 30V<sub>RMS</sub> 交流电压。
- 将所需变量添加到观察窗口以更改运行时中的值。在观察窗口中添加的值只能通过暂停 R5 内核来改变。
- 设置 `TINV_startPowerStage = 1` 和 `TINV_closeGiLoop = 1`。
- 由于浪涌电流，过流检测逻辑会使 PWM 跳闸，并设置跳闸。
- 通过将 `TINV_clearPwmTrip` 设置为 1 来清除 PWM 跳闸以查看直流电压轻微升高的情况。
- 在 PFC 操作开始之前，由于 Vbus 上的负载，将汲取整流电流。一旦 `clearPWMTrip` 设置为 1，就会看到直流电压略微升高。

使用非常低的交流输入开始实验。一旦 `clearPWMTrip` 成功清除浪涌过流跳闸，就会缓慢地将交流输入逐步增加到所需的值。避免因交流输入突然增加而导致 PWM 跳闸。

### 5.3.1.1 Memory Browser - 持续刷新

如需在运行时持续监控存储器区域而不暂停代码，请使用 Memory Browser。执行以下步骤进行持续监控：

1. 按照 MCU API 指南 [CCS\\_Target\\_Launch](#) 中的说明启动 AM263x controlCARD 的 .ccxml 文件
2. 在 *Debug* 窗口中找到 AM263X 目标配置。右键单击 .ccxml 并选择 *Show all cores*。
3. 连接到 R50\_0 内核并加载工程的 .out 固件映像。在继续执行之前，首先执行剩余的步骤。

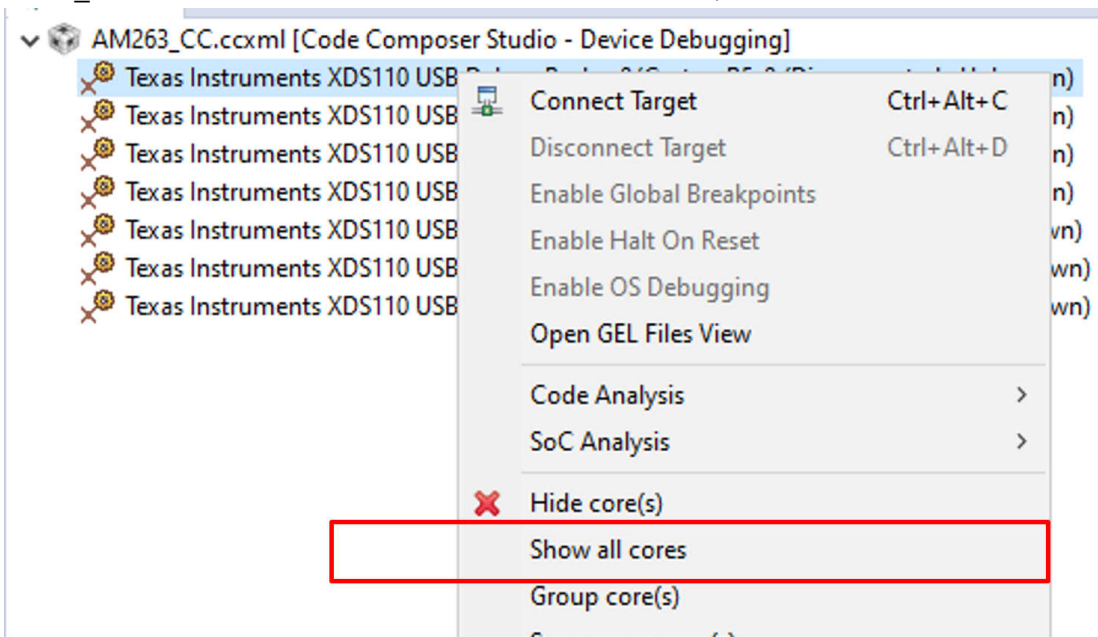


图 5-8. 显示 AM263x 中的所有内核

4. 转至 *View* 菜单项并打开 *Memory Browser*。选择 DAP 并将所选 DAP 固定到“Memory Browser”窗口。点击 *System\_View* 并选择 *Continuous Refresh Option*（请参阅图 5-10）以进行持续监控，然后运行代码。这允许在运行时调试存储器区域，而无需暂停 R5 内核。

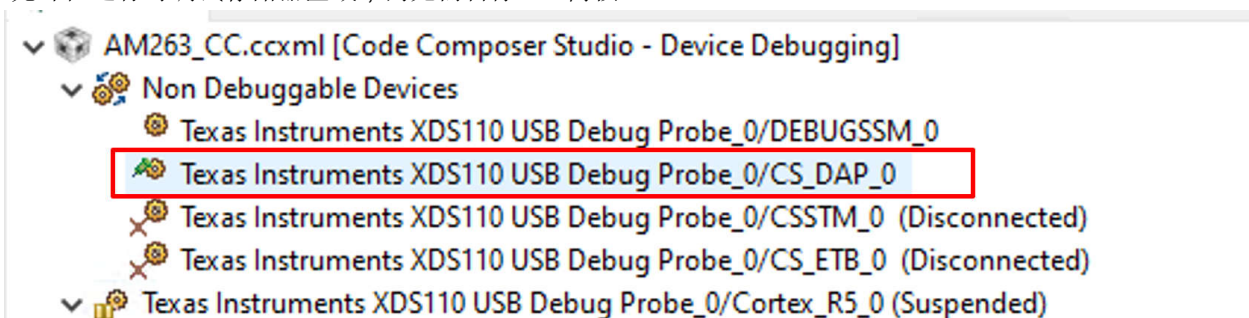


图 5-9. 将 CS\_DAP\_0 固定到 Memory Browser

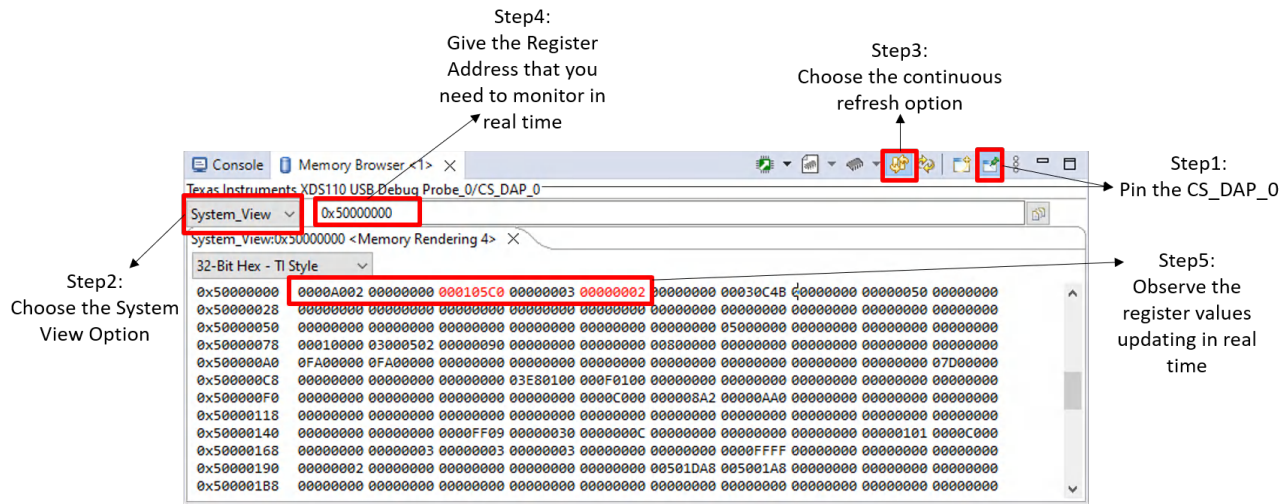


图 5-10. 已固定 CS\_DAP\_0 用于实时调试的 Memory Browser

- 如果 TCM 中存在变量，则使用 `attribute` 关键字将该变量放置在 OGRAM 中，以使其在 Memory Browser 中可见。例如，若要在 Memory Browser 中查看 `TINV_startPowerStage` 状态变量，请按以下方式定义该变量：

```
volatile int32_t TINV_startPowerStage __attribute__((__section__(".DebugData")));
```

可以在 Memory Browser 中监视和修改这些变量，如图 5-11 和图 5-12 所示。

Expression	Type	Value	Address
(x)= TINV_closeGiLoop	int	1	0x7004C5BC
(x)= TINV_startPowerStage	int	1	0x7004C5C0

图 5-11. “Expressions” 窗口中的 Gi 环路和功率级状态

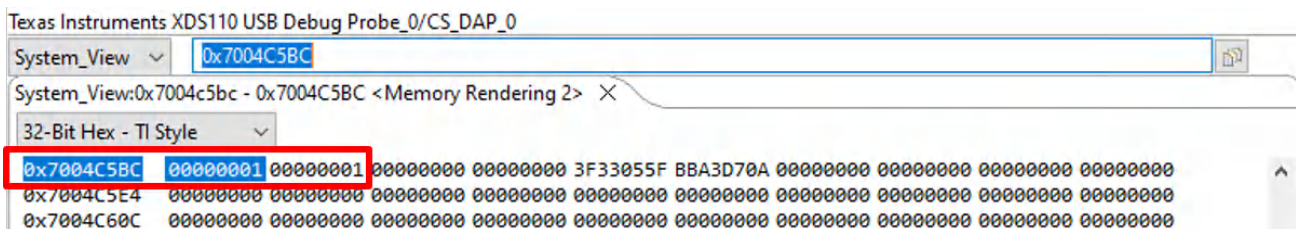


图 5-12. Memory Browser 中的 Gi 环路和功率级状态

- 如果未实时刷新值，请暂停内核，并尝试禁用数据缓存。为此，请转至 `Tools` → `ARM Advanced Features`，并取消选中 `Data Cache Enabled` 选项。然后恢复内核执行，可查看实时存储器更新。

### 5.3.2 实验 6

通过更改用户设置文件中的 `LAB_NUMBER` 将工程设置为实验 6。在该条件下，转换器作为整流器运行。在此实验中，`ISR1` 包含电流闭环和电压开环 PFC 运行。此外，将 `TINV_idRef_pu` 变量定义为负号表示 PFC 工作模式，定义为正号表示逆变器工作模式。可以在此实验中验证 SPLL 环路的锁定。图 5-13 说明了此实验的工作流程。



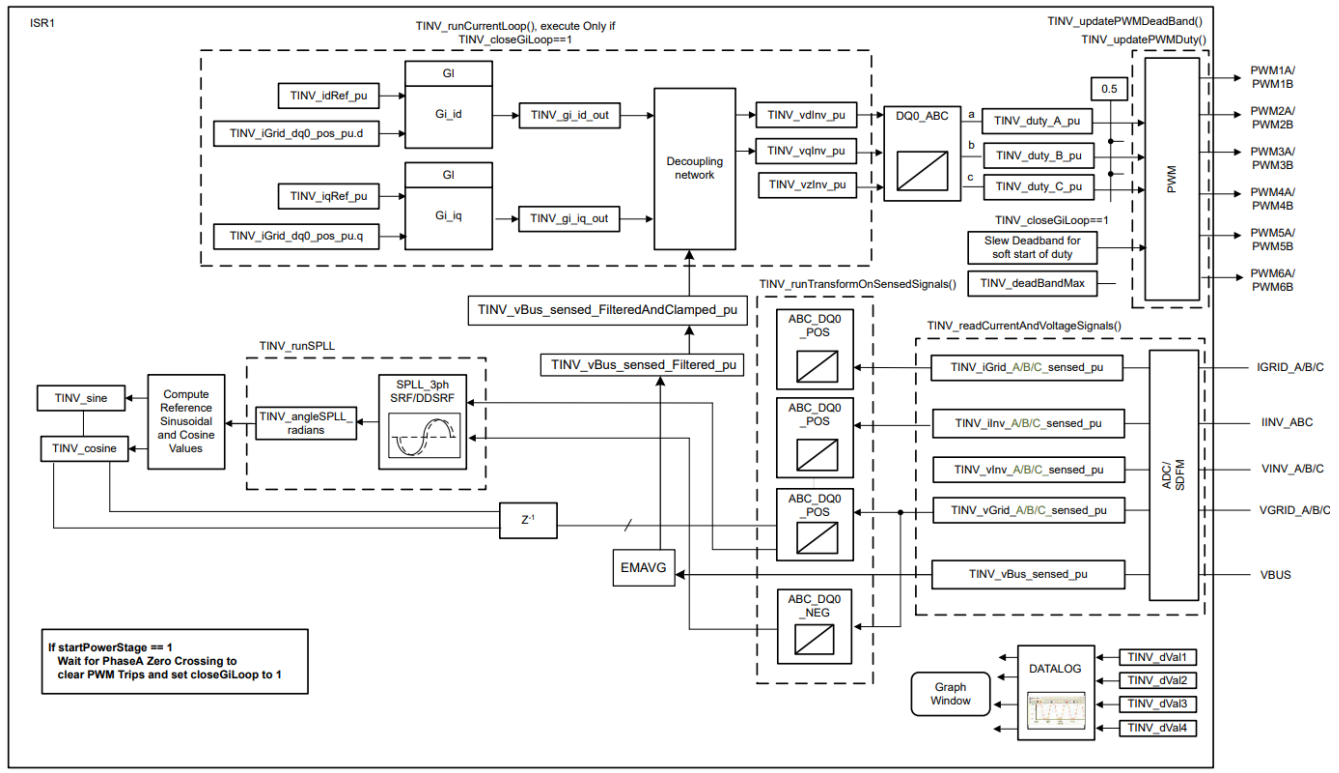


图 5-13. 实验 6 流程图

在此实验中，最佳  $K_p$  和  $K_i$  系数定义如下。用户可修改这些系数以满足必要的环路带宽和相位裕度。

```
#define TINV_GI_PI_KP ((float32_t)0.3)
#define TINV_GI_PI_KI ((float32_t)0.0120860479)
```

要验证实验 6，请按以下步骤操作：

- 打开辅助电源，将其设置为 15V 并加载代码。执行代码时，继电器在初始化时默认处于开启状态。
- 将负载电阻设置为高值（例如 3.18k $\Omega$ ）。将交流输入电压设置为 30V<sub>RMS</sub>，并进行适当的电流限制。
- 下面描述了在 Memory Browser 或 Watch 窗口中设置变量的顺序。设置 Memory Browser 窗口以更新值，而无需暂停代码，如节 5.3.1.1 中所述。
  - 将 *TINV\_idRef\_pu* 设置为 -0.013pu。
  - 若要启动 PFC 模式，请设置 *TINV\_startPowerStage* = 1。现在会以正弦信号（带有一些谐波，因为电流处于低功耗）从电网汲取电流，并会在 vBus 上看到升压操作。如果设置了过流跳闸，则通过将 *TINV\_clearPwmTrip* 设为 1 来清除 PWM 跳闸。
- 随着负载的增加，电流变为正弦信号。这验证了 PFC 会在 30V<sub>RMS</sub> 下启动。一旦 clearPWMTrip 成功清除浪涌过流跳闸，就会缓慢地将交流输入逐步增加到所需的值。避免因交流输入突然增加而导致 PWM 跳闸。
- 在无监控电压环路的情况下以较高的输入电压运行本实验会导致直流端子上出现较大的过压。如前所述，出于安全原因，始终应在低电压和低功率下启动本实验，然后缓慢斜升至所需电压以进行闭合电流环路调整。

对于软启动，死区的宽度可调节为 (30 $\times$ 5) ns，并且可逐步降至所需的电平。PWM 脉冲宽度随着死区宽度的增加而减小。图 5-14 展示了此设置中的 PWM 配置，其中死区设置为较大值并缓慢减小至标称值以限制电流尖峰。如果没有软启动，就会出现巨大的电流尖峰，导致过流跳闸，直流母线电压也会崩溃。

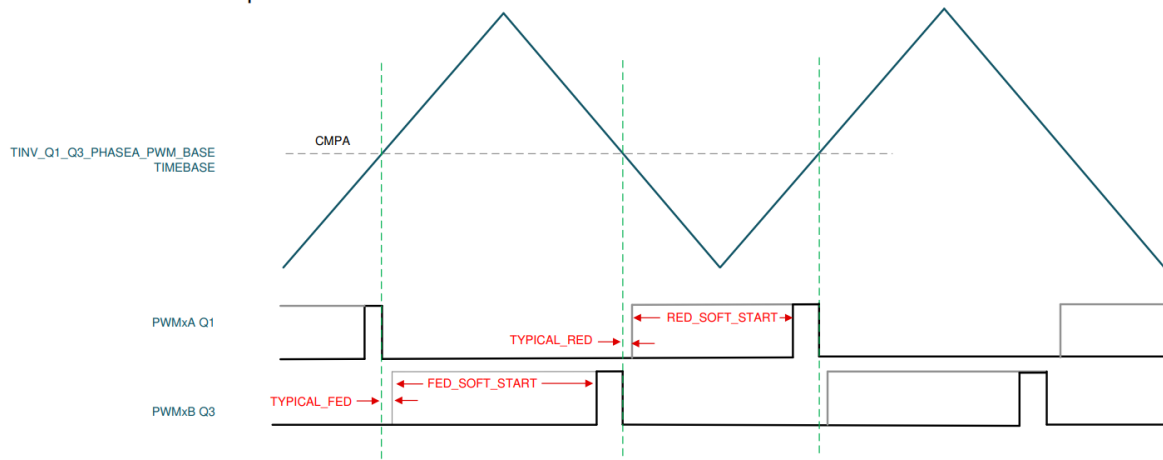


图 5-14. PWM 死区软启动

### 5.3.3 实验 7

实验 7 与实验 6 类似，只是实验 7 将直流母线电压环路闭合。可以测试此实验中的电流闭环和电压闭环 PFC 运行。图 5-15 解释了实验中 ISR1 的代码流程。

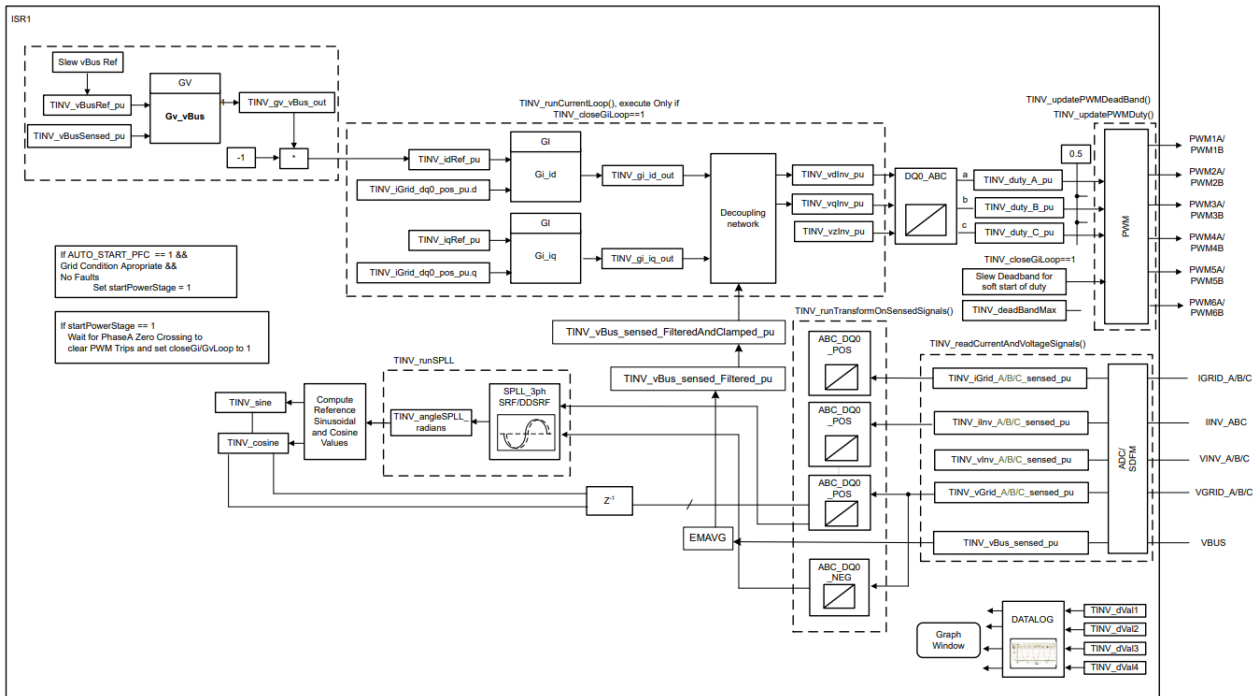


图 5-15. 实验 7 流程图

按照以下步骤操作来验证实验 7：

- 通过更改用户设置文件中的 **LAB\_NUMBER** 将工程设置为实验 7。在该条件下，转换器作为整流器运行。在此实验中，ISR1 包含电流闭环和电压闭环 PFC 运行。
- 此实验的环路系数  $K_p$  和  $K_i$  与之前的实验类似。
- 可连接至电网测试此实验，这是因为电压环路闭合并且  $TINV\_idRef\_pu$  值针对输入电压进行了调整。
- 构建代码并将其加载到 AM263x controlCARD 中。
- 在测试开始时将交流输入设置为较低的  $30V_{RMS}$  值，并在清除 EPWM 跳闸后逐步将输入增加到  $230V_{RMS}$ 。
- 加载代码时，继电器默认为打开状态。打开交流电源。由于浪涌电流，PWM 进入跳闸状态。

- 下面描述了在 *Memory Browser* 或 *Watch* 窗口中设置变量的顺序。设置 *Memory Browser* 窗口以更新值，而无需暂停代码，如节 5.3.1.1 中所述。
  - 要启动 PFC 模式，请为 *TINV\_startPowerStage* 变量输入“1”。现在会以正弦信号（带有一些谐波，因为处于低功率）从电网汲取电流，并会在 vBus 上看到升压操作。如果设置了过流跳闸，则通过将 *TINV\_clearPwmTrip* 设为 1 来清除 PWM 跳闸。
  - 随着负载的增加，电流变为正弦信号。这验证了 PFC 会在  $30V_{RMS}$  下启动。一旦 *clearPWMTrip* 成功清除浪涌过流跳闸，就会缓慢地将交流输入逐步增加到所需的值。
  - 将 *TINV\_vBusRef\_pu* 增加到 0.684，对应于 800V 的总线电压。
  - 确认系统稳定后，尝试在启动时将 *TINV\_vBusRef\_pu* 设置为 0.684pu。

### 5.3.4 实验 7 中的 PFC 运行测试结果

本节中以表格形式列出了实验 7 获得的 PFC 运行结果，其中包含各种总线基准值和输入交流电压。交流正弦波输入指定给 PFC TIDA-1606 设置的三个阶段。ADC 应该能检测这些电压并产生 PWM，这些 PWM 在经过滤波后将根据输入提供正弦输出。这证明 SPLL 角度锁定与输入交流电压同步工作。图 5-16 显示了向 PFC 硬件提供 200V 交流输入时三相电网侧的电流波形输出。



图 5-16. PFC 模式下 TIDA 硬件电网侧 200V 交流输入的  $I_a$ 、 $I_b$ 、 $I_c$  波形

向 TIDA-01606 输入端子施加此三相输入电压，以产生稳压直流母线电压，同时保持正弦输入电流。这证明了 PFC 在 TIDA-1606 设置中与 AM263x 配合使用时的运行情况。在给定的总线基准值下，当输入交流电压发生变化时，功率分析仪工具测得的输出直流电压保持恒定且稳定，从而证明电压环路正常工作。这验证了闭合电压环路运行。图 5-17 在 200V 输入交流电源下捕获了此结果。该直流电压是使用双向直流电源和分析仪工具从 PFC 硬件总线侧捕获的。

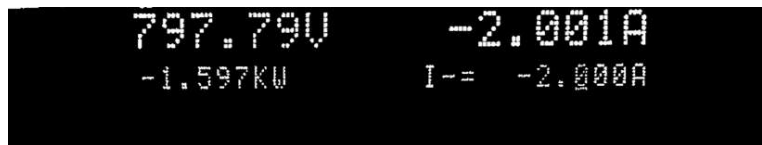


图 5-17. PFC 硬件在 200V 交流输入下的输出总线电压

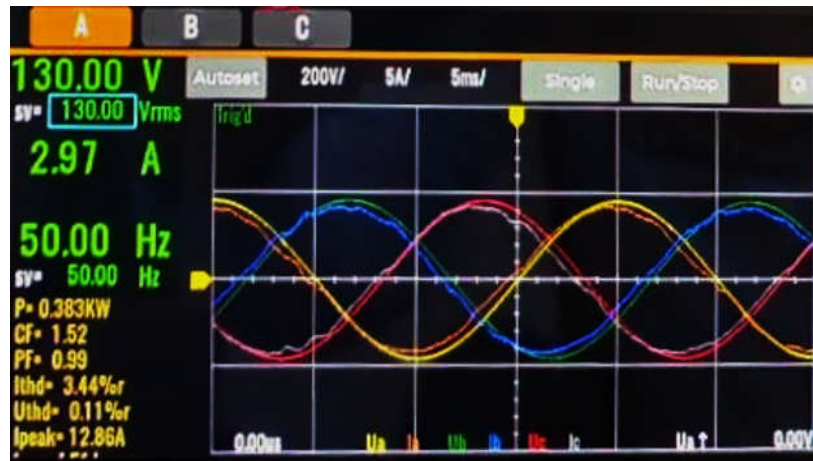


图 5-18. PFC 模式下 TIDA 硬件电网侧 130V 交流输入的  $I_a$ 、 $I_b$ 、 $I_c$  波形



图 5-19. PFC 硬件在 130V 交流输入下的输出总线电压

PFC 运行模式下输出直流电压与不同总线基准值和正弦交流输入的比较情况记录在表 5-3 中。实验 7 测试需要以下记录，为三相输入的 A 相捕获了交流输入。在输入交流电压为 200V 时，总线电压基准值为 0.7，在 PFC 级之后对应于 797.83V 输出直流电压。在此条件下测得的电流 THD (总谐波失真) 为 2.47%。该装置尚未针对 230V<sub>RMS</sub> 的标称交流电压进行测试。

表 5-3. 实验 7 与 AM263x controlCARD™ 配合使用 PFC 时的运行性能

vBus Ref	输入电压 AC A	输入电流 A	频率 A	输入功率 A	CF A	PF A	ITHD A	VTHD A	I <sub>peak</sub> A	测得的总线电压	总线侧电源	恒定电流设置测得的总线侧电流
	(V <sub>RMS</sub> )	(A)	(Hz)	(KW)			(%)	(%)		(V)	(KW)	(A)
0.7	200	2.79	50	0.547	1.51	0.98	2.47	0.13	9.73	797.83	1.596	2
0.6	170	2.75	50	0.46	1.54	0.98	2.64	0.13	12.86	672.83	1.345	2
0.5	130	2.97	50	0.383	1.52	0.99	3.44	0.11	12.86	560.24	1.12	2
0.4	80	3.89	50	0.311	1.49	1	3.73	0.18	12.86	447.84	0.896	2
0.3	70	3.32	50	0.232	1.48	1	5.38	0.22	12.86	335.22	0.67	2
0.3	50	3.96	50	0.198	1.45	1	5.32	0.31	12.86	280.83	0.562	2

对于给定的总线电压基准值，观察到在较高的交流输入值下，PF 和 ITHD 值变得更合适。在图 5-20 和图 5-21 中，所有三个相位的输入电压和其他特性都通过连接到硬件的功率分析工具进行捕获。



图 5-20. 200V 交流输入和恒定总线电压基准值下的 ITHD 和 PF 值



图 5-21. 130V 交流输入和恒定总线电压基准值下的 ITHD 和 PF 值

### 5.3.5 PFC 中断基准测试

本节包含 PFC 实验中 ISR1 的中断执行时间，并介绍运行分析测试的过程。

在构建和加载代码之前，为所需的 ISR 添加分析代码。使用 API `CycleCounterP_getCount32()` 在 ISR 开始时捕获周期计数器的值。使用同一 API 在 ISR 结束时捕获周期计数器的值。现在，结束计数器值和开始计数器值之间的差异给出了执行 ISR1 代码所需的确切 CPU 时钟周期数。此分析代码在应用程序中进行了注释，因为它仅用于对 ISR1 执行时间进行基准测试。要使用此代码，请在工程中定义 `PROFILING_ISR1_MAIN` 宏。

通过 CCS 构建代码更改并在 AM263x controlCARD 中加载代码。在“Watch”窗口中添加变量 `TINV_startPowerStage`、`TINV_closeGiLoop`、`Max_CPU_Cycles` 和 `Min_CPU_Cycles`。在 CCS 中运行应用程序，并设置 `TINV_startPowerStage = 1` 和 `TINV_closeGiLoop = 1` 的值，以启动电流和电压环路转换。暂停代码并查找 `Max_CPU_Cycles` 和 `Min_CPU_Cycles` 的值以查看所记录的值。要复位 CPU 周期值，请将 `Max_CPU_Cycles` 设置为 0，将 `Min_CPU_Cycles` 设置为任意最大值。

此基准测试中未捕获 ISR 中断延迟，即从硬件中断触发 ISR 到软件应用程序中开始执行 ISR1 所用的时间。

表 5-4 中的 CPU 周期值在“Max”列中较高，因为未缓存代码并且为首次执行设置了一些标志。表的“Min”列中列出了 Control ISR1 在首次执行后的执行。“Min”列中的值用于捕获理想情况下的 ISR1 执行时间。

表 5-4. PFC 运行的中断基准测试

实验编号	最小值		最大值	
	CPU 周期数	执行时间 (μs)	CPU 周期数	执行时间 (μs)
实验 5	599	1.4975	627	1.5675
实验 6	842	2.105	895	2.2375
实验 7	856	2.14	924	2.31

## 6 参考文献

- 德州仪器 (TI), [TIDA-01606 : 10kW 双向三相三级 \(T 型\) 逆变器和 PFC 参考设计指南](#)
- 德州仪器 (TI), [适用于三相并网应用并采用 C2000™ 微控制器的软件锁相环设计应用手册](#)

## 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 © 2023，德州仪器 (TI) 公司