

PMSM 上使用快速电流环路、基于 EtherCAT 的互联伺服驱动器



Ramesh T Ramamoorthy

摘要

本应用报告有助于评估 EtherCAT® 通信，并使用 TI 的 TMS320F28388D 实时控制器对互联伺服驱动器实现快速电流环路 (FCL) 的控制环路进行频率响应分析。该器件中的可配置逻辑块 (CLB) 可在没有外部逻辑电路或者 FPGA 的情况下，连接工业位置感应驱动器中常见的各类绝对串行编码器。本次评估使用的位置编码器为 QEP 和 T-format 类型。电流感测方法是使用 ADC 的霍尔传感器或使用 SDFM 的分流电阻器。使用的平台是 TI 的 DesignDRIVE IDDK EVM 套件和 C2000Ware MotorControl SDK。

频率响应分析是使用 TI 的 Software Frequency Response Analyzer (SFRA) 库在软件中执行的，该库在控制 MCU F28388D 上实时运行。除了控制电机外，MCU 还与 PC 上运行的 TwinCAT 进行 EtherCAT 通信。F2838x MCU 属于 C2000™ 微控制器系列，具有两个 C28x CPU 内核和一个 Arm® Cortex®-M4 内核，旨在使用一系列功能强大的外设进行有效的控制和通信，适用于数字电源和电机控制应用。在伺服驱动器领域，此控制器通过减少系统元器件数量和提高效率，为三相电机提供具有成本效益的智能、高带宽控制器设计。此控制器由于其实时特性、硬件加速器和完善的软件支持，可有效实现 FCL 算法，与使用 FPGA 的同类实现方案相比，在性能、成本和开发时间方面有过之而无不及。

本应用报告描述以下内容：

- 在 CPU、M4 和 CLA 内核上增量构建 FCL 函数模块
- 借助于使用 ADC 的 LEM 传感器进行内嵌式电流感测，或使用 SDFM 执行基于隔离式分流电阻器的电流测量
- 使用 QEP 或 T-Format 编码器的位置反馈
- 实验结果

内容

1 引言	4
1.1 本文中使用的首字母缩写词.....	4
2 在高带宽电流环路中使用 TMS320F2838x MCU 的优势	5
3 伺服驱动器中的电流环路	5
4 快速电流环路库概述	6
5 快速电流环路评估	8
5.1 评估设置.....	9
6 增量构建级别 1	13
6.1 SVGEN 测试.....	15
6.2 使用 DAC 测试 SVGEN.....	16
6.3 逆变器功能验证.....	16
7 增量构建级别 2	17
7.1 在软件中设置过流限制.....	17
7.2 电流感测方法.....	18
7.3 电压感测方法.....	18
7.4 设置电流调节器限制.....	18
7.5 验证电流感测.....	19
7.6 位置编码器反馈.....	19
8 增量构建级别 3	22
8.1 观察结果一 - PWM 更新延迟.....	23
9 增量构建级别 4	26
9.1 观察结果.....	28
10 增量构建级别 5	28
11 增量构建级别 6	29
11.1 集成 SFRA 库.....	29
11.2 启动 SFRA 前的初始设置.....	30
11.3 SFRA GUI.....	31
11.4 设置 GUI 以连接到目标平台.....	34
11.5 运行 SFRA GUI.....	36
11.6 电流反馈 SNR 的影响.....	38
11.7 推论.....	39
11.8 相位裕度与增益交叉频率间的关系.....	39
12 增量构建级别 7	40
12.1 在 CPU1 上运行代码以将 ECAT 分配给 CM.....	40
12.2 在 CM 上运行代码以设置 ECAT.....	40
12.3 设置 TwinCAT.....	41
12.4 通过 TwinCAT 扫描 EtherCAT 器件.....	42
12.5 针对 ESC 的 ControlCard EEPROM 编程.....	43
12.6 运行应用程序.....	44
13 增量构建级别 8	46
13.1 在 CPU1 上运行代码以将 ECAT 分配给 CM.....	46
13.2 在 CM 上运行代码以设置 ECAT.....	46
13.3 运行应用程序.....	47
14 参考文献	47
15 修订历史记录	48

插图清单

图 3-1. 针对交流电机的 FOC 基本系统配置.....	5
图 4-1. 反馈采样和 PWM 更新之间的延迟时间.....	6
图 4-2. 快速电流环路库方框图.....	7
图 5-1. 在伺服驱动器中使用 Δ - Σ 调制器进行电流感测.....	11
图 6-1. 1 级方框图.....	13
图 6-2. 构建级别 2 的表达式窗口.....	14
图 6-3. 电压角以及 SVGEN Ta、Tb 和 Tc.....	15
图 6-4. 显示 Ta 和 Tb 波形的 DAC 输出.....	16
图 7-1. 2 级方框图.....	17

图 7-2. 测量电流和感测电流的示波器图.....	19
图 7-3. 参考角和转子位置的示波器图.....	20
图 7-4. 表达式窗口.....	21
图 8-1. 3 级方框图，显示了最内层环路 - FCL.....	23
图 8-2. “Expressions”（表达式）窗口延迟快照.....	24
图 8-3. ADCSoC 和 FCL 完成事件的示波器图.....	25
图 9-1. 4 级方框图，显示了具有内层 FCL 的速度环路.....	26
图 9-2. 0.33pu 步进负载和 0.3pu 速度下的同步坐标系内定子电流的磁通分量和扭矩分量.....	27
图 10-1. 5 级方框图，显示了具有内层 FCL 的位置环路.....	29
图 10-2. 参考位置到伺服和反馈位置的示波器图.....	29
图 11-1. 6 级方框图.....	30
图 11-2. SFRA GUI.....	32
图 11-3. SFRA GUI MC.....	33
图 11-4. GUI 设置图.....	34
图 11-5. 连接到 C2000 MCU 的 SFRA GUI.....	35
图 11-6. 显示幅度和相角的电流环路 SFRA 开环波特图.....	36
图 11-7. 显示幅度和相角的电流环路 SFRA 闭环波特图.....	37
图 11-8. 电流环路的 SFRA 开环波特图 - 具有高 SNR 的电流反馈.....	38
图 11-9. 电流环路的 SFRA 闭环波特图 - 具有高 SNR 的电流反馈.....	38
图 11-10. 增益交叉频率与相位裕度（通过实验获得）间的关系图.....	39
图 12-1. TwinCAT 配置模式图标.....	41
图 12-2. TcSwitchRunTime 窗口已激活.....	42
图 12-3. TwinCAT 解决方案资源管理器.....	43
图 12-4. TwinCAT EEPROM 窗口.....	44
图 12-5. 使用配置模式图标重新启动 TwinCAT.....	44
图 12-6. EtherCAT 从站处于 OP 模式.....	45
图 12-7. TwinCAT 解决方案资源管理器输入和输出.....	45
图 12-8. F2838x CM EtherCAT Slave (SYSTEM) 输入/输出数据.....	46

表格清单

表 4-1. FCL 接口函数汇总.....	8
表 5-1. 每个增量系统构建中已验证的功能.....	12
表 5-2. 每个增量系统构建中使用的功能模块.....	12

商标

C2000™ is a trademark of Texas Instruments.

EtherCAT® is a registered trademark of Beckhoff Automation GmbH, Germany.

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

交流驱动器的 FOC 概念是众所周知的，并且在 TI 的许多早期文档中已有概述。根据不同应用，现代的交流伺服驱动器需要高带宽电流控制和速度控制以实现出色的性能，例如在数控机床中或在快速和精确控制应用中就是如此。由于这些系统有大量的实时计算和灵活的 PWM 需求，许多设计人员选用了 FPGA、外部快速 ADC 和多个 MCU 的组合。

借助 TMS320F2838x MCU 及其更高的集成度，可以实现快速电流环路 (FCL) 算法，与传统外部硬件电路的 FOC 方法一样，C2000 器件上的 FCL 算法同样可以实现高电流环路带宽。TI 已在此 MCU 上开发了 FCL 算法，并在 Design DRIVE IDDK 平台上实现了该算法。

对于 10kHz PWM 载波，电流环路增益交叉频率预计超过 3kHz，闭环带宽预计约为 5 KHz (根据 NEMA ICS 16 和中国 GBT 16439-2009 指导原则)，最大占空比为预计约为 96%。使用 TI 的软件频率响应分析器 (SFRA) 库，可以实时执行电流环路的频率响应分析，以验证上述基准测试。电机驱动系统实时动态频率响应分析是 MCU 供应商特有的，目前仅在 C2000 MCU 上受支持。

由于存在可配置逻辑块 (CLB)，现在可为采用不同协议 (例如 EnDAT、BiSS、T-format 等) 的各种绝对编码器实现各种定制接口逻辑，而无需外部逻辑或 FPGA。

除控制之外，TMS320F2838x MCU 还具有基于 Arm Cortex-M4 的连接管理器和 EtherCAT 从站控制外设。这有助于在单个芯片中无缝集成控制和通信，从而为工业伺服驱动器开发具有成本效益的解决方案。

本文档评估了 FCL 算法在 C2000 器件上的实现，分析了电流环路的实时频率响应，使用片上可配置逻辑块 (CLB) 验证了 T-format 编码器的接口逻辑，还与用作互联驱动器的主站进行 EtherCAT 通信。位置闭环控制可以使用 QEP 编码器或者 T-Format 编码器，并且这两种情况都可以用 FCL 算法实现。文中讨论了频率响应分析的定量测试结果。

1.1 本文档中使用的首字母缩写词

- ACIM - 交流感应电机
- ADC - 模数转换器
- CLA - 控制律加速器 (位于 C2000 MCU)
- CLB - 可配置逻辑块 (位于 C2000 MCU)
- CMPSS - 比较器子系统外设 (位于 C2000 MCU)
- CNC - 计算机数控
- DMC - 数字电机控制
- eCAP - 增强型捕捉模块
- ECAT - EtherCAT
- ePWM - 增强型脉宽调制器
- eQEP - 增强型正交编码器脉冲模块
- FCL - 快速电流环路
- FOC - 场定向控制
- FPGA - 现场可编程门阵列
- IDDK - 工业驱动开发套件 (来自 TI)
- HVDMC - 高压 DMC
- MCU - 微控制器单元
- PMSM - 永磁同步电机
- PWM - 脉宽调制
- SDFM - Σ - Δ 滤波器模块
- TMU - 三角数学单元 (位于 C2000 MCU)

2 在高带宽电流环路中使用 TMS320F2838x MCU 的优势

C2000 MCU 系列器件具有执行复杂控制算法所需的计算能力，并提供了正确的外设组合以连接 DMC 硬件的各个组件（例如 ADC、ePWM、QEP 和 eCAP）。这些外设具有所有必要的机制来提供灵活的 PWM 保护，例如 PWM 的 Trip zones 保护模块及内部的模拟比较器模块。

F2838x MCU 的 C28x 内核包含如下其他硬件特性：

- 更高的 CPU 和 CLA 时钟频率
- 四个高速 12 位和 16 位 ADC
- 三角数学单元 (TMU)
- 并行处理块，例如 CLA
- 可配置逻辑块 (CLB)
- Σ - Δ 滤波器模块 (SDFM)

这些特性共同提供了足够的硬件支持，与其前代产品相比增加了计算带宽，并提供了出色的实时控制性能。CLB 有助于灵活地连接使用各类串行通信协议的各种绝对串行编码器，从而使 FPGA 在典型的高带宽伺服控制应用中变得多余。此外，C2000 软件（库和应用软件）和硬件 (TMDXIDDK379D) 生态系统有助于减少用户开发高端数字电机控制解决方案所需的时间和精力。

3 伺服驱动器中的电流环路

图 3-1 展示了 FOC 伺服驱动器中使用的基本电流环路。

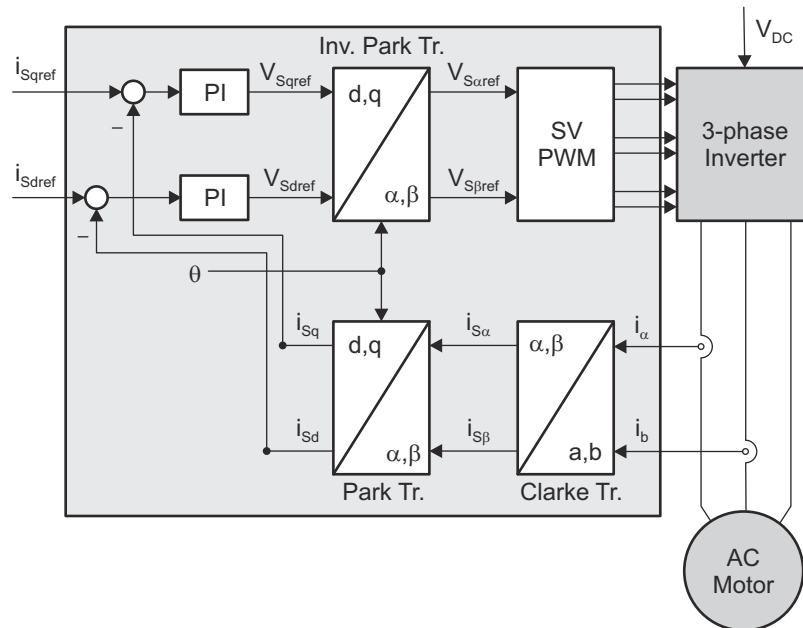


图 3-1. 针对交流电机的 FOC 基本系统配置

测量了两个电机相电流。这些测量值馈入 Clarke 变换模块。此设计的输出为 $i_{s\alpha}$ 和 $i_{s\beta}$ 。这两个电流分量以及转子磁通位置是 Park 变换的输入，此变换可以将它们转换为 d 和 q 旋转坐标系中的电流 (i_{sd} 和 i_{sq})。 i_{sd} 和 i_{sq} 分量与基准 i_{sdref} （磁通基准）和 i_{sqref} （扭矩基准）相比较。此时，控制结构具有一个有意思的现象：通过简单地改变磁通基准并获得转子磁通位置，可对同步或异步电机进行控制。在同步永磁电机中，转子磁通是固定的，并由永磁体确定，因此无需产生转子磁通。所以，在控制 PMSM 电机时，除了弱磁期间外，可以将 i_{sdref} 设置为零。

由于 ACIM 电机的运转需要生成一个转子磁通，磁通基准一定不能为零。这很方便地解决了经典控制结构的一个主要缺陷：异步至同步驱动的可移植性。扭矩命令 i_{sqref} 可以连接到速度调节器的输出端。电流调节器的输出为 V_{sdref} 和 V_{sqref} 。这些输出会应用于逆向 Park 变换。利用转子磁通的位置，此设计生成 $V_{s\alpha ref}$ 和 $V_{s\beta ref}$ ，它们是固定正交坐标系中定子电压的分量。这些分量作为空间矢量 SVPWM 的输入，SVPWM 的输出是驱动逆变器的信号。

备注

Park 和 Park 逆变换均需要转子磁通位置。这个转子磁通位置的获得方式由交流电机类型（同步还是异步）而定。

4 快速电流环路库概述

数字电机控制系统存在的主要问题是，内部控制环路的传输滞后降低了系统的响应，从而影响系统在更高频率和运行速度下的性能。解决此问题将有助于改善电流环路带宽。但是，若要在不损失直流母线电压利用率的情况下实现此目标，以下各项是必需的：

- 高计算能力
- 正确的控制外设组合
- 出色的控制算法

TMS320F2838x 提供必要的硬件支持来实现更高的性能，而在此实时控制器上运行的 TI FCL 库则提供了必要的算法支持。MC SDK v2.01.00.00 及后续版本可提供源代码。

为了提高电流环路的速度并优化电流环路的工作范围，反馈采样和 PWM 更新之间的延迟必须尽可能小，如图 4-1 所示。实质上，在所示 PWM 载波周期内，在载波峰值处会发生系统采样，并在 $T_{\text{PWM_update}}$ 时间点更新由控制器输出的 PWM 占空比。这样可以大幅减少传输滞后并改善带宽。但是，缺点是会失去用于计算逆变器 PWM 占空比新值的有效 PWM 时间窗口。此时间窗口被称为消隐窗口。此消隐窗口越小，直流母线电压利用率就越大，电机运行速度范围也越大。

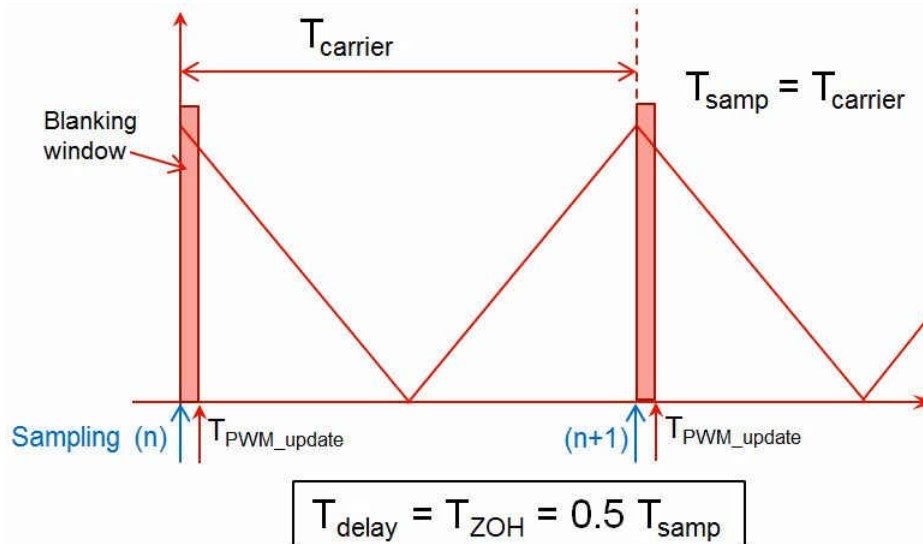


图 4-1. 反馈采样和 PWM 更新之间的延迟时间

通常，在许多载波频率为 10kHz 的应用中，不超过 2 μ s 的延迟是可以接受的。过去，此任务是将高端 FPGA、外部 ADC 和 MCU 结合使用来实现的。但是，C2000 具有相应的硬件支持功能（例如三角数学单元 (TMU)、CLA、高速 ADC 或 SDFM、单周期 ADC 读取和 PWM 写入），因此无需 FPGA 或外部 ADC 即可在此 MCU 上实现这种控制。

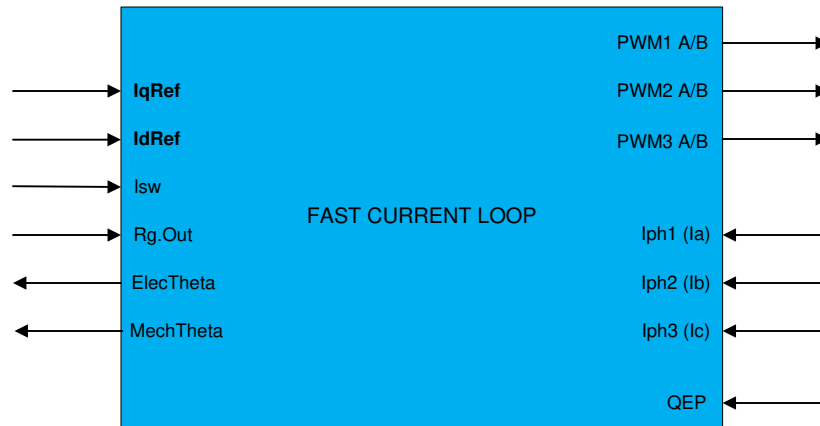


图 4-2. 快速电流环路库方框图

FCL 库利用 F2838x MCU 中的以下特性：

- TMU
- 四个高速 12 位或 16 位 ADC，或 Σ - Δ 滤波器模块 (SDFM)
- 其他处理块，例如 CLA

图 4-2 展示了 FCL 库及其输入和输出的方框图。FCL 库在 CPU、CLA 和 TMU 之间对算法进行了分区，旨在将延迟降低到 1.0 μ s 以内，小于可接受的 2.0 μ s。如果算法是以汇编语言编写的，则可以进一步优化。

FCL 库支持两种类型的电流调节器，即标准 PI 控制器和复杂控制器。复杂控制器能够以更快的速度提供比标准 PI 控制器更高的带宽。两种电流调节器均用于用户评估。在示例工程中，可以通过适当设置 FCL_CNTLRLR 宏并研究电流调节器的比较结果来选择电流调节器。

表 4-1 列出了 FCL API 函数及其说明。

表 4-1. FCL 接口函数汇总

API 函数	说明
<code>uint32_t FCL_getSwVersion(void)</code>	此函数返回 32 位常量，对于此版本，返回的值为 0x00000007。
<code>void FCL_runComplexCtrl(void)</code>	此函数在具有 QEP 编码器和霍尔传感器（使用 ADC）的 FCL 中执行复杂控制
<code>FCL_runSDFMComplexCtrl(void)</code>	此函数在具有 QEP 编码器和分流电阻器（使用 SDFM）的 FCL 中执行复杂控制
<code>void FCL_runAbsEncComplexCtrl(void)</code>	此函数在具有典型绝对编码器和霍尔传感器（使用 ADC）的 FCL 中执行复杂控制
<code>void FCL_runSDFMAbsEncComplexCtrl(void)</code>	此函数在具有典型绝对编码器和分流电阻器（使用 SDFM）的 FCL 中执行复杂控制
<code>void FCL_runPICtrl(void)</code>	此函数在具有 QEP 编码器和霍尔传感器（使用 ADC）的 FCL 中执行 PI 控制
<code>void FCL_runSDFMPICtrl(void)</code>	此函数在具有 QEP 编码器和分流电阻器（使用 SDFM）的 FCL 中执行 PI 控制
<code>void FCL_runAbsEncPICtrl(void)</code>	此函数在具有典型绝对编码器和霍尔传感器（使用 ADC）的 FCL 中执行 PI 控制
<code>void FCL_runSDFMAbsEncPICtrl(void)</code>	此函数在具有典型绝对编码器和分流电阻器（使用 SDFM）的 FCL 中执行 PI 控制
<code>void FCL_runPICtrlWrap(void)</code>	当使用 QEP 作为位置传感器时，在 PI 控制模式下，在退出 ISR 之前，用户应用程序在 FCL 结束时将调用此函数
<code>void FCL_runAbsEncPICtrlWrap(void)</code>	当使用绝对编码器作为位置传感器时，在 PI 控制模式下，在退出 ISR 之前，用户应用程序在 FCL 结束时将调用此函数
<code>void FCL_runQEPWrap(void)</code>	此函数由用户应用程序调用以结束 QEP 反馈过程。此函数仅在 FCL_LEVE2 中使用
<code>void FCL_runComplexCtrlWrap(void)</code>	当使用 QEP 作为位置传感器时，在复杂控制器模式下，在退出 ISR 之前，用户应用程序在 FCL 结束时将调用此函数
<code>void FCL_runAbsEncComplexCtrlWrap(void)</code>	当使用绝对编码器作为位置传感器时，在复杂控制模式下，在退出 ISR 之前，用户应用程序在 FCL 结束时将调用此函数
<code>void FCL_initPWM(uint32_t basePhaseU, uint32_t basePhaseV, uint32_t basePhaseW)</code>	此函数用于初始化 PWM 以支持 FCL 运行，在初始化或设置过程中，用户应用程序将调用此函数
<code>void FCL_resetController(void)</code>	调用此函数可以重置 FCL 变量，此函数在您需要停止电机并重新启动电机时很有用
<code>void FCL_initQEP(uint32_t baseA)</code>	此函数对 eQEP 外设进行初始化以便连接到 QEP
<code>void FCL_initADC(uint32_t resultBaseA, ADC_PPBNumbert baseA_PPB, uint32_t resultBaseB, ADC_PPBNumbert baseB_PPB, uint32_t adcBasePhaseW)</code>	此函数对用于感测电机相电流的 ADC 进行初始化

如需了解该库的更多信息，请参阅 [快速电流环路 MotorControl SDK 库用户指南](#)，此文档位于：

`\\ti\c2000\C2000Ware_MotorControl_SDK_2_01_00_00\libraries\fcl\docs。`

FCL 的源代码位于：

`\\ti\c2000\C2000Ware_MotorControl_SDK_2_01_00_00\libraries\fcl\source`

备注

该库是以模块化格式编写的，如果满足以下条件，则可以移植到使用 F2838x 器件的用户平台：

- 电机相电流反馈被读入该库内部变量。但是，可提供 D 轴和 Q 轴电流反馈。
- 用于控制电机 A 相、B 相和 C 相的 PWM 模块链接到该库。
- 连接到 QEP 传感器的 QEP 模块链接到该库。
- 该库使用 CLA 任务 1-4。这必须包含在用户应用程序中。
- 编写了独立但相似的控制函数以用于高延迟 T-format 位置编码器
- 编写了类似的控制函数来进行 SDFM 电流感测

5 快速电流环路评估

TI 提供了 FCL 算法软件库及其源代码（从 MC SDK v2.01.00.00 起），用于在 TI 的 DesignDRIVE IDDK 平台上使用 TMS320F2838x、TMS320F2837x 或 TMS320F28004x MCU 进行评估。本节通过一个示例工程介绍了逐步评估 FCL 软件库以控制永磁同步电机的方法。

示例工程具有：

- PMSM 电机的有传感器 FOC
- FCL
- 位置、速度和扭矩控制环路
- 位置传感器支持：
 - 增量编码器 (QEP)
 - 绝对编码器 - T-Format
- 电流感测：
 - 使用 ADC 的模拟反馈 (来自 LEM 传感器 (磁通门/霍尔传感器))
 - 使用 SDFM 的分流电阻器

5.1 评估设置

5.1.1 硬件

示例工程在 TI 的 [Design DRIVE 开发套件 IDDK - TMDXIDDK379D](#) 上进行了评估。该套件的 [DesignDRIVE IDDK 硬件参考指南](#) 和 [DesignDRIVE IDDK 用户指南](#) 可以在以下位置找到：

C:\ti\c2000\C2000Ware_MotorControl_SDK_2_01_00_00\solutions\tmdxiddk379d\docs\.

[DesignDRIVE IDDK 硬件参考指南](#) 概述了套件中的各种硬件功能块，并讨论了套件支持的各种接地配置以及与套件一同使用所必需的安全措施。在使用套件之前，了解硬件和安全方面的知识很重要。

[DesignDRIVE IDDK 用户指南](#) 帮助用户设置硬件以及开始使用软件、连接到目标平台并通过 Code Composer Studio 使用调试环境。

5.1.2 软件

安装 MCSDK 软件包后，可以在 C:\ti\c2000\C2000Ware_MotorControl_SDK_2_01_00_00\libraries\fcl\ 中找到 FCL 软件。

FCL 示例工程可在以下位置找到：

C:\ti\c2000\C2000Ware_MotorControl_SDK_2_01_00_00\solutions\tmdxiddk379d\2838x\ccs\sensored_foc

软件基于各种构建级别构建，旨在逐步集成 FCL 库以便在速度模式和位置模式下运行电机。

5.1.3 具有 T-Format 类型位置编码器的 FCL

5.1.3.1 将 T-Format 编码器连接到 IDDK

T-format 编码器连接到 IDDK 上的接头 H6，而 IDDK 连接到 M12 模块，请参阅 [DesignDRIVE IDDK 硬件参考指南](#)。该模块支持与各种串行编码器协议 (例如 EnDAT、BiSS、T-format 等) 的连接。确保将 H6 的引脚排列细节与编码器信号相匹配，并填充 IDDK 上的跳线 [Main]-J6、[Main]-J7 和 [Main]-J8 (在 M9 模块前面)。此编码器类型不需要 H6 的 CLK+ 和 CLK- 信号。仅需要 D+ 和 D-。

备注

在 IDDK 上的串行编码器接口 M12 模块中，有一个负载开关 U4，可根据需要向编码器提供 5V 电源。由于某些编码器的高容性负载，5V 电源可能会因干扰出现故障并使 MCU 复位。

如果发生这种情况，有两种方法可以解决此问题：

- 将电容器 C5 更换为 10nF (原为 330pF)
- 通过直接在此模块上连接 '5V0' 和 '5V0-Enc' 测试点来绕过开关 U4。这样，只要 IDDK 通电，编码器就会通电。

5.1.3.2 T-Format 接口软件

TI 提供了一个软件库及其源代码以连接到 T-format 编码器，该代码库基于器件中的可配置逻辑块 (CLB)。网上提供了 [CLB 工具](#) 用户指南、[使用 C2000 可配置逻辑块进行设计](#) 和 [如何将自定义逻辑从 FPGA/CPLD 迁移到 C2000 微控制器](#) 等应用报告。也可以在 C2000ware (版本 2_00_00_03 或更高版本) 中的以下位置找到这些资料：

`c:\ti\c2000\C2000Ware_<version>\utilities\clb_tool\clb_syscfg\doc`

在 T-format 接口的实现中，主要通过集成以下元件来实现通信：

- CPU
- 可配置逻辑块 (CLB)
- 串行外设接口 (SPI)
- 器件互连 (XBAR)

SPI 执行编码器数据发送和接收功能，而 CLB 控制时钟生成。CLB 模块内部实现了以下函数。请注意，只能通过 PM T-format 库中提供的库函数来访问 CLB 模块，用户无法通过其他方法对其进行配置。

- 能够生成时钟以进入芯片上的串行外设接口并循环回到 SPICLK 输入端
- 确定发送到编码器的时钟沿与接收到的数据之间的临界延迟
- 能够调整时钟延迟
- 通过 SPI 从输入/主输出 (SPISIMO) 引脚，监视来自编码器的数据并轮询启动脉冲
- 能够根据接口需求以特定间隔测量传播延迟
- 能够通过软件配置块并调整延迟

通过 SPI 接收位置信息，并通过 SPI receive ISR `spiRxFIFOISR()` 使用 `readTformatEncPosition()` 来读取位置信息。

`c:\ti\c2000\C2000Ware_MotorControl_SDK_2_01_00_00\libraries\position_sensing\tformat\Docs` 中提供了关于在 LaunchPad 平台实现 T-format 的详细信息

与 IDDK 相比，该 LaunchPad 对 T-format 编码器接口使用不同的 GPIO。但是，核心实现是相同的。IDDK 平台上用于交叉参考的 GPIO 列表如下所列：

```
#define ENCODER_SPI_BASE SPI_BASE
#define ENC_CLK_PWM_PIN 7
#define ENC_SPI_SIMO_PIN 24
#define ENC_SPI_SOMI_PIN 25
#define ENC_SPI_CLK_PIN 26
#define ENC_SPI_STE_PIN 27
#define ENC_TXEN_PIN 34
#define ENC_PWREN_PIN 32
```

5.1.3.3 T-Format 编码器延迟注意事项

实验室测试中使用的 T-format 编码器是 TS 5700 N 8501。这是一种延迟近 50 微秒的 17 位编码器。因此，不可能以 20kHz 的系统采样频率执行闭环控制。这意味着 20kHz PWM 频率的 SINGLE_SAMPLING 和 10kHz PWM 频率的 DOUBLE_SAMPLING 均不能实现。

5.1.4 SDFM

在要求高性能扭矩和运动控制的汽车牵引和伺服驱动器应用中，准确的高分辨率隔离式相电流测量至关重要。可用于相电流测量的选项包括使用霍尔效应传感器、磁通门传感器、电流互感器和分流电阻器。前三个选项具有内在的电隔离优势和高电流测量范围，但与分流电阻器选项相比，其线性度、带宽和漂移的性能较低。分流电阻器提供了一种高线性、高带宽且具有成本效益的测量解决方案。增强型隔离可同时提供隔离以及高线性和高带宽的优势。使用隔离式放大器或隔离式 Δ - Σ 调制器可实现基于分流的隔离式电流测量。

图 5-1 显示了伺服驱动电机中的相位电流感测机制，在设计中实施了两条路径，来自隔离调制器的高速位流输出由 TI 的 C2000 系列实时控制器进行滤波处理，此类控制器具有内置的 Σ - Δ 滤波器模块 (SDFM)，使用户能够微调信号带宽和准确度。SDFM 是一种四通道数字滤波器，专为电机控制应用中的电流测量而设计。每个通道都可以接收一个独立的 Δ - Σ 调制器位流，该位流由四个单独可编程的数字抽取滤波器进行处理。这些滤波器包括一个快速比较器，用于过流和欠流监测的即时数字阈值比较。此外，滤波器旁路模式可用于启用数据日志记录、分析和定制滤波。SDFM 引脚使用 GPIO 多路复用器进行配置。SDFM 的一个主要优点是，它能够实现简单、具有成本效益且安全的高电压隔离边界。

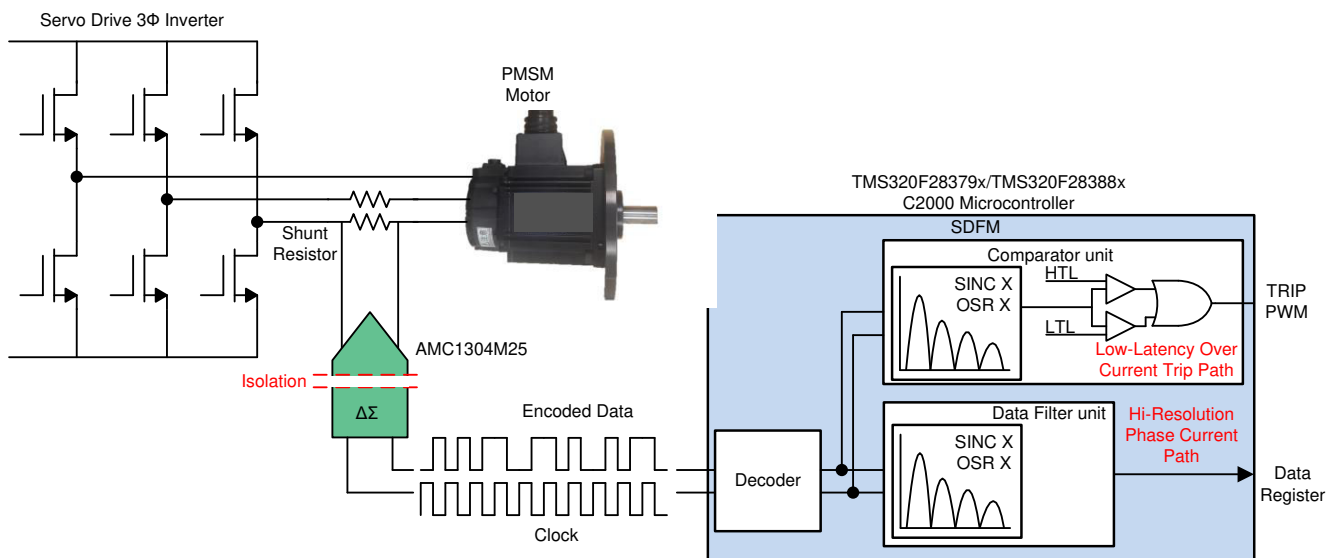


图 5-1. 在伺服驱动器中使用 Δ - Σ 调制器进行电流感测

在 IDDK 电路板上，分流电阻器上的电压将馈入 AMC1304M25 Σ - Δ 调制器，而该调制器可生成由 C2000 MCU 上的 SDFM 解调器进行解码的 Σ - Δ 流。调制器的时钟由 C2000 MCU 上的 EPWM5 模块生成，并且使用内置的 SDFM 调制器来决定 AMC1304M25 数据。

通过在 `fcl_f2838x_tmdxiddk_settings_cpu1.h` 中设置此配置来选择使用 SDFM 进行基于分流电阻器的电流感测，如下所示。

```
#define CURRENT_SENSE          SD_CURRENT_SENSE
```

5.1.5 增量系统构建

系统是通过各种构建级别逐步构建的，这一过程会验证每个级别的特定功能，以便在最终级别对系统进行全面验证并确保在各个方面都是完善的。增量系统构建的八个级别旨在验证系统中使用的各种模块。1 至 5 级逐渐构建一个启用 FCL 的位置控制伺服驱动器。第 6 级有助于使用 SFRA 进行频率响应分析。7 至 8 级逐渐构建一个用于执行速度控制或位置控制的互联伺服驱动器。

表 5-1 和表 5-2 总结了在每个构建级别集成和测试的核心功能，以及用于执行每个增量构建任务的内核。

表 5-1. 每个增量系统构建中已验证的功能

构建级别	所用内核		功能集成/验证
	CPU1	CM	
1 级	✓		基本 PWM 生成
2 级	✓		电机开环控制/反馈校准
3 级	✓		电流模式 - 使用 FCL 库来闭合电流环路
4 级	✓		速度模式 - 使用已在 3 级进行验证的内部 FCL 来闭合速度环路
5 级	✓		位置模式 - 使用已在 4 级进行验证的内部速度环路来闭合位置环路
6 级	✓		SFRA 分析 - 在速度模式 (4 级) 中对运行电流环路的电机执行 SFRA
7 级	✓	✓	ECAT 循环回到 TwinCAT - 不涉及电机控制
8 级	✓	✓	使用 TwinCAT 在速度模式 (4 级) 或位置模式 (5 级) 下控制电机的完全互联伺服驱动器

表 5-2. 每个增量系统构建中使用的功能模块

软件模块	1 级	2 级	3 级	4 级	5 级	6 级	7 级	8 级
PWM 生成	✓ ✓	✓	FCL 库	FCL 库	FCL 库	FCL 库		FCL 库
CLA 中的 QEP 接口/T-format 接口		✓ ✓	✓	✓	✓	✓		✓
FOC 函数			✓ ✓	✓	✓	✓		✓
SFRA 函数						✓ ✓		
ECAT 函数							✓ ✓	✓

6 增量构建级别 1

图 6-1 显示了构建级别 1 中所构建系统的方框图。在此步骤中，请保持电机断开连接。

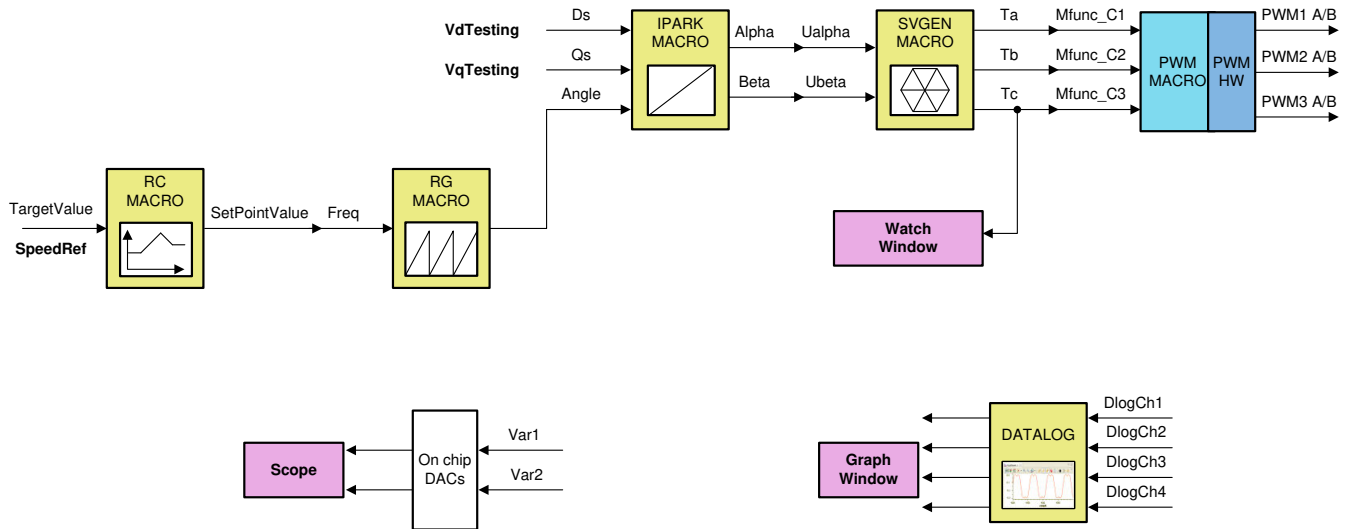
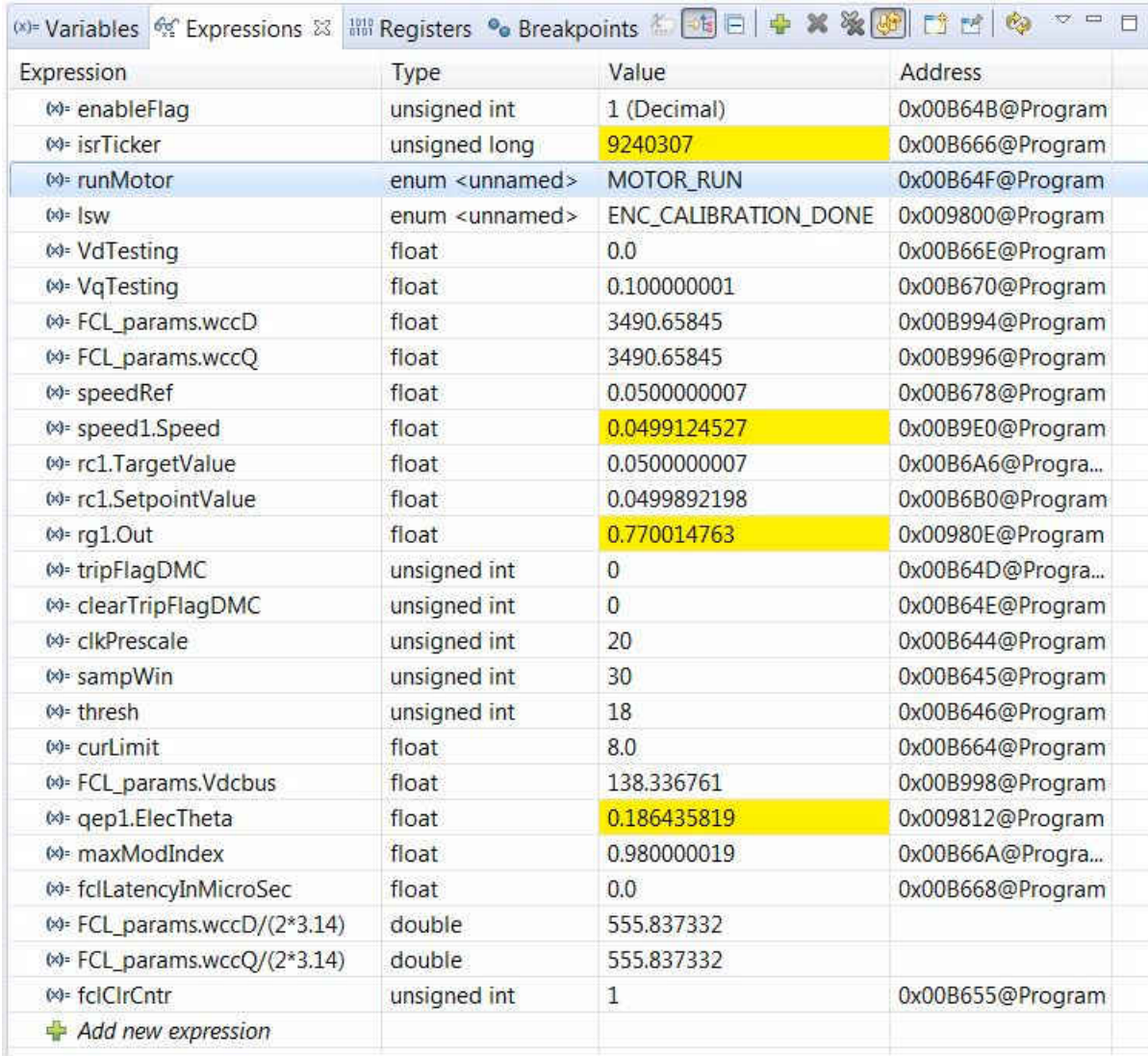


图 6-1. 1 级方框图

假定 *DesignDRIVE IDDK 用户指南* 中描述的负载和构建步骤已成功完成，本节将介绍“超简短”系统检出步骤，这些步骤用于确认系统中断、外设和独立于目标的逆向 Park 变换和空间矢量发生器模块，以及 PWM 初始化和更新模块的运行情况。

1. 打开 `fcl_f2838x_tmdxiddk_settings_cpu1.h` 并通过将 BUILDLEVEL 设置为 FCL_LEVEL1 (#define BUILDLEVEL FCL_LEVEL1) 来选择 1 级增量构建选项。
2. 右键单击工程名称，然后单击“Rebuild Project”（重建工程）。
3. 编译完成时，单击 Debug（调试）按钮，重置 CPU，重新启动，启用实时模式，然后运行。
4. 通过在“Expressions”（表达式）窗口中右键单击并从调试目录中导入 `fcl_f2838x_tmdxiddk_vars_cpu1.txt` 文件，将变量添加到“Expressions”（表达式）窗口。图 6-2 显示了从此文件导入到“Expressions”（表达式）窗口中的变量。暂时忽略针对变量显示的值。



Expression	Type	Value	Address
enableFlag	unsigned int	1 (Decimal)	0x00B64B@Program
isrTicker	unsigned long	9240307	0x00B666@Program
runMotor	enum <unnamed>	MOTOR_RUN	0x00B64F@Program
lsw	enum <unnamed>	ENC_CALIBRATION_DONE	0x009800@Program
VdTesting	float	0.0	0x00B66E@Program
VqTesting	float	0.100000001	0x00B670@Program
FCL_params.wccD	float	3490.65845	0x00B994@Program
FCL_params.wccQ	float	3490.65845	0x00B996@Program
speedRef	float	0.0500000007	0x00B678@Program
speed1.Speed	float	0.0499124527	0x00B9E0@Program
rc1.TargetValue	float	0.0500000007	0x00B6A6@Progra...
rc1.SetpointValue	float	0.0499892198	0x00B6B0@Program
rg1.Out	float	0.770014763	0x00980E@Program
tripFlagDMC	unsigned int	0	0x00B64D@Progra...
clearTripFlagDMC	unsigned int	0	0x00B64E@Program
clkPrescale	unsigned int	20	0x00B644@Program
sampWin	unsigned int	30	0x00B645@Program
thresh	unsigned int	18	0x00B646@Program
curLimit	float	8.0	0x00B664@Program
FCL_params.Vdcbus	float	138.336761	0x00B998@Program
qep1.ElecTheta	float	0.186435819	0x009812@Program
maxModIndex	float	0.980000019	0x00B66A@Progra...
fclLatencyInMicroSec	float	0.0	0x00B668@Program
FCL_params.wccD/(2*3.14)	double	555.837332	
FCL_params.wccQ/(2*3.14)	double	555.837332	
fclClrCntr	unsigned int	1	0x00B655@Program
+ Add new expression			

图 6-2. 构建级别 2 的表达式窗口

5. 在“Watch”（观察）窗口中将 *enableFlag* 设置为 1。在“Watch”（观察）窗口中看到 *isrTicker* 变量递增，可确认中断正常运转。

在软件中，被调整的关键变量为：

- speedRef*：用于更改转子速度（标么值）。
- VdTesting*：用于更改 d 轴电压（标么值）。
- VqTesting*：用于更改 q 轴电压（标么值）。

6.1 SVGEN 测试

为使速度命令加快，此测试将 *speedRef* 值馈送到斜坡控制模块中。斜坡模块的输出被馈送到斜坡发生器中，从而产生可生成正弦波的角度。该角度以及变量 *VdTesting* 和 *VqTesting* 馈送到逆向 Park 变换模块中，然后再馈送到空间矢量调制模块中以生成三相 PWM。

通过点击 Tools --> Graph --> Dual Time，可以在调试环境中使用图形工具查看空间矢量生成模块的输出。然后，在图形窗口中点击 Import (导入)，然后找到并选择：

```
\\solutions\tmdxiddk379d\2838x\debug\fc1_f2838x_tmdxiddk_graph1.graphProp
```

这样将绘制两个图，分别表示 *dlogCh1* 和 *dlogCh2* 指向的变量。同样，可以通过选择以下工程来打开另一个图：

```
solutions\tmdxiddk379d\2838x\debug\fc1_f2838x_tmdxiddk_graph2.graphProp
```

这样将绘制两个图，分别表示 *dlogCh3* 和 *dlogCh4* 指向的变量。

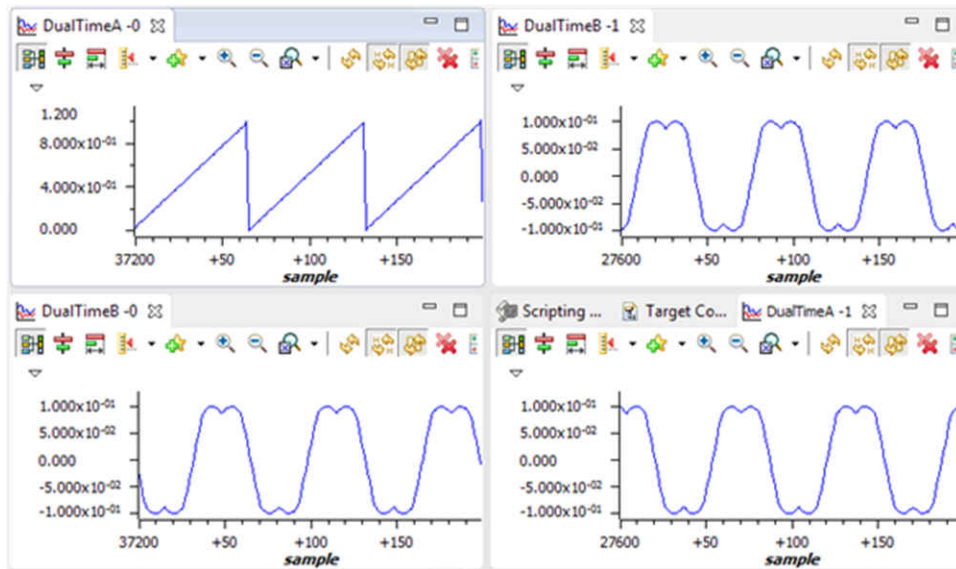


图 6-3. 电压角以及 SVGEN Ta、Tb 和 Tc

如图 6-3 所示。这些是电压矢量角，并且 A 相、B 相和 C 相的脉冲宽度值分别表示为 *Ta*、*Tb* 和 *Tc*，其中 *Ta*、*Tb* 和 *Tc* 波形均相差 120° 。具体来讲，*Tb* 比 *Ta* 滞后 120° ，而 *Tc* 比 *Ta* 超前 120° 。这些是根据 *speedRef*、*VdTesting* 和 *VqTesting* 的值生成的。可通过更改这些值来查看对波形的影响。检查电路板上的 PWM 测试点来观察 PWM 脉冲 (PWM-1H 至 3H 和 PWM-1L 至 3L) 并确保 PWM 模块运转正常。

6.2 使用 DAC 测试 SVGEN

为了实时监测内部信号值，使用了片上 DAC。DAC 是模拟模块的一部分。DAC B 和 C 可用于内部信号值的实时监测。如图 6-4 所示。

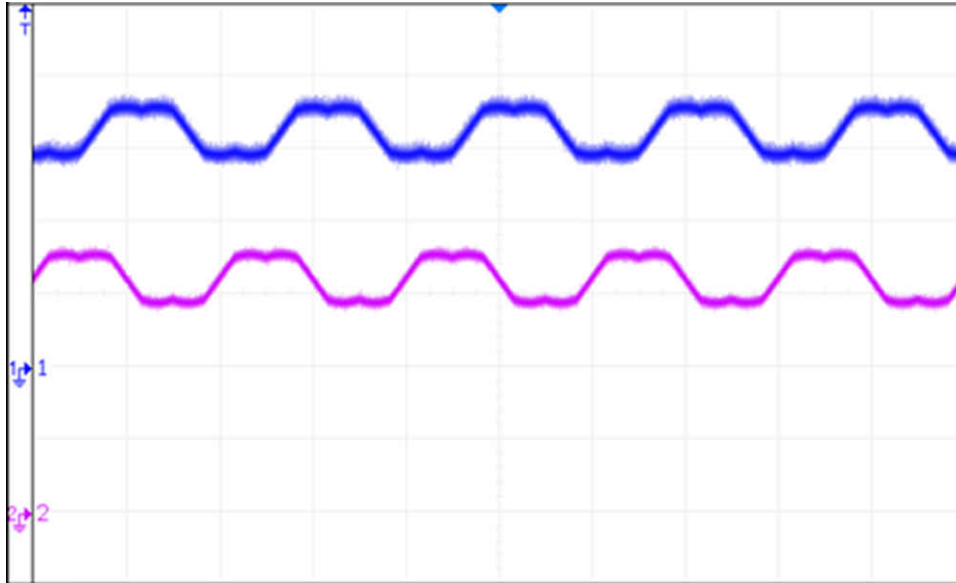


图 6-4. 显示 Ta 和 Tb 波形的 DAC 输出

6.3 逆变器功能验证

在验证了空间矢量生成模块和 PWM 模块之后，可以通过查看示波器上的逆变器输出 U、V 和 W 来测试三相逆变器硬件。可以将这些值与馈送到逆变器中的 PWM 脉冲 (PWM-1H 至 3H) 进行比较。建议在此测试期间逐渐增加直流总线电压。使用示波器检查逆变器输出 U、V 和 W (相对于逆变器 GND)，同时注意示波器的隔离要求。这可以确保逆变器正常工作。

7 增量构建级别 2

假设构建级别 1 已经成功完成，本节将验证逆变器和在 CLA 之外运行的 QEP 接口的过流保护限制。在此构建中，电机以开环方式运行。

电机可连接至 HVDMC 电路板，这是因为 PWM 信号已通过增量构建级别 1 成功验证。

1. 打开 `fcl_f2838x_tmdxiddk_settings_cpu1.h` 并通过将 BUILDLEVEL 设置为 BUILDLEVEL to FCL_LEVEL2 (`#define BUILDLEVEL FCL_LEVEL2`) 来选择 2 级增量构建选项。
2. 将 CURRENT_SENSE 选为 LEM_CURRENT_SENSE
3. 根据与电机相连的编码器，将 POSITION_ENCODER 选为 QEP_POS_ENCODER 或 T_FORMAT_ENCODER。
4. 右键点击工程名称，然后点击 *Rebuild Project* (重建工程)。
5. 编译完成时，点击 *Debug* (调试)，重置 CPU，重新启动，启用实时模式，然后运行。

图 7-1 显示了 2 级方框图。

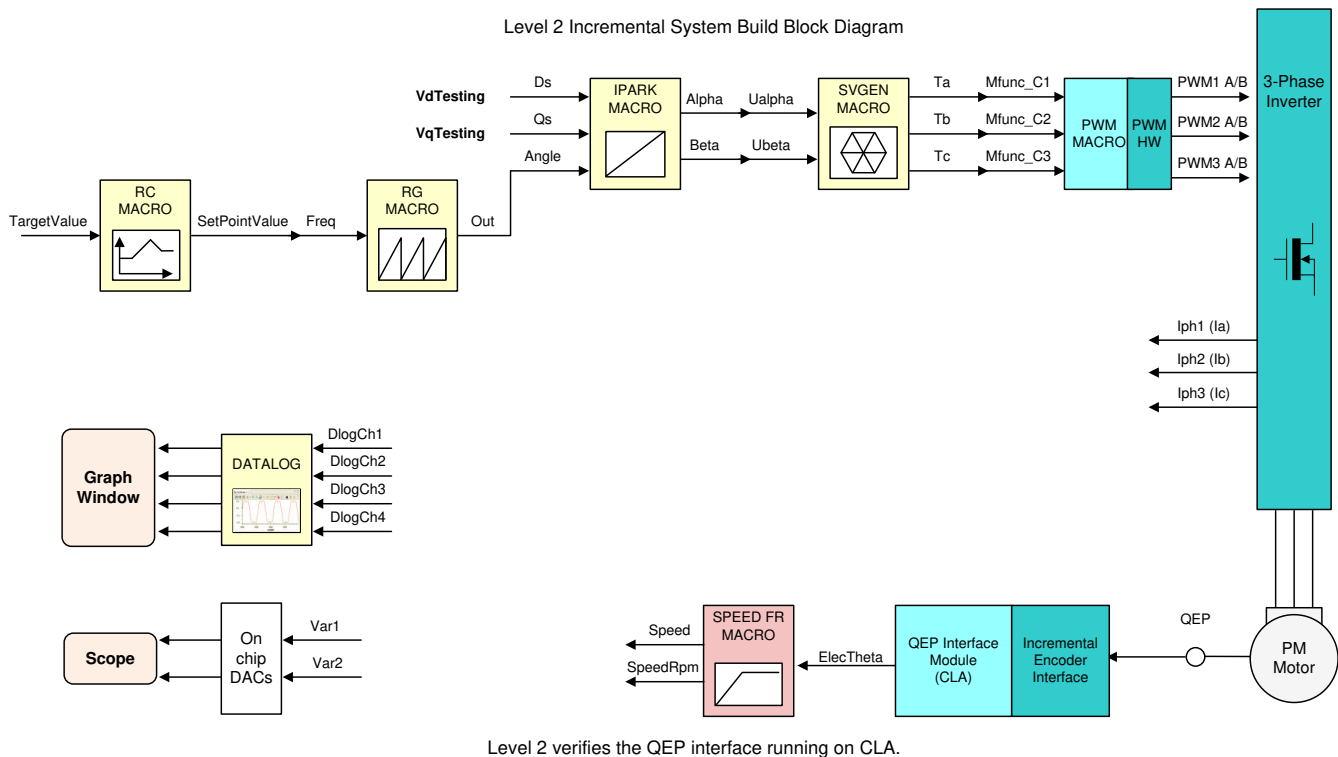


图 7-1. 2 级方框图

6. 在“Watch”（观察）窗口中将 `enableFlag` 设置为 1。在“Expressions”（表达式）窗口中看到 `isrTicker` 变量递增，可确认中断正常运转。现在将名为 `runMotor` 的变量设置为 `MOTOR_RUN`；如果对直流总线施加了足够的电压，则电机在几秒钟后开始旋转。

在软件中，若要调整的关键变量与上一级相同，下面将再次给出以供参考。

- a. `speedRef`：用于更改转子速度（标么值）
- b. `VdTesting`：用于更改 d 轴电压（标么值）
- c. `VqTesting`：用于更改 q 轴电压（标么值）

开环测试期间，必须仔细调节 PM 电机的 `VqTesting`、`speedRef` 和直流总线电压，这样，生成的 `Bemf` 将低于施加到电机绕组上的平均电压。这种调整可以防止电机失速或振动。

7.1 在软件中设置过流限制

该电路板具有多种电流采样方法，例如分流器、LEM 和 SDFM。分流器和 LEM 产生的信号由片上比较器子系统 (CMPSS) 模块进行过流监测。该模块具有一个可编程比较器和一个可编程数字滤波器。比较器产生保护信号。对于正电流和负电流，用户均可对比较器的基准进行编程。数字滤波器模块可以限定比较器输出信号，通过在特定

计数窗口的特定计数时间内定期采样和验证信号来确认信号完整性（其中的周期、计数和计数窗口可由用户编程）。

在“Expressions”（表达式）窗口中，用户可以看到以下变量：

- *clkPrescale* - 设置数字滤波器的采样频率
- *sampWin* - 设置计数窗口
- *thresh* - 设置最小数量以将信号限定在 *sampWin* 内
- *curLimit* - 同时通过分流器和 LEM 电流传感器设置最大允许电流

tripFlagDMC 是一个标志变量，代表逆变器的过流跳闸状态。如果设置了此标志，则可以调整以前的设置，并尝试通过将 *clearTripFlagDMC* 设置为 1 来重新运行逆变器。这将清除 *tripFlagDMC* 并重新启动 PWM。

默认电流限制设置是在 8A 时关断。所有这些设置都可以进行微调以适合您的系统。发现符合条件的值后，将这些值写下来，使用这些新值来修改代码，然后重新构建并重新加载以进行进一步的测试。

可以通过 H9 使用外部来源的数字信号关闭逆变器。当前没有提供代码，但可将其作为一项练习来试验和学习。

7.2 电流感测方法

为此工程选择的 CURRENT_SENSE 方法为 LEM_CURRENT_SENSE 或 SD_CURRENT_SENSE。通过在 *fcl_f2838x_tmdxiddk_settings_cpu1.h* 中设置此配置来选择电流感测方法，如下所示。

```
#define CURRENT_SENSE          SD_CURRENT_SENSE
or
#define CURRENT_SENSE          LEM_CURRENT_SENSE
```

7.3 电压感测方法

使用 Σ - Δ 滤波器模块 3 来感测电压。在“Expressions”（表达式）窗口中查找变量 *FCL_params.Vdcbus*。缓慢改变直流总线电压，并验证此变量是否正确跟踪此变化。例如，此变量将 100V 直流电压显示为 100.0。

7.4 设置电流调节器限制

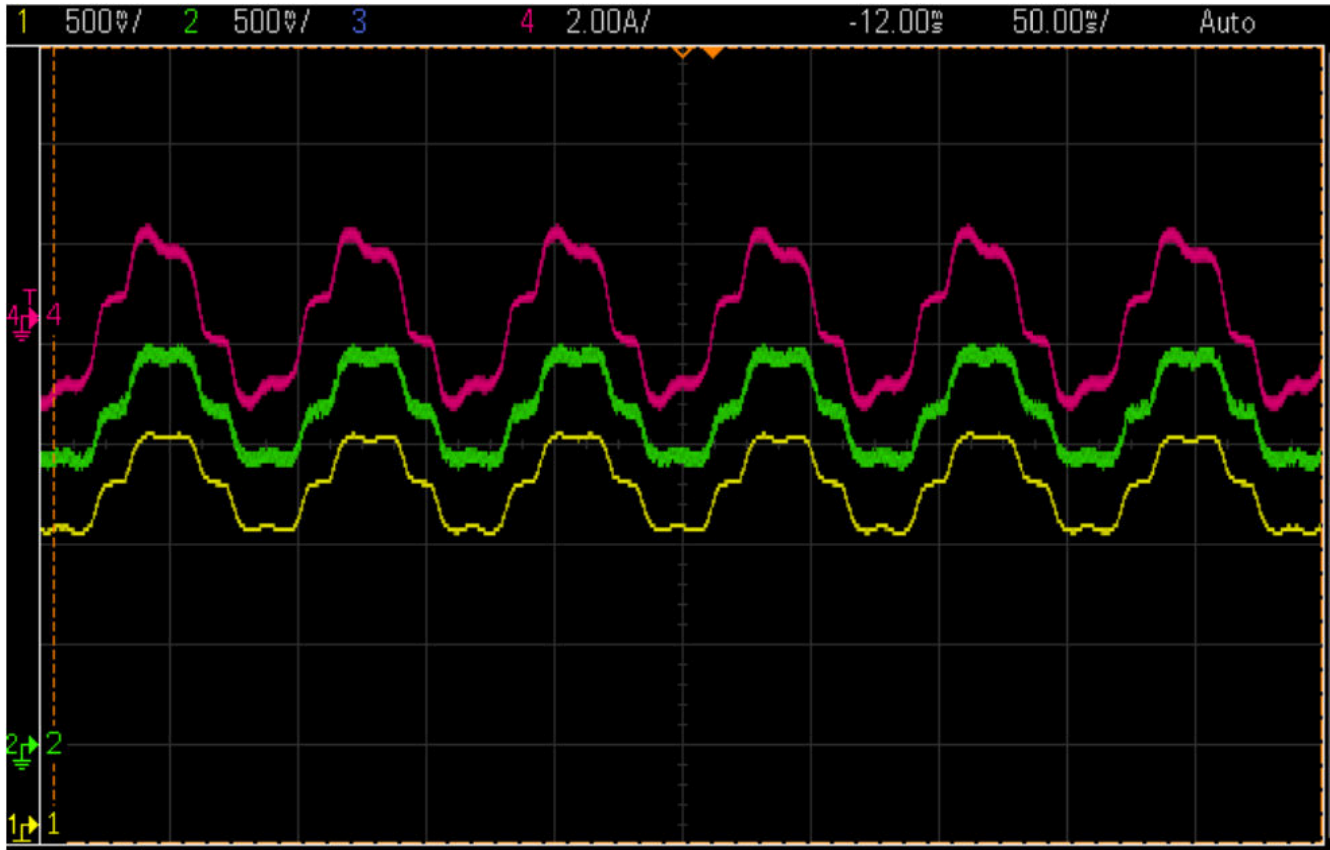
电流调节器的输出可以控制施加在 d 轴和 q 轴上的电压。d 和 q 输出的矢量总和必须小于 1.0，这表示 SVGEN 宏的最大占空比。在此特定应用中，最大允许占空比设置为 0.96。更高的计算速度可以实现更高的占空比运作，并且可以更好地利用直流总线电压。

在 PI 和复杂控制器模式下，电流调节器输出均由同一变量 *pi_id.out* 和 *pi_iq.out* 表示。调节器限制由 *pi_id.Umax/min* 和 *pi_iq.Umax/min* 设置。

将总线电压降为零、使控制器脱离实时模式并进行复位，从而使系统安全停止。

7.5 验证电流感测

在电机开环控制运行期间，使用 ADC 的 LEM 传感器和使用 SDFM 的分流电阻器测量电机相电流，DAC-B 和 DAC-C 上显示这两个感测电流波。图 7-2 显示了 IDDK 上从 H10 发出的这些信号及其示波器图。



CH4: Test Current Wave from Current Probe
CH2: DAC output, Current sense by LEM sensor with ADC
CH1: DAC output, Current sense by Shunt-resistor with SDFM

图 7-2. 测量电流和感测电流的示波器图

7.6 位置编码器反馈

在先前的所有测试中，位置编码器接口持续估算位置信息。所以，无需新代码即可验证位置编码器接口。

当命令电机运行时，电机将进入初始对齐阶段，在该阶段中，电角度和 QEP 角度值将设置为零。如果使用了旋转变压器或绝对编码器（EnDat 或 BiSS-C 或 T-Format），则会识别其电角度为零度时的初始位置以实现运行时校正。

7.6.1 速度观测器和位置估算器

绝对串行编码器的位置感应延迟通常处在几十微秒的数量级，无法与快速电流环路顺利协同，因为快速电流环路希望在从反馈采样时刻开始的 $1\mu\text{s}$ 至 $2\mu\text{s}$ 内施加控制动作。为解决此问题，可使用附带估算器的位置观测器来滤除位置测量中的抖动，并估算下一个采样时刻的位置。估算准确性取决于是否知道位置信息在 PWM 载波周期内的采样时间。

备注

本文档未具体介绍如何确定位置采样时刻，这个问题由您自行解决。在这里，位置采样在电机控制算法结束时启动，也就是 ADC SOC 在进行电流感应几微秒后开始。

7.6.2 位置编码器方向验证

DAC-C 上提供了测量出的或估算出的位置信息，而 DAC-B 上显示用于执行开环电机控制的参考位置 (*rg1.Out*)。图 7-3 显示了 IDDK 上从 H10 发出的这些信号及其示波器图。

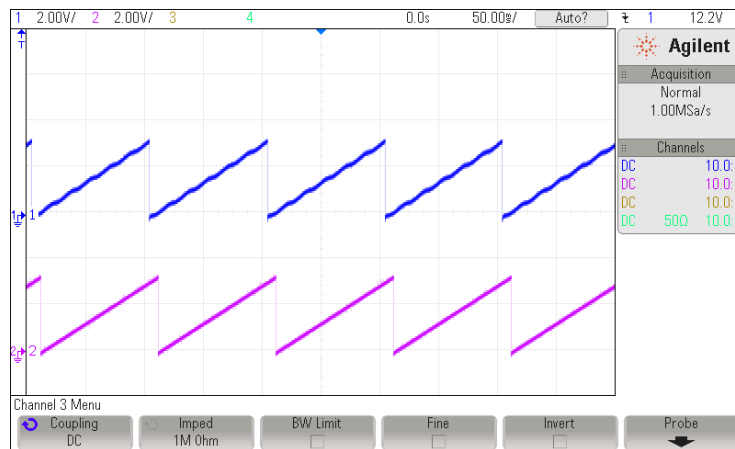
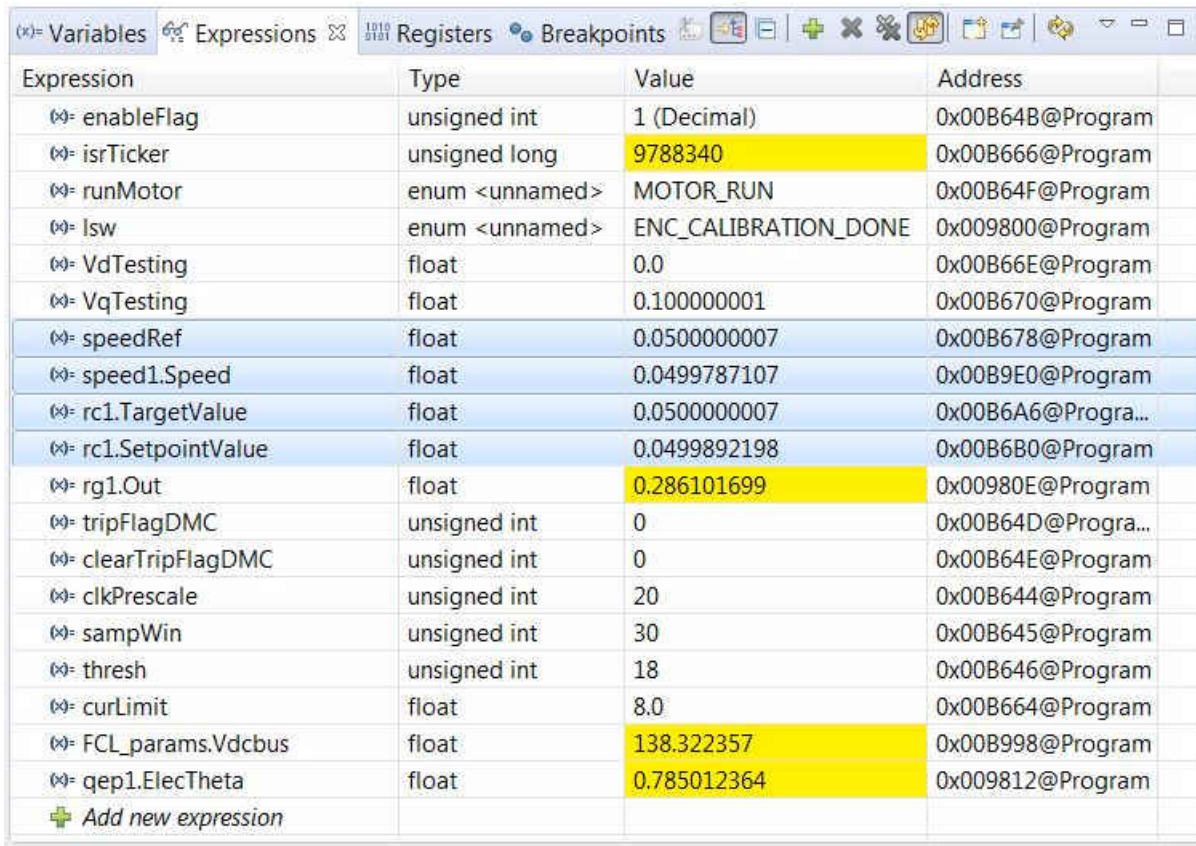


图 7-3. 参考角和转子位置的示波器图

通道 2 的波形代表的是参考位置，而通道 1 的波形代表的是估算位置。位置估算值的波动表明电机在运行时有一些较小的速度振荡。开环控制导致转子位置和参考位置无法对齐。但是，必须确保估算角度变化方向与参考方向相同；否则，表明电机旋转方向相反。为解决此问题，可以交换连接到电机的任意两根电线，或按照软件伪代码中的方法逆转角度估算（请参阅方程式 1）。

$$\text{angle} = 1.0 - \text{angle} \quad (1)$$

为确保速度估算正确，请在“Expressions”（表达式）窗口中更改 *speedRef* 变量（如图 7-4 所示）并检查估算速度变量 *speed1.Speed* 是否遵循命令速度。这是永磁同步电机，是没有转差的，因此不管控制是不是开环的，运行速度都遵循命令速度。



Expression	Type	Value	Address
enableFlag	unsigned int	1 (Decimal)	0x00B64B@Program
isrTicker	unsigned long	9788340	0x00B666@Program
runMotor	enum <unnamed>	MOTOR_RUN	0x00B64F@Program
lsw	enum <unnamed>	ENC_CALIBRATION_DONE	0x009800@Program
VdTesting	float	0.0	0x00B66E@Program
VqTesting	float	0.100000001	0x00B670@Program
speedRef	float	0.0500000007	0x00B678@Program
speed1.Speed	float	0.0499787107	0x00B9E0@Program
rc1.TargetValue	float	0.0500000007	0x00B6A6@Progra...
rc1.SetpointValue	float	0.0499892198	0x00B6B0@Program
rg1.Out	float	0.286101699	0x00980E@Program
tripFlagDMC	unsigned int	0	0x00B64D@Progra...
clearTripFlagDMC	unsigned int	0	0x00B64E@Program
clkPrescale	unsigned int	20	0x00B644@Program
sampWin	unsigned int	30	0x00B645@Program
thresh	unsigned int	18	0x00B646@Program
curLimit	float	8.0	0x00B664@Program
FCL_params.Vdcbus	float	138.322357	0x00B998@Program
qp1.ElecTheta	float	0.785012364	0x009812@Program
+ Add new expression			

图 7-4. 表达式窗口

测试完成时，减小总线电压、使控制器脱离实时模式并对其进行复位，从而使系统安全停止。现在，电机停止运转。

8 增量构建级别 3

假定前一节的任务已经成功完成，本节将验证 FCL 执行的 dq 轴电流调节。可以选择两种电流控制器之一：PI 控制器或复杂控制器。可以在调试窗口中设置控制器的带宽。

备注

在此构建中，控制是基于转子的实际位置进行的；因此，如果命令的 I_qRef 较高并且电机上没有负载，则电机能够以更高的速度运行。TI 建议在测试之前在电机上增加一些机械负载，或者应用较低的 I_qRef 值。当命令电机运行时，电机将进入初始对齐阶段，在该阶段中，电角和编码器角度计数将设置为零。

如果编码器是 QEP，则 QEP 计数将设置为零。电机将会基于强制角度运行，直至收到 QEP Index 脉冲为止。然后，电机可以基于其自身的角位置以完全自控模式运行。

如果编码器是绝对编码器，则对齐时的角度将为初始角度 ($initTheta$)，该角度此后会用作确定位置的参考。对齐完成后，电机可以立即基于其自身的角位置以完全自控模式运行。

以下初始步骤可以帮助评估此构建级别；

1. 打开 `fcl_f2838x_tmdxiddk_settings_cpu1.h` 并通过将 BUILDLEVEL 设置为 FCL_LEVEL3 (#define BUILDLEVEL FCL_LEVEL3) 来选择 3 级增量构建选项。
2. 通过将 FCL_CNTLR 设置为 PI_CNTLR 或 CMLPX_CNTLR，可以将电流环路调节器选为 PI 控制器或复杂控制器。
3. 将 CURRENT_SENSE 选为 LEM_CURRENT_SENSE
4. 根据耦合到电机的编码器，将 POSITION_ENCODER 选为 QEP_POS_ENCODER 或 T_FORMAT_ENCODER。
5. 每个 PWM 周期可以对电流和位置反馈进行一次或两次采样，具体取决于采样方法。在单采样方法中，采样与载波最大值同步，在双采样方法中，采样与载波最大值和载波零值同步。示例中通过将 SAMPLING_METHOD 选择为 SINGLE_SAMPLING 或将 SAMPLING_METHOD 选择为 DOUBLE_SAMPLING 来完成此采样方法的选择。最大调制指数从 SINGLE_SAMPLING 方法中的 0.98 变为 DOUBLE_SAMPLING 方法中的 0.96。如果 PWM_FREQUENCY 从 10kHz 改变，则最大调制指数也会改变。如果使用的是 T-format 编码器，请选择 SINGLE_SAMPLING，且 PWM 频率不超过 10kHz。如需更多信息，请参阅节 5.1.3.3。
6. 右键点击工程名称，然后点击 **Rebuild Project** (重建工程)。编译完成时，点击 **Debug** (调试) 按钮，重置 CPU，重新启动，启用实时模式，然后运行。

在软件中，要添加、调整或监视的关键变量汇总如下：

- `maxModIndex`：最大调制指数
- `IdRef`：用于更改 d 轴电流 (标么值)
- `IqRef`：用于更改 q 轴电流 (标么值)
- `FCL_params.WccD`：d 轴电流环路的推荐带宽
- `FCL_params.WccQ`：q 轴电流环路的推荐带宽
- `fclLatencyInMicroSec`：显示 ADC 采样与 PWM 更新之间的延迟 (以 μs 为单位)
- `fclClrCntr`：这是一个标志，用于清除变量 `fclLatencyInMicroSec` 并使其刷新
- `runMotor`：这是一个标志，用于运行或停止电机

图 8-1 显示了 3 级方框图。

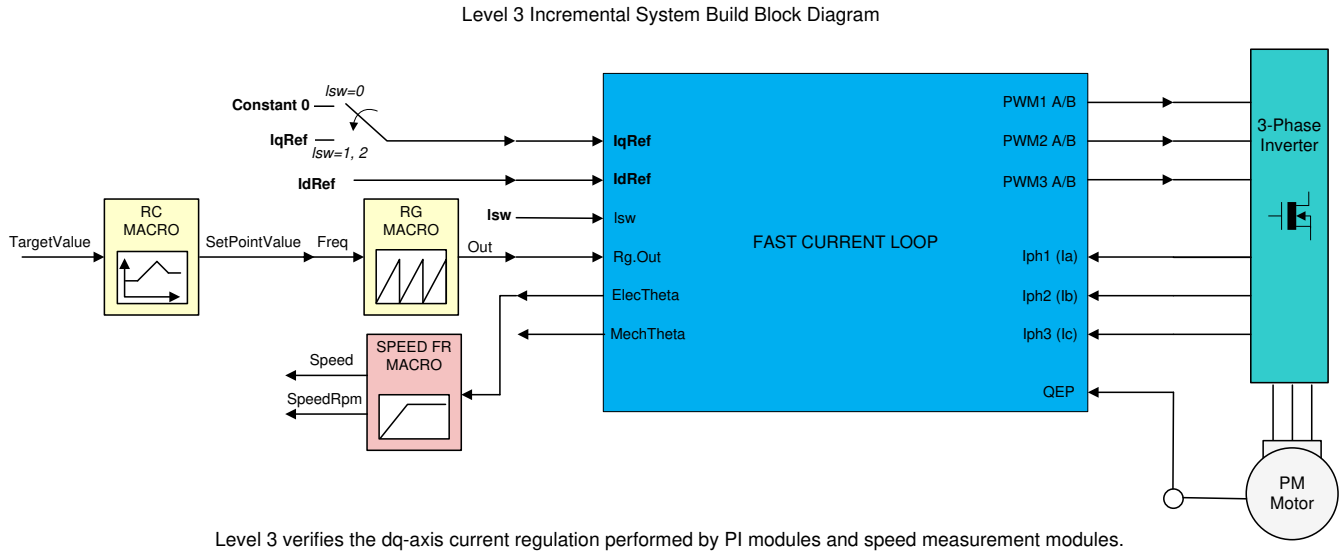


图 8-1. 3 级方框图，显示了最内层环路 - FCL

关键步骤解释如下：

1. 在“Watch”（观察）窗口中将 *enableFlag* 设置为 1。在“Watch”（观察）窗口中看到 *isrTicker* 变量递增，可确认中断正常运转。
2. 验证 *maxModIndex* 值是否为 0.96（双采样方法）或 0.98（单采样方法）。
3. 将 *speedRef* 设置为 0.3pu（如果基速不同的话，设置为其他合适的值），将 *IdRef* 设置零，并将 *IqRef* 设置为 0.03pu（或其他合适的值）。使用 QEP 编码器时，*speedRef* 仅在收到 QEP Index 脉冲之前用于强制模式运行电机。此后会根据电机的转子位置来控制电机。另一方面，使用绝对编码器时则没有这种强制模式，电机在对齐后立即根据转子位置运行。
4. 将自耦变压器/直流电源的电压逐渐增加至特定的电压，例如额定电压的 20%。
5. *Isw* 是一个状态机变量，表示与正交编码器交互的功能状态。其各种状态如下：
 - a. *Isw* = ENC_ALIGNMENT --> 将电机的转子锁定为与定子 A 相对齐
 - b. *Isw* = ENC_WAIT_FOR_INDEX --> 电机处于运行模式并等待 QEP Index 脉冲的第一个实例（仅在使用 QEP 编码器时适用）
 - c. *Isw* = ENC_CALIBRATION_DONE --> 电机处于运行模式 -（表示出现 QEP Index 脉冲，并在在使用 QEP 时完成校准或在使用其他编码器时完成校准）
6. 将 *runMotor* 标志设置为 MOTOR_RUN 以运行电机。
7. 请注意，*Isw* 状态按顺序自动增加（代码在 FCL 库中）
8. 使用连续刷新功能在“Watch”（观察）窗口中检查 *pi_id.fbk*，查看其是否可以跟踪 *IdRef*。
9. 使用连续刷新功能在“Watch”（观察）窗口中检查 *pi_iq.fbk*，查看其是否可以跟踪 *IqRef*。
10. 若要确认这两个电流调节器模块，请尝试不同的 *pi_id.ref* 和 *pi_iq.ref* 值，方法是分别更改 *IdRef* 和 *IqRef* 的值。
11. 通过调整 *FCL_params.wccD* 和 *FCL_params.wccQ* 的值，为电流环路尝试不同的带宽。带宽的默认设置是采样频率的 1/18。
12. 如果可以固定电机轴，则可以在 0.5 与 -0.5 之间来回更改 *IqRef* 值，以便研究环路带宽带来的影响。
13. 降低总线电压、使控制器脱离实时模式并进行复位，从而使系统安全停止。现在，电机停止运转。

8.1 观察结果一 - PWM 更新延迟

8.1.1 使用“Expressions”（表达式）窗口

在此构建级别和后续构建级别运行电机时，请在“Expressions”（表达式）窗口中观察变量 *fclLatencyInMicroSec*。

图 8-2 显示了“Expressions”（表达式）窗口的快照。

fclLatencyInMicroSec	float	0.959999979	0x0000B032@Data
FCL_params.wccD/(2*3.14)	double	555.837371	
FCL_params.wccQ/(2*3.14)	double	555.837371	

图 8-2. “Expressions” (表达式) 窗口延迟快照

此变量指示在反馈采样和 PWM 更新之间经过的时间长度。经过时间 (或延迟) 是根据 PWM 更新后 EPWM 计时器立即开始的计数计算得出的。此处显示的值比实际更新时间长几个时钟周期。将 *runMotor* 标志设置为 **MOTOR_RUN** 并且电机开始运行后, 由于代码中的初始设置, 延迟时间会立即显示为大约 $1.25\mu\text{s}$ 。此延迟大小是在占空比适中时发生的, 因此可以接受。在此时间段之后, 可以通过将 *fclClrCntr* 设置为 1 来刷新延迟时间。不管 **SAMPLING_METHOD** 是多少, 对于给定的 **FCL_CNTLRLR**, 延迟都保持不变。当 **FCL_CNTLRLR** 为 **PI_CNTLRLR** 时, 延迟约为 $0.96\mu\text{s}$, 相比之下, 设置为 **CMPLX_CNTLRLR** 时延迟为 $0.98\mu\text{s}$ (请参阅以下注意事项)。

备注

- 通过使用**代码内联**和其他优化技术, 可以将这些时间进一步减小约 $0.1\mu\text{s}$ 的区间。由于评估代码为库格式, 因此有一定的开销。
- 为确保采集到更清晰的信号, ADC 的**采样窗口**保持足够的宽度。根据电路板布局布线和馈入 ADC 通道的电路的不同, 可以将这个时间窗口减小近 60%。

8.1.2 使用示波器图

备注

因为未填充 H7，所以 GPIO16 和 GPIO18 用于计时目的，而不是根据设计用于功能分配。如果要填充 H7，请注释掉关联的代码并恢复功能分配。

图 8-3 以示波器图的形式显示了前面讨论的延迟，其中通道 2 和通道 1 波形的上升沿分别表示 ADC SoC 事件的实例和所有 PWM 更新的完成状态。这些事件通过 GPIO16 和 GPIO18 发出，并且可以分别在 [Main]-R31 和 [Main]-R33 上探测到。示波器图中看到的时间可能会略大于 *fclLatencyInMicroSec*。CPU 需要这段额外的时间以从 FCL 库返回并设置 GPIO。

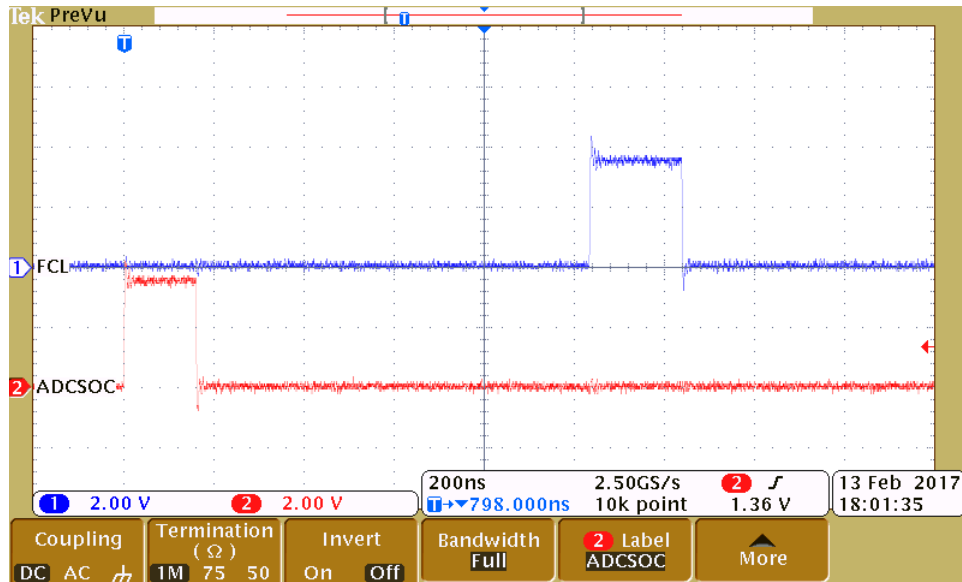


图 8-3. ADCSoC 和 FCL 完成事件的示波器图

9 增量构建级别 4

假定前一节的任务已经成功完成，本节将验证速度 PI 模块和速度环路。当命令电机运行时，电机将进入初始对齐阶段，在该阶段中，电角度和位置编码器角度值将设置为零。确保稳定对齐后，电机开始运行。

1. 打开 `fcl_f2838x_tmdxiddk_settings_cpu1.h` 并通过将 BUILDLEVEL 设置为 FCL_LEVEL4 (#define BUILDLEVEL FCL_LEVEL4) 来选择 4 级增量构建选项。
2. 通过将 FCL_CNTLR 设置为 PI_CNTLR 或 CMLPX_CNTLR，可以将电流环路调节器选为 PI 控制器或复杂控制器。
3. 将 CURRENT_SENSE 选为 LEM_CURRENT_SENSE
4. 根据与电机相连的编码器，将 POSITION_ENCODER 选为 QEP_POS_ENCODER 或 T_FORMAT_ENCODER。
5. 将 SAMPLING_METHOD 选为 SINGLE_SAMPLING 或 DOUBLE_SAMPLING。如果使用的是 T-format 编码器，请选择 SINGLE_SAMPLING，且 PWM 频率不超过 10kHz。如需更多信息，请参阅 节 5.1.3.3。
6. 右键点击工程名称，然后点击 *Rebuild Project* (重建工程)。编译完成时，点击 *Debug* (调试) 按钮，重置 CPU，重新启动，启用实时模式，然后运行。

在软件中，要调整的关键变量汇总如下：

- *speedRef* : 用于更改转子速度 (标么值)。
- *IdRef* : 用于更改 d 轴电流 (标么值)。
- *IqRef* : 用于更改 q 轴电流 (标么值)。

图 9-1 显示了实现方框图。

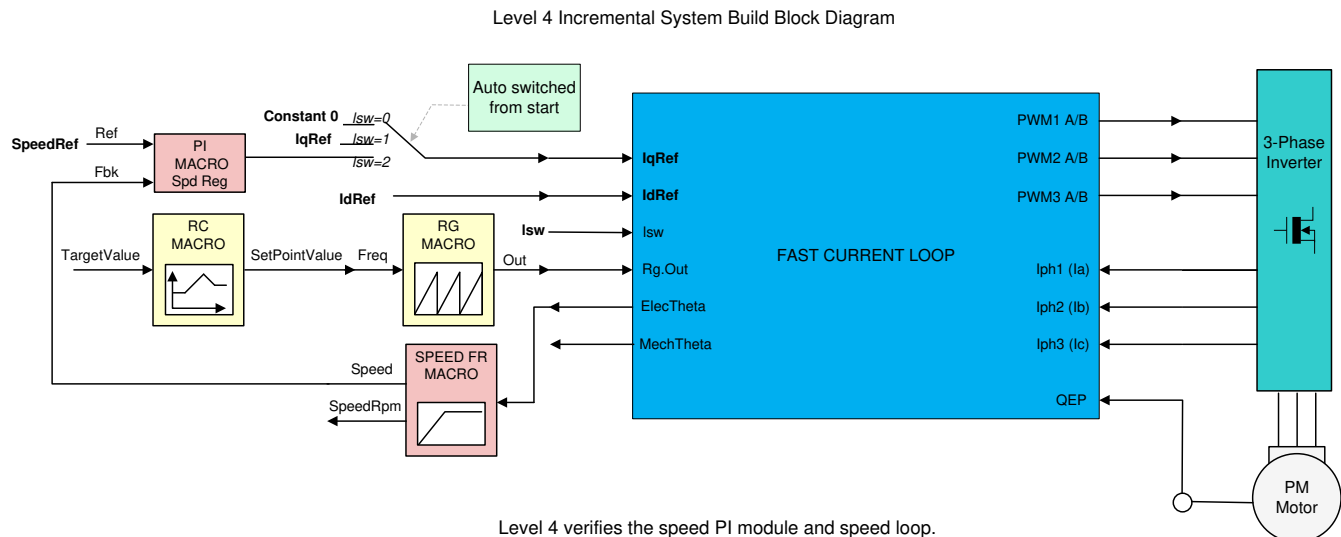


图 9-1. 4 级方框图，显示了具有内层 FCL 的速度环路

关键步骤可解释如下：

1. 在“Watch”（观察）窗口中将 *enableFlag* 设置为 1。在“Watch”（观察）窗口中看到 *isrTicker* 变量递增，可确认中断正常运转。
2. 将 *speedRef* 设置为 0.3pu（如果基速不同的话，设置为其他合适的值）。
3. 向“Expressions”（表达式）窗口添加 *pid_spd* 变量
4. 逐渐增加自耦变压器上的电压以获得适当的直流总线电压。
5. 将 *runMotor* 设置为 MOTOR_RUN
6. 请注意，状态机变量 (*Isw*) 会按顺序自动增加，其状态如下：
 - a. *Isw* = ENC_ALIGNMENT --> 将电机的转子锁定为与定子 A 相对齐
 - b. *Isw* = ENC_WAIT_FOR_INDEX --> 电机处于运行模式并等待 QEP Index 脉冲的第一个实例（仅在使用 QEP 编码器时适用）
 - c. *Isw* = ENC_CALIBRATION_DONE --> 电机处于运行模式 -（表示出现 QEP Index 脉冲，并在在使用 QEP 时完成校准或在使用其他编码器时完成校准）
7. 使用持续刷新功能来比较该速度 (*speed1.Speed*) 与“Watch”（观察）窗口中的 *speedRef*，查看速度是否几乎相同。
8. 若要确认此速度 PID 模块，请尝试使用不同的 *speedRef* 值（正值或负值）。可以调整 P、I 和 D 增益来获得满意的响应。
9. 在超低速度范围内，速度响应的性能严重依赖于位置编码器提供的转子位置角精度。
10. 降低总线电压、使控制器脱离实时模式并进行复位，从而使系统安全停止。现在，电机停止运转。

图 9-2 显示了同步坐标系中定子电流的磁通分量和扭矩分量。

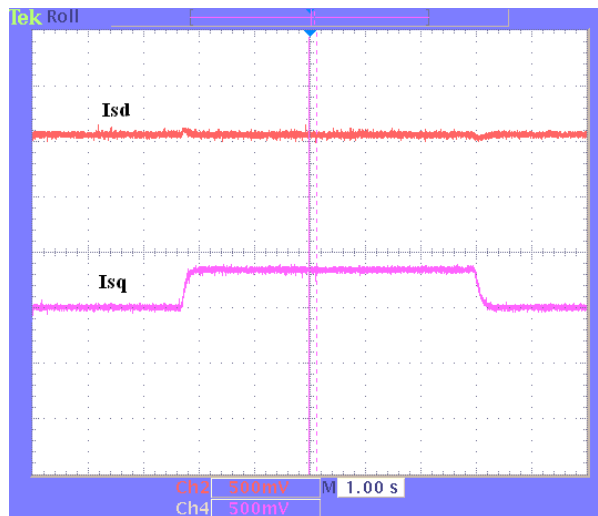


图 9-2. 0.33pu 步进负载和 0.3pu 速度下的同步坐标系内定子电流的磁通分量和扭矩分量

9.1 观察结果

在默认设置中，对于 `CMPLEX_CNTL`R 和 `PI_CNTL`R，电流环路带宽均设置为 `SAMPLING FREQUENCY`（采样频率）的 $1/18$ 。随着带宽的增加，控制器会对反馈中的细微扰动做出反应以试图进行纠正，因此控制变得困难，电机在运行时的噪声变得更大，请参阅以下注意事项。增益交叉频率（或开环带宽）可以达到 `SAMPLING FREQUENCY`（采样频率）的 $1/6$ ，并在包括较高速度的在内的整个速度范围内仍能获得良好的瞬态响应。如果出现任何故障偶尔使电机的旋转方向反转，请尝试通过将 `runMotor` 设置为 `MOTOR_STOP` 并随后再次设置为 `MOTOR_RUN` 来重新启动电机。对于 QEP，在从 `Isw = ENC_WAIT_FOR_INDEX` 过渡到 `Isw = ENC_GOT_INDEX` 时可能需要一些微调。这一任务可以用作修复练习。

备注

快速电流环路有助于实现高带宽。当设计的带宽很高时，环路增益也很高。这在很大程度上使环路性能与电流反馈质量密切相关。如果进入数字域的电流反馈信号的 `SNR` 较小，则由于控制器试图使误差最小化，环路会产生很大的噪声。如果噪声令人厌烦，可能需要减小带宽以避免可闻噪声。

10 增量构建级别 5

本节将验证位置 `PI` 模块和位置环路。为了使该环路正常工作，必须成功完成速度环路。当命令电机运行时，电机将进入初始对齐阶段，在该阶段中，电角度和位置编码器角度值将设置为零确保稳定对齐后，电机开始运行。

1. 打开 `fcl_f2838x_tmdxiddk_settings_cpu1.h` 并通过将 `BUILDLEVEL` 设置为 `FCL_LEVEL5` (`#define BUILDLEVEL FCL_LEVEL5`) 来选择 5 级增量构建选项。
2. 通过将 `FCL_CNTL`R 设置为 `PI_CNTL`R 或 `CMPLEX_CNTL`R，可以将电流环路调节器选为 `PI` 控制器或复杂控制器。
3. 将 `CURRENT_SENSE` 方法选为 `LEM_CURRENT_SENSE`。
4. 根据与电机相连的编码器，将 `POSITION_ENCODER` 选为 `QEP_POS_ENCODER` 或 `T_FORMAT_ENCODER`。
5. 将 `SAMPLING_METHOD` 选为 `SINGLE_SAMPLING` 或 `DOUBLE_SAMPLING`。如果使用的是 `T-format` 编码器，请选择 `SINGLE_SAMPLING`，且 `PWM` 频率不超过 `10 KHz`。如需更多信息，请参阅 [节 5.1.3.3](#)。
6. 右键点击工程名称，然后点击 `Rebuild Project`（重建工程）。编译完成时，点击 `Debug`（调试）按钮，重置 `CPU`，重新启动，启用实时模式，然后运行。

关键步骤可解释如下：

1. 在“`Watch`”（观察）窗口中将 `enableFlag` 设置为 `1`。在“`Watch`”（观察）窗口中看到 `isrTicker` 变量递增，可确认中断正常运转。
2. 向“`Expressions`”（表达式）窗口中添加变量 `pi_pos`、`posArray`、`ptrMax` 和 `posSlewRate`。
3. 逐渐增加自耦变压器上的电压以获得适当的直流总线电压。
4. 将 `runMotor` 设置为 `MOTOR_RUN` 以运行电机。电机必须在旋转以跟随命令的位置（如果电机不能正常运转，请参阅以下注意事项）。
 - a. 电机通过由 `refPosGen()` 模块设定的预定义运动曲线和位置设置来运行。该模块基本上通过一组在数组 `posArray` 中定义的值来循环位置基准。这些值表示相对于初始对齐位置的转数。一旦达到数组中定义的某个位置值，电机会暂停一会儿，然后再转向数组中的下一个位置。因此，可以将这些数组值称为停驻位置。在从一个停驻位置过渡到下一个停驻位置期间，过渡速率（或速度）由 `posSlewRate` 设置。从第一个值重新开始之前在 `posArray` 中通过的位置数由 `ptrMax` 决定。因此，请向“`Expressions`”（表达式）窗口中添加变量 `posArray`、`ptrMax` 和 `posSlewRate`。
5. 可以将 `posArray` 中的停驻位置更改为不同的值，以确定电机是否按照设置的转数旋转。
6. 也可以更改停驻位置数量 `ptrMax` 以设置旋转模式。
7. 可以使用 `posSlewRate` 更改位置压摆率。该速率表示每个采样瞬间的角度（以 `pu` 为单位）。
8. 为获得满意的响应，可重新调整速度和位置 `PI` 控制器的比例和积分增益。TI 建议先调整速度环路，然后再调整位置环路。
9. 降低总线电压、使控制器脱离实时模式并进行复位，从而使系统安全停止。现在，电机停止运转。

图 10-1 显示了实现方框图。

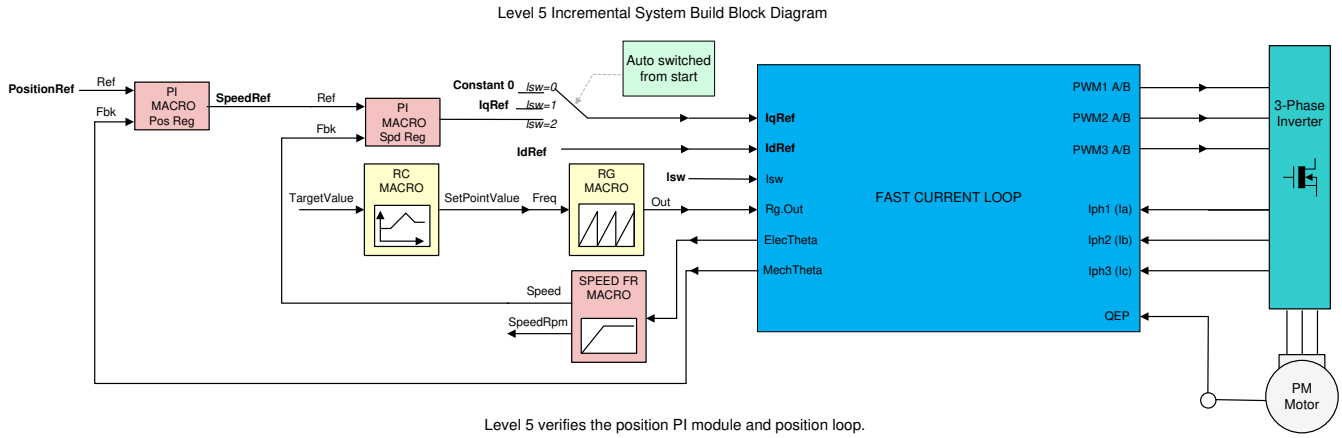


图 10-1. 5 级方框图，显示了具有内层 FCL 的位置环路

在图 10-2 显示的示波器图中绘制了位置参考和位置反馈。它们存在可以忽略不计的滞后，这可能是软件造成的。如果没有正确选择位置环路控制器的 Kp 和 Ki 增益，则可能导致反馈振荡或响应滞后。

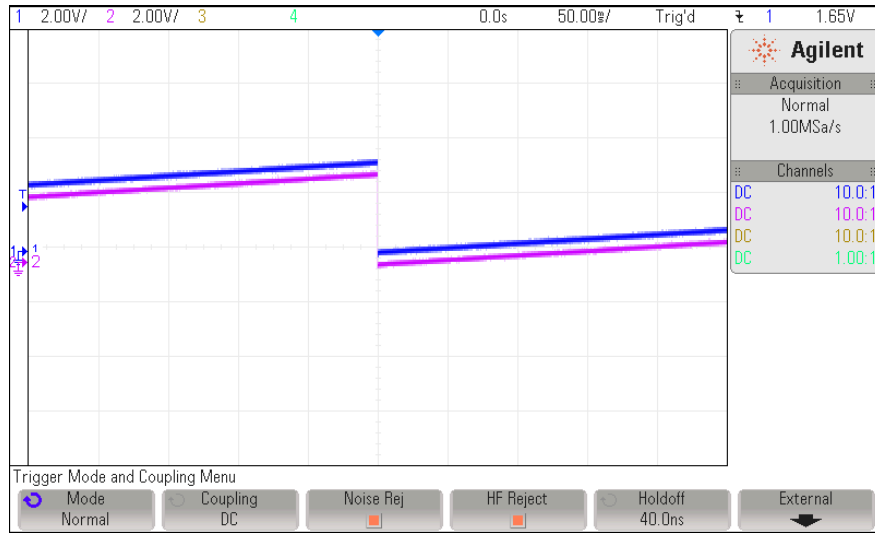


图 10-2. 参考位置到伺服和反馈位置的示波器图

备注

- 如果电机响应不稳定，则电机轴和编码器的旋转方向可能相反。交换与电机任何两相的连接，然后重复进行测试。
- 此处实现的位置控制基于初始对齐的电气位置 (= 0)。如果电机有多个极对，这种对齐方式可能使轴处于不同的机械位置，具体取决于转子的预启动机械位置。如果需要机械位置可重复性或一致性，则必须使用 QEP Index 脉冲来设置一个参考点。这一任务可以用作练习。对于绝对编码器，这可能是显而易见的。

11 增量构建级别 6

假设先前的构建级别已成功完成，则此构建级别将尝试使用 C2000 的软件频率响应分析器 (SFRA) 工具来研究快速电流环路的频率响应分析 (该工具在 DigitalPower SDK 中作为库提供)。

11.1 集成 SFRA 库

C2000™ 软件频率响应分析器 (SFRA) 库和 SDK 框架中的补偿设计工具介绍了 SFRA 工具并指导您如何将此工具集成到 C2000 平台中，其位于以下位置：

C:\ti\c2000\C2000Ware_DigitalPower_SDK_<version>\libraries\sfra\Doc

嵌入式固件以库的形式在 DigitalPower SDK 中提供，其位于以下位置：

C:\ti\c2000\C2000Ware_DigitalPower_SDK_<version>\libraries\sfra

SFRA GUI 以可执行应用程序的形式在 DigitalPower SDK 中提供，其位于以下位置：

C:\ti\c2000\C2000Ware_DigitalPower_SDK_<version>\libraries\sfra\gui

以下位置提供了一些用于了解 SFRA 的示例工程：

c:\ti\c2000\C2000Ware_DigitalPower_SDK_<version>\libraries\sfra\examples

在 ISR 代码中，有两个函数为 SFRA 注入噪声，然后从环路中收集反馈数据，这两个函数是：

- `injectSFRA()`
- `collectSFRA()`

这些函数的作用在名称上是不言自明的。它们应按代码中使用的顺序使用，从而以正确的顺序收集数据。

备注

- 由模拟信号路径和量化引起的干扰将影响环路性能，并妨碍可以使用 SFRA 结果进行验证的高带宽选择。因此，提供具有较高 SNR 的电流反馈非常重要。
- 在评估电流环路时，如果可以使电机速度保持恒定，则将有助于最大限度地减小 SFRA 结果中与速度抖动相关的误差带来的影响。这在研究 Iq 环路时特别有用。如果电流环路的电压去耦良好，则此要求可能无关紧要。

该工具可用于研究 D 轴或 Q 轴电流环路或速度环路。电机可以在不同的速度/负载条件下和不同的带宽下运行，并可以在每种条件下评估性能。可以看出，控制器可以在所有这些条件下提供设计的带宽并有一定的容差。

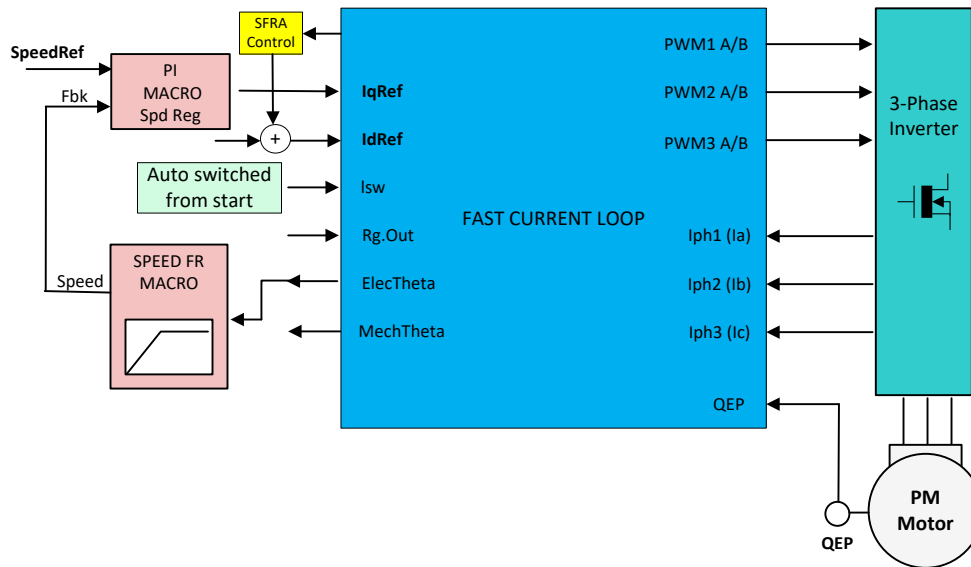


图 11-1. 6 级方框图

图 11-1 中给出了实现方框图。SFRA 工具以各种频率将噪声信号注入系统，分析系统响应，并提供测试过程中看到的实际物理系统的波特图。

11.2 启动 SFRA 前的初始设置

设置过程涉及在调试环境和 SFRA GUI 之间进行调整。在熟悉 SFRA GUI 与目标平台的连接步骤之前，宜切断目标平台的高压电源输入。

以下初始步骤可以帮助评估此构建级别：

1. 打开“`fcl_f2838x_tmdxiddk_settings_cpu1.h`”并通过将 BUILDLEVEL 设置为 FCL_LEVEL6 (#define BUILDLEVEL FCL_LEVEL6) 来选择 6 级增量构建选项。

2. 通过将 FCL_CNTLRL 设置为 PI_CNTLRL 或 CMLPX_CNTLRL，可以将电流环路调节器选为 PI 控制器或复杂控制器。
3. 将 CURRENT_SENSE 方法选为 LEM_CURRENT_SENSE。
4. 根据与电机相连的编码器，将 POSITION_ENCODER 选为 QEP_POS_ENCODER 或 T_FORMAT_ENCODER。
5. 将 SAMPLING_METHOD 选为 SINGLE_SAMPLING 或 DOUBLE_SAMPLING。如果使用的是 T-format 编码器，请选择 SINGLE_SAMPLING，且 PWM 频率不超过 10 KHz。如需更多信息，请参阅 节 5.1.3.3。

打开 “fcl_f2838x_sfra_settings_cpu1.h” 并注意以下定义：

- SFRA_FREQ_START
- SFRA_FREQ_LENGTH
- FREQ_STEP_MULTIPLY

这些定义为 GUI 分别规定了噪声频率的起始值、要扫描的不同噪声频率的数量以及连续扫描频率之间的比率。如需更多信息，请参见与 SFRA 相关的 [C2000™ 软件频率响应分析器 \(SFRA\) 库和补偿设计器用户指南](#)。在此评估工程中，重要的是在了解和理解这些参数的基础上对其进行调整，以进一步执行重复测试。

在此电机控制工程中，可以在三个控制环路的任何一个环路上运行 SFRA，例如速度环路、D 轴电流环路和 Q 轴电流环路。从技术上讲，这也可以在位置环路上运行，但并未包含在此工程范围内，如果需要，可以对此进行实验。

1. 右键点击工程名称，然后点击 “Rebuild Project”（重建工程）。编译完成时，点击 Debug（调试）按钮，重置 CPU，重新启动，启用实时模式，然后运行。在调试环境中，要执行的步骤如下所示。在 “Expressions 窗口”（表达式窗口）中添加以下变量：
 - a. sfraTestLoop：用于选择要在哪个控制环路上评估 SFRA，可选择的选项为：
 - i. SFRA_TEST_D_AXIS - D 轴电流环路
 - ii. SFRA_TEST_Q_AXIS - Q 轴电流环路
 - iii. SFRA_TEST_SPEEDLOOP - 速度环路

关键步骤可解释如下：

1. 在 “Watch”（观察）窗口中将 enableFlag 设置为 1。在 “Watch”（观察）窗口中看到 IsrTicker 变量递增，可确认中断正常运转。
2. 与 GUI 进行通信所需的 SCI 初始化工作现在应该全部完成。
3. 设置 GUI 连接到目标平台后，将执行有关调试窗口的其他步骤。

11.3 SFRA GUI

有两种可用于执行频率响应分析的 GUI，一个 (SFRA_GUI) 可以绘制开环和受控体波特图，另一个 (SFRA_GUI_MC) 可以绘制开环和闭环波特图。可以调用这些 GUI 并连接到目标平台以研究控制环路。在 节 11.1 中所述的位置提供了 GUI 可执行文件。

双击选择的 GUI 可执行文件，然后将出现 GUI 屏幕，如 图 11-2（对于 SFRA_GUI）或 图 11-3（对于 SFRA_GUI_MC）中所示。两个 GUI 几乎相同，不同之处可参见 “FRA Settings” 下的下拉菜单（从标签 “Open Loop” 开始）。

- 在 SFRA_GUI 中，使用此下拉菜单可选择 Open Loop Model 或 Plant Model。
- 在 SFRA_GUI_MC 中，使用此下拉菜单可选择 Open Loop Model 或 Closed Loop Model。

备注

这两个 GUI 对带宽的解释是不同的。

- 在 SFRA_GUI 中，带宽定义为开环增益交叉频率
- 在 SFRA_GUI_MC 中，带宽是根据中国的伺服驱动器标准 **GBT 16439-2009** 和 **NEMA ICS 16 (速度环路)** 定义的。该 GUI 将带宽定义为速度环路的闭环输出幅度下降 3dB 或相移滞后超过 90° 的频率。这表示带宽是这两个实例中第一个实例发生时的频率。在本演示中，该方法也用于分析电流环路。

本演示使用 GUI “SFRA_GUI_MC”。但是，建议您也尝试使用另一个 GUI 来研究受控体。使用 SFRA_GUI 时，可以通过在函数 `configureSFRA()` 中传递参数 “SFRA_GUI_PLOT_GH_CL” 来绘制与 SFRA_GUI_MC 中相同的图，如下面的代码片段所示。

```
configureSFRA(SFRA_GUI_PLOT_GH_CL, ISR_FREQUENCY); // to plot GH and CL plots using SFRA_GUI
```

但是，从图得出的推论与 SFRA_GUI_MC 中的推论并不相符。所以，建议为 “SFRA_GUI_PLOT_GH_H” 配置 SFRA，以便您可以使用两个 GUI (一次使用一个) 来查看开环图、闭环图和受控体模型图。对于数字电源应用，SFRA_GUI_MC 推论可能不适用，因此 SFRA_GUI 也可以用于显示闭环图。

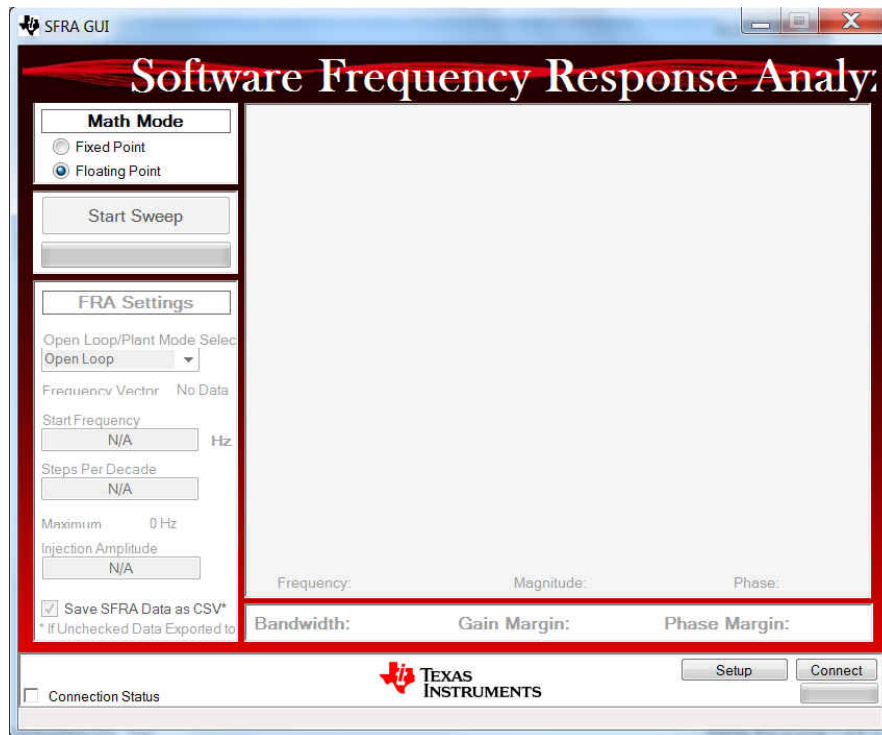


图 11-2. SFRA GUI

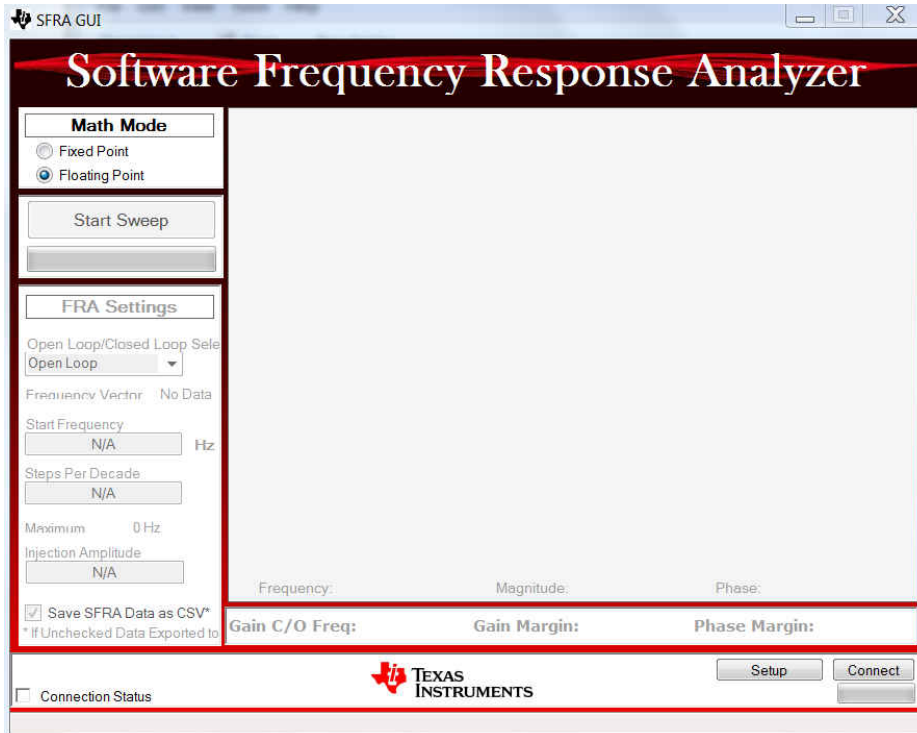


图 11-3. SFRA GUI MC

11.4 设置 GUI 以连接到目标平台

两个 GUI 采用相同的过程连接到目标平台。使用 GUI 可实现根据目标平台和开发计算机选择适当的设置。

下面列出了开始分析之前在 GUI 上要执行的操作：

1. 数学模式：根据目标 C2000 开发平台，选择定点还是浮点选项。对于 F2837x 或更高版本的 C2000 MCU，请选择“Floating Point”（浮点）选项
2. controlCARD/LaunchPad 上的 USB 端口已连接到计算机以用于 JTAG，因此不需要其他连接。但是，若要独立运行，需要将 USB 连接器连接到目标板。在 controlCARD/LaunchPad 上的 XDS100 仿真器中，除了 JTAG 链路外，还提供了 SCI 端口链路，而 GUI 会使用此链路连接到目标平台的 SCI 端口。在 CCS 的调试环境使用 JTAG 时，GUI 也可以同时使用 SCI。
3. 点击右下角的 Setup (设置) 按钮。此时会弹开一个 Setup Connection (设置连接) 窗口，如图 11-4 所示。

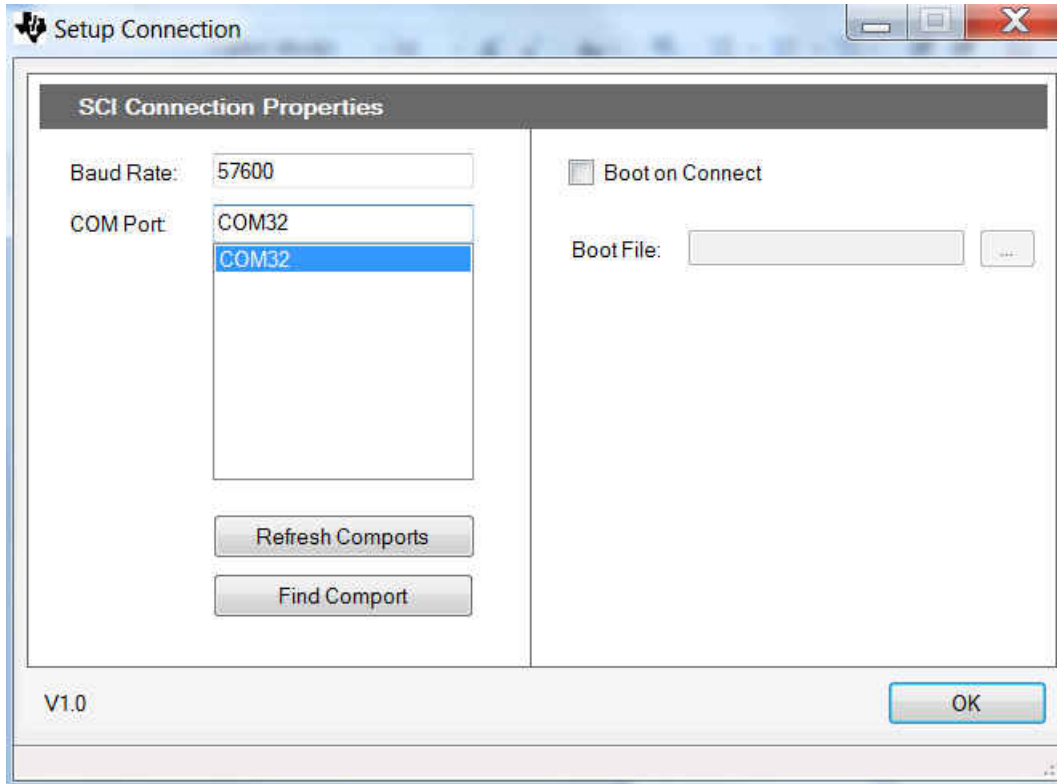


图 11-4. GUI 设置图

4. 点击“Refresh Comports”（刷新 Comport）按钮，使 Comport 号码显示在窗口中。
5. 选择代表与目标 C2000 板连接的 Comport。
6. 取消选中“Boot on Connect”（启动连接）
7. 点击“OK”（确定）按钮

8. 此时应该会建立与 MCU controlCARD/LaunchPad 的连接，并且 GUI 的显示结果如图 11-5 所示（在左下角显示连接状态）。

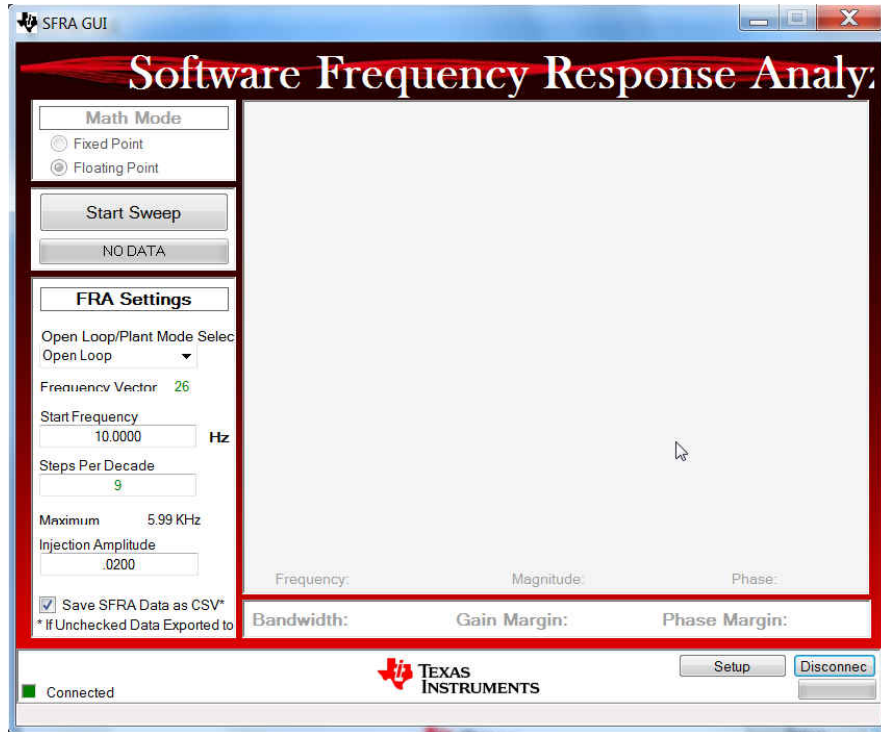


图 11-5. 连接到 C2000 MCU 的 SFRA GUI

9. “FRA Settings”（FRA 设置）面板显示与频率扫描相关的设置。这些值已从 C2000 器件中预先填充，可以保留原样。
10. 如前所述，在“FRA Settings”（FRA 设置）下的下拉菜单中（从 Open Loop 开始），有两种外观不同的 GUI 可选。
- 在 SFRA_GUI 中，使用此下拉菜单可选择 Open Loop Model 或 Plant Model
 - 在 SFRA_GUI_MC 中，使用此下拉菜单可选择 Open Loop Model 或 Closed Loop Model
 - 在各种频率下对系统完成噪声注入扫描后，此菜单可用。然后，您可以使用此菜单选择并查看选择的图。
 - 带宽报告与前面提到的不同。SFRA_GUI_MC 中的开环图显示的是增益交叉频率，而 SFRA_GUI 中显示的是带宽。

这样就完成了 GUI 环境的初始设置。

11.5 运行 SFRA GUI

如果先前切断了目标平台的高压 (HV) 电源输入，请立即将其恢复。在调试环境中，要执行的步骤如下所示：

1. 确认 `sfraTestLoop` 设置为 `SFRA_TEST_D_AXIS` 以便测试 Id 环路。
2. 将 `FCL_params.wccD` 设置为所需的值（在限制范围内）（对 Q 轴执行测试时，请针对 Q 轴调整此参数 - `FCL_params.wccQ`）
3. 设置 `speedRef = 0.05`（以 pu 为单位， $1\text{pu} = 250\text{Hz}$ ），然后设置 `runMotor = MOTOR_RUN` 以运行电机。现在，电机轴应开始旋转并以命令的速度稳定下来。
4. 状态机变量 (`!sw`) 会按顺序自动增加，其状态如下：
 - a. `!sw = ENC_ALIGNMENT` --> 将电机的转子锁定为与定子 A 相对齐
 - b. `!sw = ENC_WAIT_FOR_INDEX` --> 电机处于运行模式并等待 QEP Index 脉冲的第一个实例（仅在使用 QEP 编码器时适用）
 - c. `!sw = ENC_CALIBRATION_DONE` --> 电机处于运行模式 -（表示出现 QEP Index 脉冲，并在在使用 QEP 时完成校准或在使用其他编码器时完成校准）
5. 现在，通过点击 GUI 中的“Start Sweep”（开始扫描）按钮，可以调用 GUI 来进行 D 轴电流环路的频率扫描。扫描进度将由标记为“NO DATA”（无数据）的位置中的绿色条表示。
6. 频率扫描全部完成后，将计算波特图并显示结果（如图 11-6 和图 11-7 所示）。
7. GUI 还会计算并显示环路带宽、增益裕度和相位裕度。
8. 如果需要，可以通过在不同的速度和负载条件下更改 `FCL_params.wccD` 来重复该测试。
9. 若要断开 GUI，请点击 GUI 上的“Disconnect”（断开）按钮。
10. 若要停止电机，请降低高压直流输入电压并将 `runMotor` 设置为 `MOTOR_STOP`。
11. 电机停止后，使控制器退出实时模式并进行复位。

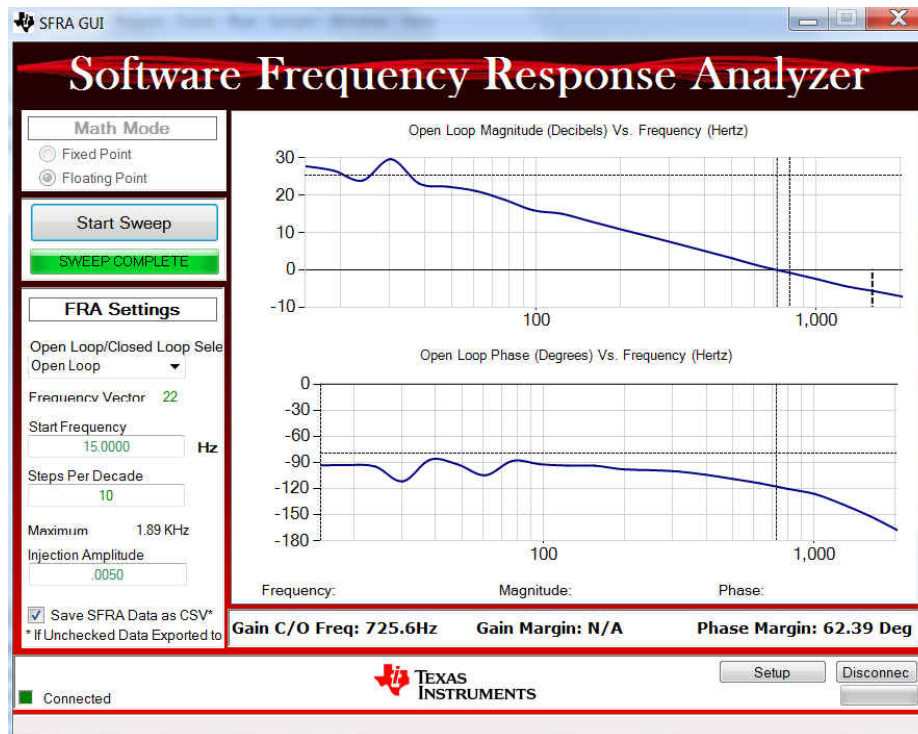


图 11-6. 显示幅度和相角的电流环路 SFRA 开环波特图

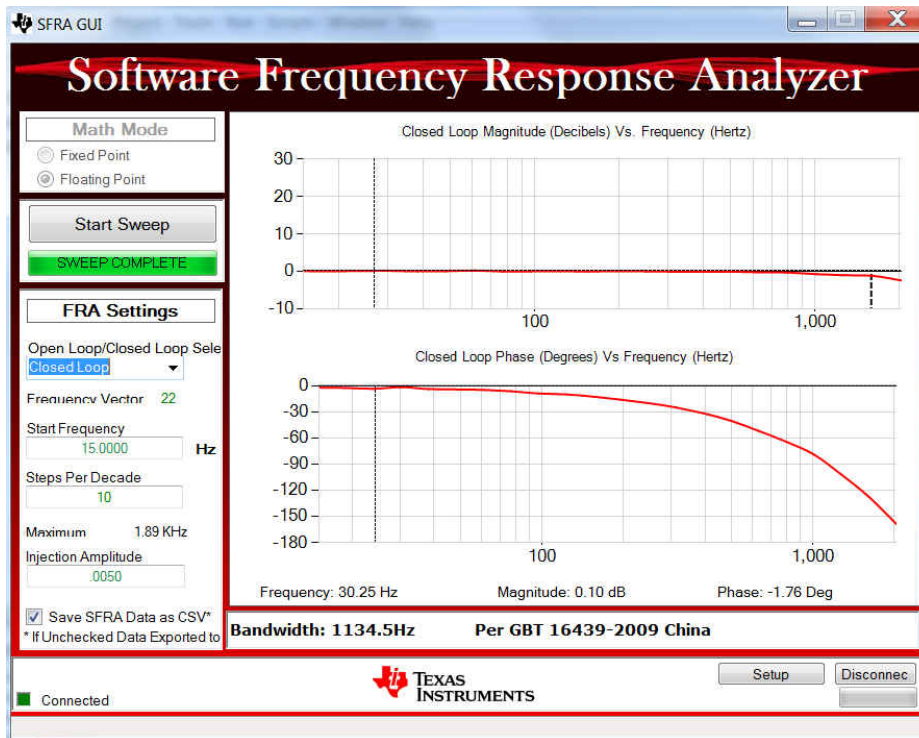


图 11-7. 显示幅度和相角的电流环路 SFRA 闭环波特图

11.6 电流反馈 SNR 的影响

快速电流环路有助于实现高带宽。当设计的带宽很高时，环路增益也很高。这在很大程度上使环路性能与电流反馈质量密切相关。如果进入数字域的电流反馈信号的 SNR 较小，则由于控制器试图使误差最小化，环路会产生很大的噪声。如果噪声令人厌烦，可能需要减小带宽以避免可闻噪声。因此，为了获得更高的带宽和性能，反馈应具有更高的 SNR 以便获取频率响应，如图 11-8 和图 11-9 所示。

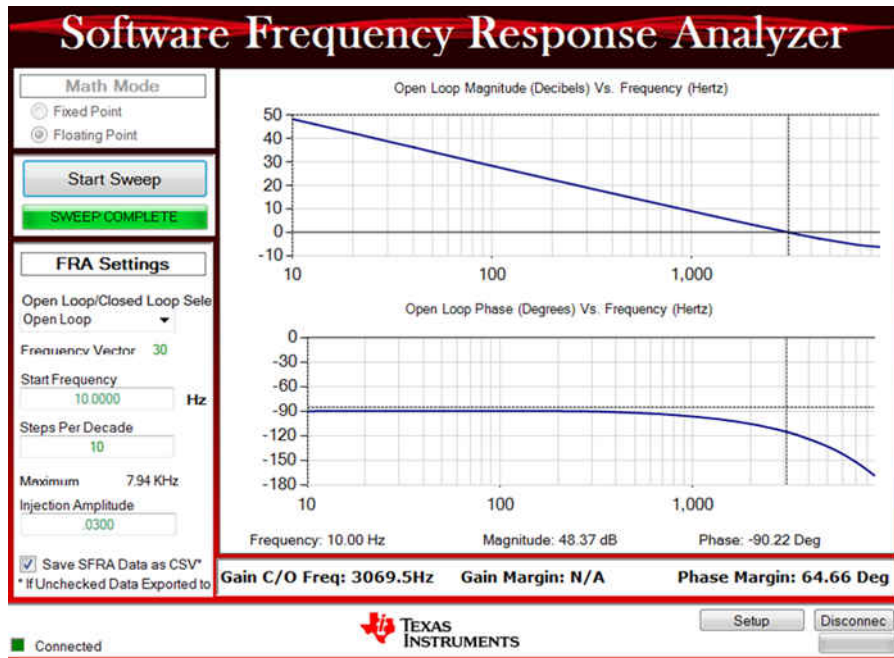


图 11-8. 电流环路的 SFRA 开环波特图 - 具有高 SNR 的电流反馈

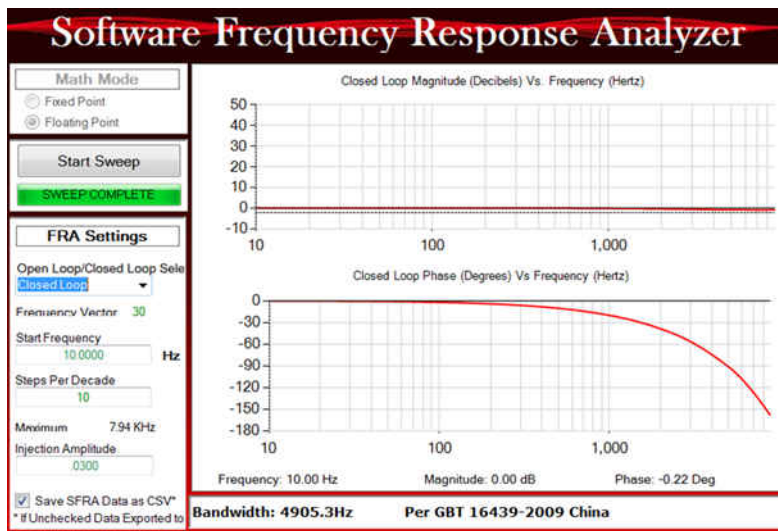


图 11-9. 电流环路的 SFRA 闭环波特图 - 具有高 SNR 的电流反馈

备注

借助硬件平台 IDDK，可以提高馈入 MCU ADC 的电流信号的 SNR。所以，在此平台上，更高带宽的测试可能会产生更多噪声（本质上是抖动）。

11.7 推论

11.7.1 根据闭环图确定带宽

实现图 11-8 和图 11-9 所示开环图和闭环图的控制器是无差拍控制器，其中输出仅在一个采样周期内就能赶上输入，而不会出现任何过冲，也无需多个周期。从闭环图中可以看出，闭环增益在所有频率下始终为 0dB（单位增益），所以，基于幅度来确定带宽是不切实际的。因此，可以根据中国标准 GBT 16439-2009 或 NEMA ICS 16（速度环路），选择相位图作为参考，并将相位滞后超过 90° 的频率作为带宽。在此测试用例中，按照这些指导原则，将 PWM 频率选为 10kHz，采样频率为 20kHz，从闭环图获得的电流环路带宽约为 5000Hz。

11.7.2 根据开环图确定相位裕度

根据开环图，得知相位裕度约为 65°。高裕度将使得在实际带宽的各个频率范围内实现非常强大的性能。

11.7.3 根据 PWM 更新时间确定最大调制指数

根据节 8.1，反馈采样到 PWM 更新之间的时间间隔约为 0.9μs。在 PWM 频率为 10kHz 的这一系统中，最大调制指数受限于采样方法，如下所示

- 双采样 - 略高于 96%
- 单采样 - 大约 98%

这一数字与基于 FPGA 的系统非常接近；在基于 FPGA 的系统中，整个算法都在硬件中实现。

11.7.4 电流环路中的电压去耦

可以在零速下执行 SFRA 测试（如果需要，可以采用将轴牢固锁定的方式完成）来获得一张图作为参考。零速时的增益交叉频率和相位裕度可能会降低。然后，在不同的速度和负载条件下，可以重复进行此测试以验证带宽或相位裕度是否有任何变化。该图在不同速度下的任何变化都表示电流环路中去耦的质量。

11.8 相位裕度与增益交叉频率间的关系

通过改变控制带宽并重复这些测试，然后记下所产生的增益交叉频率和相位裕度，可获得 Id 环路的一组图（如图 11-10 所示）。此处执行了两组测试，一组基于典型的电流控制方法，另一组则基于 FCL。这两组测试都是使用不同的电流调节器进行的。这些测试都给出了汇总结果。

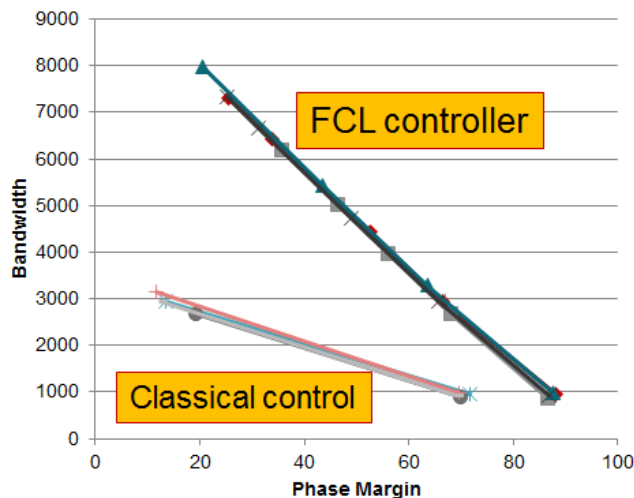


图 11-10. 增益交叉频率与相位裕度（通过实验获得）间的关系图

底部的一组图是在常规控制中获得的，很明显，增益交叉频率太低，并且随着增益交叉频率的增加，相位裕度以更快的速度下降。

顶部的一组图是使用 FCL 获得的。对于给定的相位裕度，增益交叉频率几乎是传统方法的三倍。而且，随着增益交叉频率的增加，相位裕度的相对下降与传统方法相比非常低。这实际上意味着 FCL 可以在更高的相位裕度下提供更高的带宽或增益交叉频率。

12 增量构建级别 7

此构建级别涉及在 F2838x 器件中设置 EtherCAT 从站，以便与 PC 中的 TwinCAT 主站连接。在 F2838x MCU 的三个内核（两个 C28x CPU 内核 (CPU1/2) + 一个 Arm Cortex-M4 内核 (CM)）中，EtherCAT 外设可以连接到 CPU1 或 M4。在此处的示例中，选择 M4（也被称为连接管理器 (CM)）与 EtherCAT 外设交互，而 C28x 内核 (CPU1) 将执行伺服驱动器的所有控制功能。设置过程包括 CPU1 设置 EtherCAT 所需的 GPIO 和时钟，然后将 EtherCAT 所有权分配给连接管理器 (CM)。然后，需要 CM 上的专用代码，以通过处理器间通信外设 (IPC) 在 EtherCAT 从站和 CPU1 之间传递数据。所有这些都 FCL_LEVEL7 中进行处理。

12.1 在 CPU1 上运行代码以将 ECAT 分配给 CM

1. 打开“`fcl_f2838x_tmdxiddk_settings_cpu1.h`”并通过将 BUILDLEVEL 设置为 FCL_LEVEL7 (#define BUILDLEVEL FCL_LEVEL7) 来选择 7 级增量构建选项。右键单击工程名称，然后单击 *Rebuild Project*（重建工程）。

备注

此构建级别仅需要 IDDK 的控制电源，因此，您可以关闭套件的高压直流电源。

2. 编译完成时，单击 *Debug*（调试）按钮，重置 CPU，重新启动，启用实时模式，然后运行。
3. 在“CCS Expressions”（CCS 表达式）窗口中右键单击鼠标，然后选择“Import”（导入）。然后，从调试目录中浏览至并选择 `fcl_f2838x_tmdxiddk_ecat_vars.txt` 文件。这将在“Expressions Window”（表达式窗口）中添加以下变量：
 - a. `countMainLoop`：仅显示 ECAT 主循环函数中的活动的计数器
 - b. `dataBufferFromCM`：来自 CM 的消息的缓冲区副本
 - c. `dataBuffertoCM`：到 CM 的消息的缓冲区副本
 - d. `ipcCMtoCPUDataBuffer`：由 CM 写入供 CPU1 使用的数据缓冲区
 - e. `ipcCPUtoCMDDataBuffer`：由 CPU1 写入供 CM 使用的数据缓冲区
4. 运行该代码将启用 EtherCAT 所需的 GPIO，并将 EtherCAT 从站外设的所有权传递给 CM，此外，还将在 CPU1 侧设置 IPC，以便在 CPU1 和 CM 之间进行数据传输。现在，就可以在 CM 上加载/运行代码以初始化 EtherCAT 外设了。

备注

如果 EtherCAT 在前不久的任何评估工程中被分配给 CPU1，而现在被分配给 CM，请确保在运行 CM 工程之前对 controlCARD 下电上电，以确保 CPU1 已完全出让 EtherCAT 外设的所有权。

12.2 在 CM 上运行代码以设置 ECAT

演示示例中使用 ECAT 仿真伺服驱动器的调试面板，以便提供特定的命令设置并检索特定驱动器参数的状态。因此，CM 上的演示示例是针对此应用调整的 EtherCAT 从站堆栈代码的预编译演示。

备注

修改演示中给出的要求时，需要通过 SSC 工具生成新的从站堆栈文件，还需要对应用程序源代码进行相应的更改。必须重新构建该文件以获得新的可执行文件，从而加载到 CM 中。

TI 为提供 EtherCAT 支持而给出的源文件仅为该函数所需的 HAL 文件，不足以编译和生成可执行文件。但是，如果您生成从站堆栈，此处提供的 CM 工程将能够生成可执行文件。

您应遵循以下步骤：

1. 确保 CPU1 有效运行工程（如上一节所述），将 EtherCAT 所有权移交给 CM
2. 除了 controlCARD 与计算机之间面向 JTAG 用途的 USB 连接外，还应在 controlCARD RJ45 端口 0 和计算机之间连接以太网电缆
3. 运行 `f2838x_connected_drive_ssc_file_and_demo.exe` 安装程序以提取 F2838x SSC 配置、SSC 工具所需的器件系统文件以及将在 CM 上运行的演示可执行文件。这些内容将位于新创建的 `ssc_configuration` 目录中。
4. 在调试窗口中，单击 `Cortex_M4`，右键单击并选择 *Connect Target*（连接目标）。这将有助于下载 M4 可执行文件。

5. 在 CCS 调试视图中，点击 *Run --> Load --> Load Program*，然后浏览至可执行文件 *fcl_f2838x_ecat_cm.out*（位于 *\\solutions\tdm\iddk379d\df2838x\ssc_configuration\cm*）。这是在上面的步骤 3 中提取的。
6. 运行代码。这将配置 ECAT 从站控制器，此外还配置 CM 侧的 IPC，以便在 CM 和 CPU1 之间进行数据传输。

现在，ECAT 从站控制器已准备好连接到 ECAT 主控制器，在本演示中，ECAT 主控制器是在用户的开发计算机上运行的 TwinCAT。

12.3 设置 TwinCAT

TwinCAT 软件将几乎所有兼容的 PC 机变为具有多 PLC 系统、NC 轴控制、编程环境和操作站的实时控制器。TwinCAT 是一款 PC 软件，可用作 EtherCAT 主站来控制连接到 PC 的各种 EtherCAT 从站节点。以下是设置 TwinCAT 的步骤

1. 可选：安装 Microsoft Visual Studio。这不是必需的，因为如果找不到 Visual Studio 程序，TwinCAT 将安装 Visual Studio Shell。
 - a. 下载并安装 [Microsoft Visual Studio](#)
 - b. TwinCAT 支持与 Visual Studio 2010/2012/2013/2015/2017 集成
2. 从 Beckhoff 下载并安装 [TwinCAT3](#)
 - a. 按照左侧栏进入 *Download->Software->TwinCAT 3->TE1xxx | Engineering* 并选择软件产品 *TwinCAT 3.1 eXtended Automation Engineering (XAE)*
3. 安装完成后，验证 TwinCAT 运行时是否处于激活状态
 - a. 检查 TwinCAT “Config Mode” 图标是否显示在 Windows 通知面板中，如图 12-2 所示。



图 12-1. TwinCAT 配置模式图标

- b. 右键点击此图标并选择 **Tools->TwinCAT Switch Runtime**。从“Tc- SwitchRuntime”窗口中，验证它处于 **active** (激活) 状态。处于激活状态时，它将仅提供 **Deactivate** (停用) 选项。请勿停用！

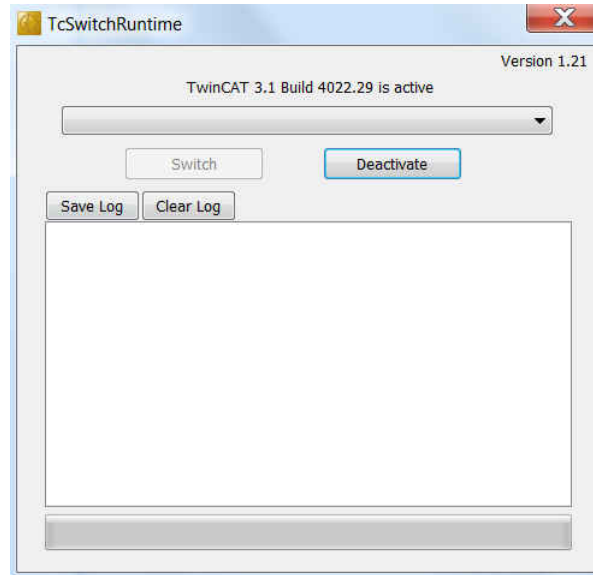


图 12-2. TcSwitchRunTime 窗口已激活

- c. 如果此图标不存在，则从文件系统中找到 TwinCAT 运行时可执行文件。(默认安装位置通常为：C:/TwinCAT/TcSwitchRuntime)
4. 使用以下方法之一通过 TwinCAT 启动 Visual Studio :
 - a. 推荐：右键点击 Windows 通知面板中的“TwinCAT Config Mode”图标，然后选择 **TwinCAT XAE**
 - b. 已安装的桌面图标：**TwinCAT XAE**
 - c. 使用 Beckhoff 文件夹下已安装的“Start Menu”图标：**TwinCAT XAE**
 5. Visual Studio 运行后，验证主工具栏已显示选项 **TwinCAT** 和 **PLC**。如果这些不存在，则 TwinCAT Switch Runtime 不处于激活状态。
 6. 在 Visual Studio 中，创建一个新的 EtherCAT 工程。选择 **File -> New -> Project**，然后在模板下依次选择 **TwinCAT Projects** 和 **TwinCAT XAE Project (XML format)**。填写名称 (如 **f2838x_iddk**)，然后点击 **OK**。
 7. 现在工程已经创建，请验证是否安装了实时以太网适配器。
 - a. 在 Visual Studio 中，从主工具栏中选择 **TwinCAT** 菜单，然后选择 **Show Realtime Ethernet Compatible Devices**
 - b. 在弹出窗口中，在 **Installed and ready to use devices (realtime capable)** 类别下，如果未显示任何连接，请从兼容器件列表中选择，然后点击 **Install**。
 8. TwinCAT 设置现已完成。

12.4 通过 TwinCAT 扫描 EtherCAT 器件

使用以下步骤通过 TwinCAT 扫描 EtherCAT 器件：

1. 打开在节 12.3 中创建的 TwinCAT 工程 (**f2838x_iddk**)
2. 验证 controlCARD 是否正在运行节 12.2 中所述的演示代码 (CM 运行 **fcl_f2838x_ecat_cm.out**)，以及开发计算机 (运行 TwinCAT) 是否通过以太网电缆连接到 controlCARD 中的端口 0。

备注

端口 0 是具有两个以太网接口的 controlCARD 一侧上的顶部以太网端口。

3. 在 Visual Studio 左侧的解决方案资源管理器中，展开 **Project** (工程)，然后展开 **I/O**
4. 右键点击 **Devices** (器件)，然后选择 **Scan** (扫描)
 - a. 将弹出一个对话框，说明 **Not all types of devices can be found automatically** (并非所有类型的器件都能自动找到)。点击 **OK** (确定)。

5. 扫描完成后，将出现一个弹出窗口。可能会出现以下选项：
 - a. 一个弹出窗口指出 *1 new I/O devices found* (找到新的 I/O 器件)，其中器件为 *Device 2 (EtherCAT Automation Protocol)*。此器件编号或除 *Device 1* 以外的任何其他器件编号是正确的，请点击 *OK* (确定)。
 - b. 一个弹出窗口，说明未找到任何器件，或说明 *1 new I/O devices found* (找到新的 I/O 器件)，其中器件为 *Device 1 (EtherCAT Automation Protocol)*。这意味着某些设置不正确。验证该示例是否正在器件上运行 (或至少已完成 EtherCAT IP 的 GPIO 设置和复位)。如果正确遵循该程序，则应已识别 *Device 2*。可以小心地再次重复设置步骤。
6. 点击 *OK* (确定) 后，另一个弹出窗口将要求 *Scan for boxes* (扫描各框)。点击 *Yes* (是)。
7. 点击 *Yes* (是) 后，另一个弹出窗口将要求 *Activate Free Run* (激活自由运行)。点击 *Yes* (是)。
8. 在解决方案资源管理器左侧的器件下，您应该看到 *Device 2 (EtherCAT)*。此处将为 *Box #* (框 #)。此 *Box* (框) 是 controlCARD ESC。
9. EtherCAT 主站现在已设置为与从器件进行通信。

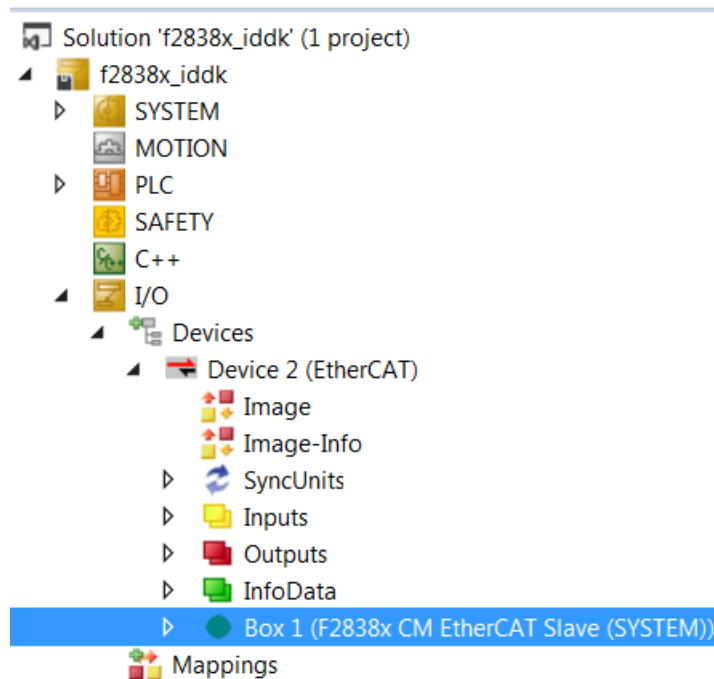


图 12-3. TwinCAT 解决方案资源管理器

12.5 针对 ESC 的 ControlCard EEPROM 编程

首先验证 TwinCAT 是否已发现 ESC，如节 12.4 所述。此外，在开始对此 EEPROM 进行编程之前，确保将 ESC XML 文件复制到 TwinCAT 目录中。

1. 将 ESI 文件 (*F2838x CM EtherCAT Slave (System).xml*)，位于 `\\solutions\tmdxiddk379d\l2838x\ssc_configuration`) 复制到 TwinCAT 目录中 (默认位置：`C:\TwinCAT\3.1\Config\Io\EtherCAT`)。如果 TwinCAT 已经打开，则必须关闭并重新打开它才能看到新的 XML 文件。
2. 在 Visual Studio 解决方案资源管理器中，双击 *Device 2 (EtherCAT)* 下的 *Box #* (框 #)。
3. TwinCAT 工程窗口应在解决方案资源管理器的右侧打开，并具有一些选项卡，如 “*General*” (常规)、 “*EtherCAT*” 等等
4. 选择 *EtherCAT* 选项卡，然后点击 *Advanced Settings* (高级设置)
5. 在新窗口中，展开 *ESC Access* (ESC 访问) 菜单，然后展开 *E2PROM* 菜单。点击 *Smart View* (智能视图)
6. 点击 *Write E2PROM* (写入 E2PROM) 并展开 *Available EEPROM* (可用 EEPROM) 说明窗口中的 *Texas Instruments Incorporated* 菜单。
7. 展开 *TI C28xx Slave Devices* (TI C28xx 从器件) 并选择 *F2838x CM EtherCAT Slave*。点击 *OK* (确认)。

8. Visual Studio 将指示正在对 EEPROM 进行编程。完成后，如果 Smart View (智能视图) 没有使用新内容自动更新，您可以选择 *Read E2PROM* (读取 E2PROM) 以读回新编程的值。
9. 此 CM 应用程序的产品代码为 *0x10003101*
10. 完成对 EEPROM 的编程后，断开控制器并对其下电上电。

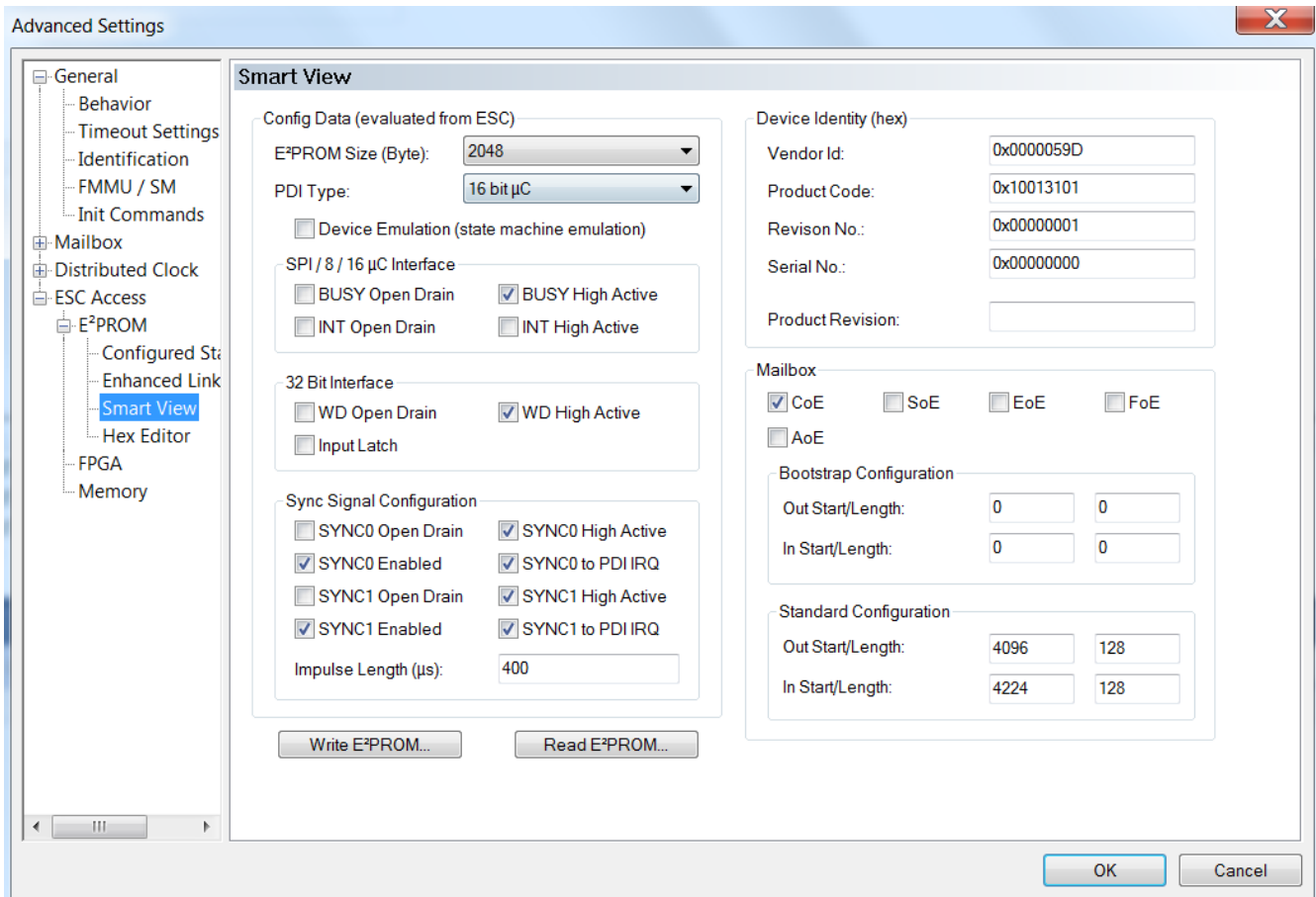


图 12-4. TwinCAT EEPROM 窗口

12.6 运行应用程序

本节介绍了运行应用程序。

1. 如果之前断开了连接，请验证或恢复连接。
2. 重新加载 CPU1 和 CM 应用程序
3. 首先运行 CPU1 并将 EtherCAT 外设移交给 CM
4. 接下来运行 CM，以设置 EtherCAT 与主站的连接以及 IPC 与 CPU1 的连接
5. 通过 TwinCAT 窗口，重新扫描器件，并在 *Config Mode* (配置模式) 下重新启动 TwinCAT。

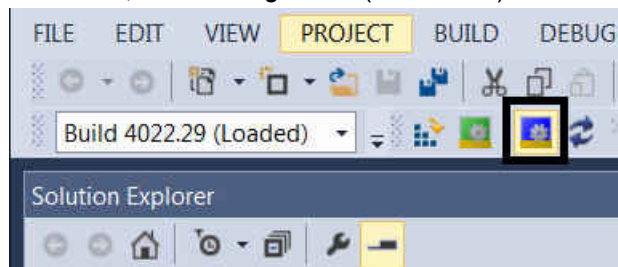


图 12-5. 使用配置模式图标重新启动 TwinCAT

6. ECAT 消息可在 CCS 调试窗口中视为变量。如果尚未完成此操作，请在 CCS “Expressions” (表达式) 窗口中右键点击鼠标，然后选择 “Import” (导入)。然后从调试目录中浏览至并选择 `fcl_f2838x_tmdxiddk_ecat_vars.txt` 文件。变量 “countMainLoop”、“dataBufferFromCM” 和 “dataBuffertoCM” 仅由 C28x 内核使用，而变量 “ipcCMToCPUDataBuffer” 和 “ipcCPUToCMDDataBuffer” 由 C28 和 CM 内核使用，因此从两个 *Debug* (调试) 空间均可查看。
7. 在 TwinCAT 中，双击所发现的 EtherCAT 框，观察 EtherCAT 从站是否以 OP 模式运行。

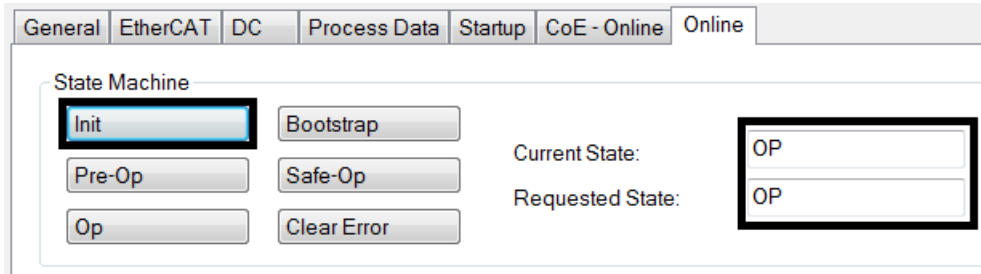


图 12-6. EtherCAT 从站处于 OP 模式

8. 在 TwinCAT 中，将资源管理器展开到 EtherCAT box (I/O --> Devices --> Device 2 (EtherCAT) --> Box 1 (F2838x CM EtherCAT Slave(SYSTEM))), 然后找到各种输出/输入映射，如图 12-7 所示。双击 Box1 以获取 F2838x CM 从站输入/输出数据窗口，如图 12-8 所示。
 - a. 选择 “Input mapping 0” 以查看驱动器给出的所有状态反馈。以下是通过 EtherCAT 监控的驱动器参数：
 - i. SpeedStatus
 - ii. PositionStatus
 - iii. TorqueStatus
 - iv. DriveStatus
 - b. 选择 “Output mapping 0” 以进行查看并输入针对驱动器的所有命令：
 - i. DriveCommand
 - ii. SpeedReference
 - iii. PositionReference

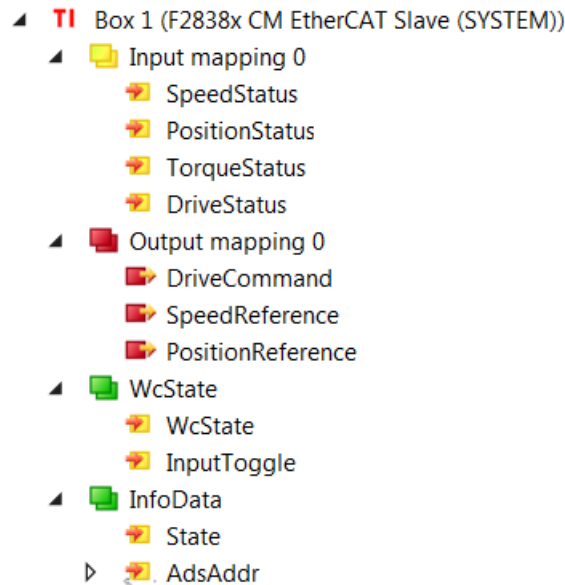


图 12-7. TwinCAT 解决方案资源管理器输入和输出

Name	Online	Type	Size	>Ad...	In/O...	User...	Linked to
SpeedStatus	22	UDINT	4.0	39.0	Input	0	
PositionStatus	333	UDINT	4.0	43.0	Input	0	
TorqueStatus	0	UDINT	4.0	47.0	Input	0	
DriveStatus	1	UINT	2.0	51.0	Input	0	
WcState	0	BIT	0.1	1522.1	Input	0	
InputToggle	1	BIT	0.1	1524.1	Input	0	
State	8	UINT	2.0	1548.0	Input	0	
.AdsAddr	10.219.50.84.3...	AMSAD...	8.0	1550.0	Input	0	
DriveCommand	1	UINT	2.0	39.0	Out...	0	
SpeedReference	22	UDINT	4.0	41.0	Out...	0	
PositionReference	333	UDINT	4.0	45.0	Out...	0	

图 12-8. F2838x CM EtherCAT Slave (SYSTEM) 输入/输出数据

9. 此构建级别是一个用于验证 EtherCAT 功能性的环回示例。因此，如果在某个输出映射变量中输入的命令数据返回到某个输入映射变量，则会确认 F2838x 上 EtherCAT 从站的功能性。环回关联如下所示：

- a. SpeedReference --> SpeedStatus
- b. PositionReference --> PositionStatus
- c. DriveCommand --> DriveStatus

例如，如果 EtherCAT 从站正常运行，则在输出映射的“SpeedReference”中输入的数据应返回并出现在输入映射的“SpeedStatus”中。

13 增量构建级别 8

假设先前的构建级别已成功完成，此构建级别将尝试采用以下方式来控制驱动器：从 TwinCAT 窗口馈入命令并通过 ECAT 监控驱动器给出的状态反馈。

13.1 在 CPU1 上运行代码以将 ECAT 分配给 CM

- 打开“`fcl_f2838x_tmdxiddk_settings_cpu1.h`”并通过将 BUILDLEVEL 设置为 FCL_LEVEL8 (#define BUILDLEVEL FCL_LEVEL8) 来选择 8 级增量构建选项。右键单击工程名称，然后单击 *Rebuild Project* (重建工程)。
- 编译完成时，单击 *Debug* (调试) 按钮，重置 CPU，重新启动，启用实时模式，然后运行。

备注

由于此构建级别尝试驱动电机，此时应恢复套件的高压直流电源。

- CPU1 应将 ECAT 的所有权授予 CM
- CPU1 将等待通过 IPC 通道从 CM 接收命令。

13.2 在 CM 上运行代码以设置 ECAT

此构建级别使用上一个构建级别中所用的相同 CM 预编译可执行文件。此处，CM 用作在 TwinCAT 和驱动控制器之间传递信息的管道。

备注

修改演示中给出的要求时，需要通过 SSC 工具生成新的从站堆栈文件，还需要对应用程序源代码进行相应的更改。必须重新构建该文件以获得新的可执行文件，从而加载到 CM 中。

您应遵循以下步骤：

1. 确保 CPU1 有效运行工程 (如节 13.1 所述)，将 EtherCAT 所有权移交给 CM
2. 除了 controlCARD 与计算机之间面向 JTAG 用途的 USB 连接外，还应在 controlCARD RJ45 端口 0 和计算机之间连接以太网电缆
3. 在 CCS 调试视图中，点击 *Run --> Load --> Load Program*，然后浏览至可执行文件 *fcl_f2838x_ecat_cm.out* (位于 `\solutions\tmdxiddk379df2838x\lssc_configuration\cm`)
4. 运行代码。这将配置 ECAT 从站控制器和 CM 侧的 IPC，以便在 CM 和 CPU1 之间进行数据传输。

现在，ECAT 从站控制器已准备好连接到 ECAT 主控制器，ECAT 主控制器是在用户的开发计算机上运行的 TwinCAT。

13.3 运行应用程序

1. 按照节 12.4 中给出的相同过程启动 TwinCAT 并扫描与其连接的 EtherCAT 器件。
2. 按照节 12.6 中给出的相同过程与驱动器交互，以馈入命令参考并恢复状态。
3. 与前一个构建级别不同，此处控制器接收的命令不会环回，而是用于实际设置驱动器的操作模式或参考。驱动器的操作状态通过 *IPC --> CM --> TwinCAT* 发送回来。
4. TwinCAT 中的输出映射设置驱动器的命令和参考，而输入映射显示驱动器的操作状态。
5. 在输出映射下，“*DriveCommand*”设置三种不同的操作模式。它可以取值为 0、1 或 2，其意义如下
 - a. 0 - 停止电机
 - b. 1 - 以速度模式运行电机 --> 外部环路为速度环路 (如 BUILDLEVEL 4)。输出映射下的“*SpeedReference*”将作为此模式的速度参考设置
 - c. 2 - 在位置模式下运行电机 --> 外部环路为位置环路 (如 BUILDLEVEL 5)。输出映射下的“*PositionReference*”将作为此模式的位置参考设置
 - d. 任何其他值 --> 默认为停止电机
6. “*SpeedReference*”和“*PositionReference*”将 1000 点尺度 (介于 0 和 +/-1000 之间) 的值作为命令。超出此范围的值将被钳制在控制端软件 (CPU1) 中，以适应驱动控制器的范围。然后由 CPU1 将其转换为标么值，这是控制软件使用的默认格式。
7. 电机运行时，电机在速度模式下报告速度反馈，并在位置模式下报告位置反馈。尝试通过编辑输出映射下的参数来更改驱动器命令，并在输入映射中查看驱动器响应。位置反馈始终以 0 到 1000 之间的正数给出。例如，如果“*PositionReference*”以 -200 给出，则反馈“*PositionStatus*”将显示接近 800 的值。就角度而言，它们代表相同的位置。

14 参考文献

- 德州仪器 (TI) : [使用 C2000 可配置逻辑块进行设计](#)
- 德州仪器 (TI) : [如何将自定义逻辑从 FPGA/CPLD 迁移到 C2000 微控制器](#)
- 德州仪器 (TI) : [DesignDRIVE IDDK 硬件参考指南](#)
- 德州仪器 (TI) : [DesignDRIVE IDDK 用户指南](#)

15 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 A (October 2019) to Revision B (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。	4
• 对 节 4 进行了更新。	6
• 添加了新的 节 5.1.4 。	11
• 添加了新的 节 7.5 。	19

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