



## 摘要

随着工业 4.0 和智能工厂的兴起，对精密、节能、安全和互联式系统提出了很高的要求。电机能耗占全球能耗的 40% 至 50%，该行业正专注于提高电机的总体生产率。恒定转速驱动器本质低效，正在被由处理器控制的可调转速驱动器大量取代。智能工业系统正在采用数字电机反馈系统。与工业以太网配合进行通信，其可为工厂和能源部门提供更高的效率。根据现场使用的自动化系统，互联驱动器使用多种工业通信协议。

AM243x/AM64x 单芯片伺服驱动器演示表明，AM243x/AM64x 器件可支持完全集成式实时伺服驱动器控制和通信路径。该路径可通过以下方式进行扩展：从接收用于扭矩/转速/位置的 EtherCAT CiA402 目标命令，到对所连接的电机执行闭环 FOC 环路控制，再到将实际值传回给 EtherCAT PLC。

本文档所讨论的工程配套资料和源代码可从以下 URL 下载：<https://www.ti.com/lit/zip/sprad03>。

## 内容

1 系统概述.....	3
2 硬件必要条件.....	3
3 硬件配置.....	4
4 HW 引脚排列、默认跳线和连接.....	4
5 原理图.....	4
6 跳线设置和说明.....	4
7 LED.....	6
8 软件架构.....	6
9 驱动和反馈时序.....	7
10 基准测试结果.....	8
10.1 电机控制 R5F 处理时间.....	8
10.2 用于进入 R5F ISR 的触发点/捕获点.....	9
11 关于演示的详细用户指南.....	10
11.1 步骤 0. 获取软件并编译.....	10
11.2 步骤 1. 开始使用硬件.....	10
11.3 步骤 2. 配置 ROQ437 EnDat2.2 编码器以缩短 EnDat 2.2 恢复时间 (只需在首次使用 ROQ437 编码器时执行一次).....	13
11.4 步骤 3. 开环 Iq 控制 (BUILDLEVEL == OPEN_LOOP_IQ_ID).....	16
11.5 步骤 4. 闭环 Iq/Id 控制 (BUILDLEVEL == CLOSED_LOOP_IQ_ID).....	21
11.6 步骤 5. 闭环速度控制 (BUILDLEVEL == CLOSED_LOOP_SPEED).....	23
11.7 步骤 6. 闭环位置控制 (BUILDLEVEL == CLOSED_LOOP_POSITION).....	25
12 使用 MCU+SDK 08.00.00.21 & CCS 10.3.1 进行编译.....	27
13 总结.....	28
14 附录 A：详细的电机控制 R5F 处理时间.....	29
15 参考文献.....	30

## 插图清单

图 1-1. 系统概述.....	3
图 3-1. 硬件配置.....	4
图 8-1. 软件架构.....	6
图 9-1. 时序图.....	7
图 9-2. 同步技术.....	8
图 10-1. 触发点和捕获点.....	9
图 11-1. 硬件设置.....	10
图 11-2. TIDEP-01015 跳线设置.....	11

图 11-3. TIDA-01619 J6 接头 1.....	12
图 11-4. TIDA-01619 J6 接头 2.....	12
图 11-5. COM 诊断输出端口.....	13
图 11-6. EnDat 2.2 诊断输出.....	14
图 11-7. 测量 EnDat CLK、TX 和 RX 引脚.....	14
图 11-8. EnDat 2.2 位置输出.....	15
图 11-9. RX 信号 (变化前).....	15
图 11-10. EnDat 2.2 恢复模式变化.....	15
图 11-11. RX 信号 (变化后).....	16
图 11-12. 开环 IQ ID - BUILDLEVEL.....	16
图 11-13. 开环 IQ ID - 运行到行.....	17
图 11-14. 开环 IQ ID - 电角.....	17
图 11-15. 开环 IQ ID - 加载符号.....	18
图 11-16. 开环 IQ ID - PWM 图 1.....	18
图 11-17. 开环 IQ ID - PWM 图 2.....	19
图 11-18. 开环 IQ ID - PWM 图 3.....	19
图 11-19. 开环 IQ ID - PWM 图 4.....	20
图 11-20. 闭环 IQ ID - BUILDLEVEL.....	21
图 11-21. 闭环 IQ ID - 运行到行.....	21
图 11-22. 闭环 IQ ID - $\alpha$ 、 $\beta$ 、IQ 和 ID 图 1.....	22
图 11-23. 闭环 IQ ID - $\alpha$ 、 $\beta$ 、IQ 和 ID 图 2.....	22
图 11-24. 闭环速度 - BUILDLEVEL.....	23
图 11-25. 闭环速度 - 运行到行.....	23
图 11-26. 闭环速度 - ID、IQ、速度图 1.....	24
图 11-27. 闭环速度 - ID、IQ、速度图 2.....	24
图 11-28. 闭环位置 - BUILDLEVEL.....	25
图 11-29. 闭环位置 - 运行到行.....	25
图 11-30. 闭环位置 - 位置和速度图 1.....	26
图 11-31. 闭环位置 - 位置和速度图 2.....	26
图 11-32. 闭环位置 - 位置控制精度.....	27

## 表格清单

表 6-1. 跳线设置和说明.....	4
表 7-1. LED.....	6
表 10-1. 编译选项.....	8
表 10-2. R5F 电机控制处理时间.....	9
表 11-1. 跳线设置表.....	11
表 14-1. 角度/位置/速度计算时间.....	29
表 14-2. 相电流扩展和转换时间.....	29
表 14-3. FOC 环路时间.....	29
表 14-4. PWM 输出时间.....	29
表 14-5. 一次测量的完整 ISR.....	29
表 14-6. 一次测量的完整 ISR (预计算的 Clarke).....	30

## 商标

Code Composer Studio™ is a trademark of Texas Instruments.

Cortex® and Arm® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

所有商标均为其各自所有者的财产。

## 1 系统概述

单芯片伺服演示表明，AM64x 和 AM243x 器件能够支持完全集成式实时伺服控制和通信路径。该路径可通过以下方式进行扩展：从接收用于扭矩/转速/位置的 EtherCAT CiA402 目标命令，到对所连接的电机执行闭环 FOC 环路控制，再到将实际值传回给 EtherCAT PLC。

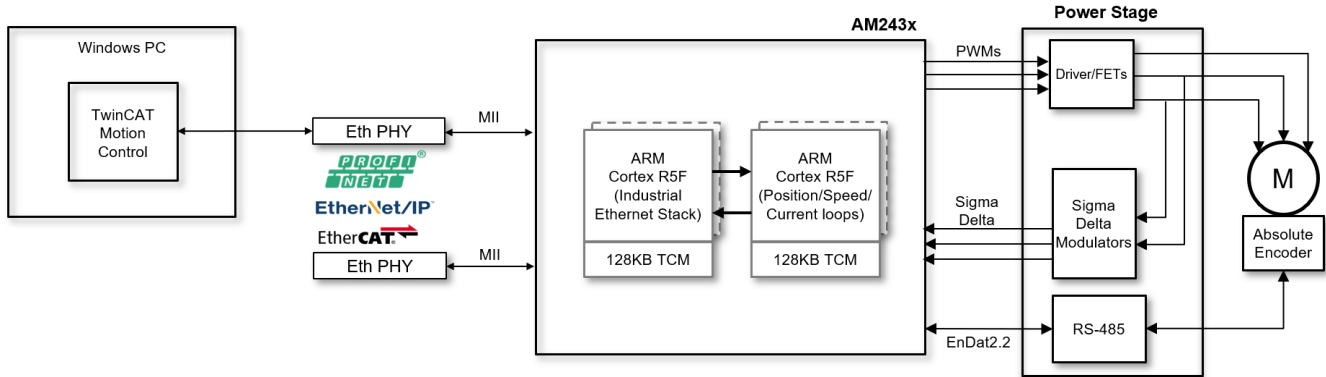


图 1-1. 系统概述

## 2 硬件必要条件

- 带有 TwinCAT 的 Windows PC
- [TMDS243GPEVM](#) 或 [TMDS64GPEVM](#) 和电源
- 3 轴适配器接口板
- TIDEP-01015 3 轴板
- [TIDA-01629](#) DRV8350 功率级板
- [TIDA-00179](#) 绝对编码器 RS485 收发器板
- BLY342D-48V-3200 Anaheim Automation 三相无刷直流电机
- ROQ-437 EnDat2.2 有线编码器
- 用于 TIDEP-01015 板的 24V 桶形插孔电源
- 用于 TIDA-01629 螺丝接线端输入的 24V 电源
- 以太网插塞式电缆
- 2 根微 USB 电缆，用于将 PC 连接到 EVM 上的仿真器和 UART 端口

### 3 硬件配置

图 3-1 显示了演示设置。

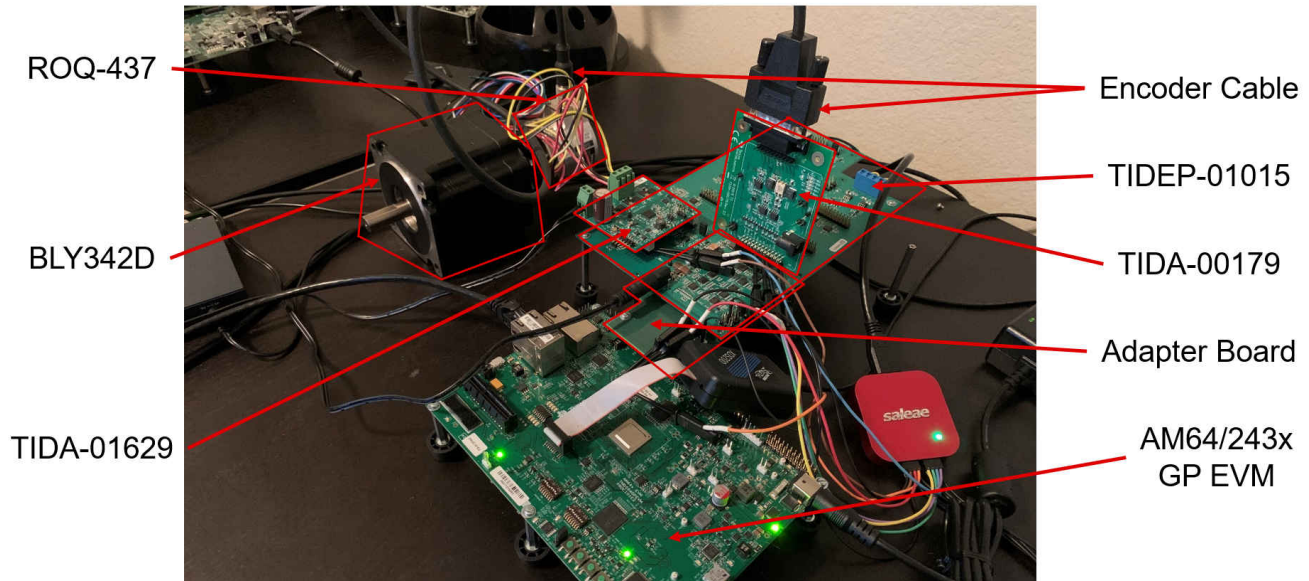


图 3-1. 硬件配置

### 4 HW 引脚排列、默认跳线和连接

- 下载 <https://www.ti.com/lit/zip/sprad03> 并将其解压缩到 <DEMO\_INSTALL\_ROOT>
- 3 轴适配器板：<DEMO\_INSTALL\_ROOT>\pinouts\3-Axis Adapter Enablement v2.pdf
- 引脚排列速查表：<DEMO\_INSTALL\_ROOT>\pinouts\AM64x\_3\_Axis\_Adapter\_Interface\_Board\_v2.xlsx

### 5 原理图

- 下载 并将其解压缩到 <DEMO\_INSTALL\_ROOT>
- 3 轴适配器板：<DEMO\_INSTALL\_ROOT>\schematics\TIDEP-01015Rev E1.1\_Sch.pdf
- TIDA-01629 功率级：<DEMO\_INSTALL\_ROOT>\schematics\TIDA-01629\_E2\_Sch.pdf
- TIDA-00179 通用编码器板：<DEMO\_INSTALL\_ROOT>\schematics\TIDA-00179 E2.pdf
- TMDS243GPEVM 或 TMDS64GPEVM：<DEMO\_INSTALL\_ROOT>\schematics\PROC101E2\_SCH.pdf
- 无源适配器板：<DEMO\_INSTALL\_ROOT>\schematics\MS\_TI\_AM64x\_EVM\_3-AXIS\_INTERFACE\_BOARD\_SCH\_REV\_E1.pdf

### 6 跳线设置和说明

表 6-1 介绍了 TIDEP-01015 ( Rev 1.1 ) 板上的跳线设置和说明。

表 6-1. 跳线设置和说明

位号	说明	设置	注意事项
J1、J3、J5	FOC 环路反馈源	(1,2) : 电流 fb ( 默认 ) 2,3 : 电压 fb 开路 : 开环	引脚编号从底部引脚开始 ( 与板上大多数其他引脚组相反 )
J2 , J4 , J6	ADC 选择	闭路 : SAR 开路 : $\Sigma$ - $\Delta$ ( 默认 )	
J7	安全转矩关闭开关输入	(1,2)(3,4)	短路的引脚 1,2 和引脚 3,4 是正常工作模式。
J8	$\Sigma$ - $\Delta$ 时钟选择	(1,2) : ECAP PWM ( 默认 ) 2,3 : 硬件延迟时钟 开路 : 无效	
J20	GND		

表 6-1. 跳线设置和说明 (continued)

位号	说明	设置	注意事项
J21	3V3_D		

## 7 LED

表 7-1 介绍了 TIDEP-01015 ( Rev 1.1 ) 板上的 LED。

表 7-1. LED

位号	说明	注意事项
D1	M1 驱动器使能	LED 亮起表示出错
D2	M2 驱动器使能	LED 亮起表示出错
D3	M3 驱动器使能	LED 亮起表示出错
D4	24V0	应始终亮起
D5	3V3	应始终亮起

## 8 软件架构

单芯片伺服驱动器演示是围绕由以下部分组成的中央实时路径来架构的：

- ICSSG1 - EtherCAT 从站控制器固件
- R5FSS1\_0 - 使用 FreeRTOS 实现 CiA402 的 EtherCAT 从站
- 在 MCU+ SDK 中提供的 IPC
- R5FSS0\_0 - 能够对电流、速度或位置进行闭环控制的闭环 FOC 控制
- EPWM - 可根据 FOC 环路的输出生成波形的增强型 PWM 外设
- ICSSG0 - PRU0 中的  $\Delta-\Sigma$  固件 (用于相电流反馈) 和 PRU1 中的 EnDat2.2 固件 (用于角度/位置/速度反馈)

此实时路径演示了组成基本框架伺服驱动器所需的组件，从接收 EtherCAT 主站中的数据，到关闭实时控制环路，以便精确控制直接连接的电机。

图 8-1 中显示了软件架构图。

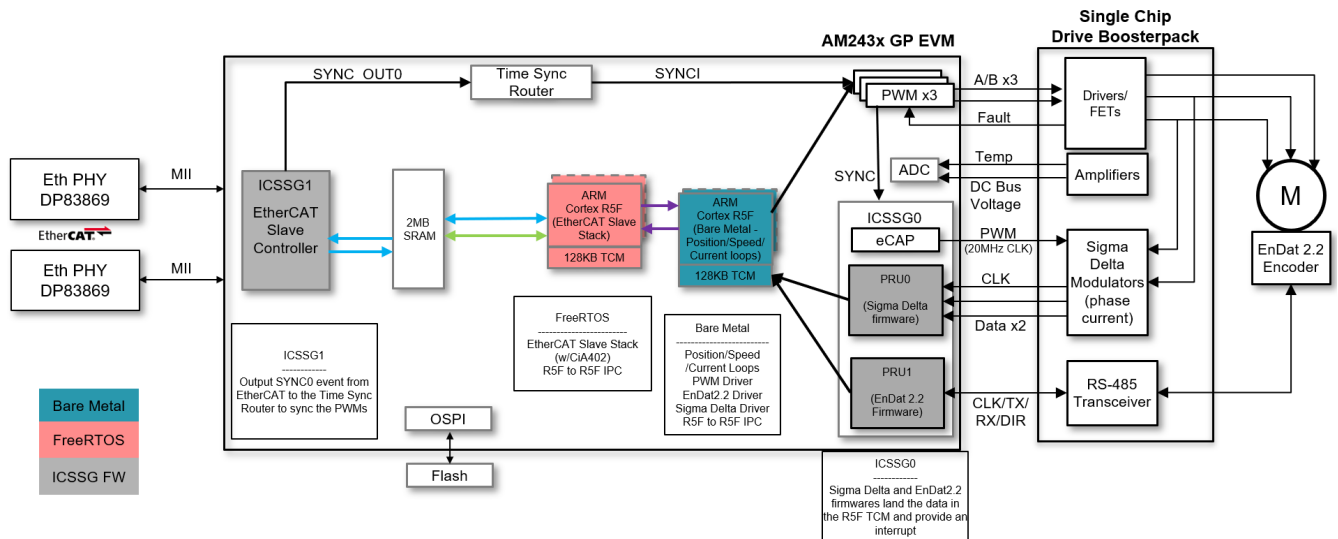


图 8-1. 软件架构

## 9 驱动和反馈时序

演示中驱动和反馈时序的精度对于正确执行 FOC 环路计算至关重要，进而对整体电机控制性能也至关重要。为了实现目标，此演示设置如下所示：

- 50KHz PWM 周期时间
  - 由于 EPWM 外设时钟为 250MHz 而实现了 12 位以上的 PWM 分辨率
    - $50\text{KHz}/4\text{ns} = 5000$ ，这大约等于 12.28 位的分辨率
- 100KHz FOC 环路更新
  - 每个 PWM 周期具有 2 个更新
  - 允许在触发点前有  $10\mu\text{s}$ ，直到可以捕获反馈，FOC 环路关闭，并且 PWM 值在影子寄存器中更新
  - 此处为实现的时序图：

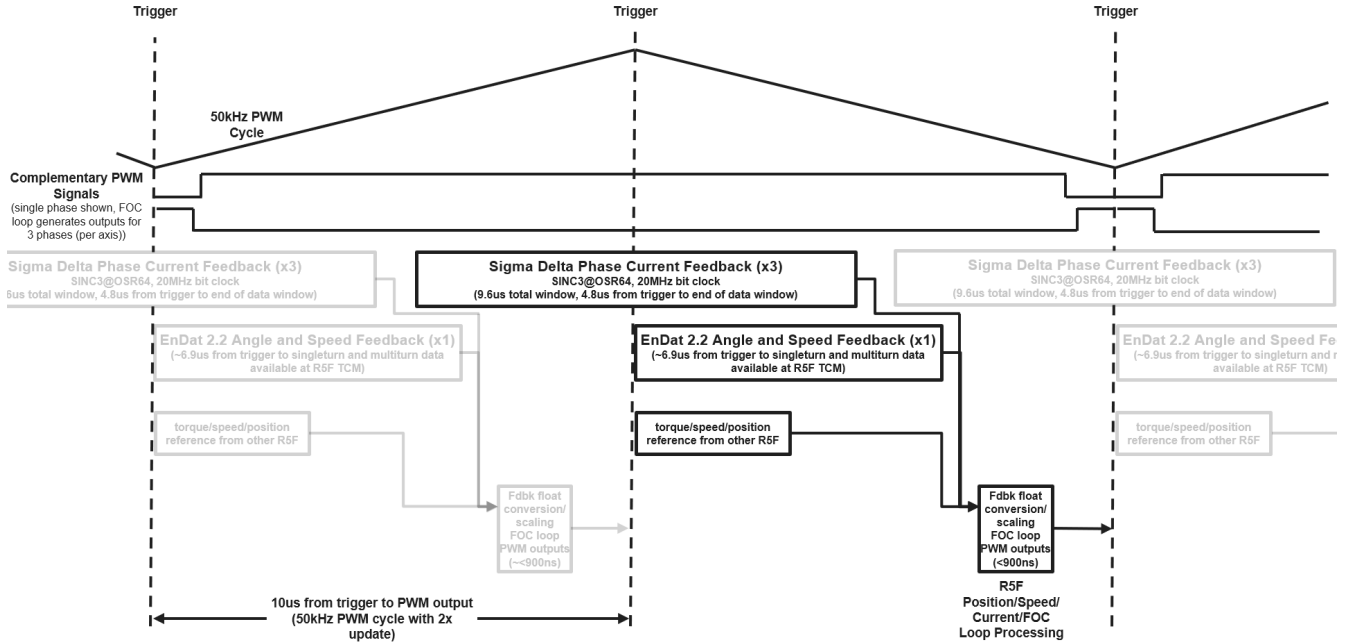


图 9-1. 时序图

- 适用于 EnDat2.2 和  $\Sigma$ - $\Delta$  反馈数据的精确触发
  - 在 PWM 外设和 ICSSG0 的 IEP 计时器之间使用同步技术，以便在 PWM 周期中的任何时刻精确放置触发器
  - EPWM SYNCO 信号可将 IEP 计时器计数器复位，并使用 CMP1 和 CMP2 值触发 EnDat 和  $\Sigma$ - $\Delta$  的起始点

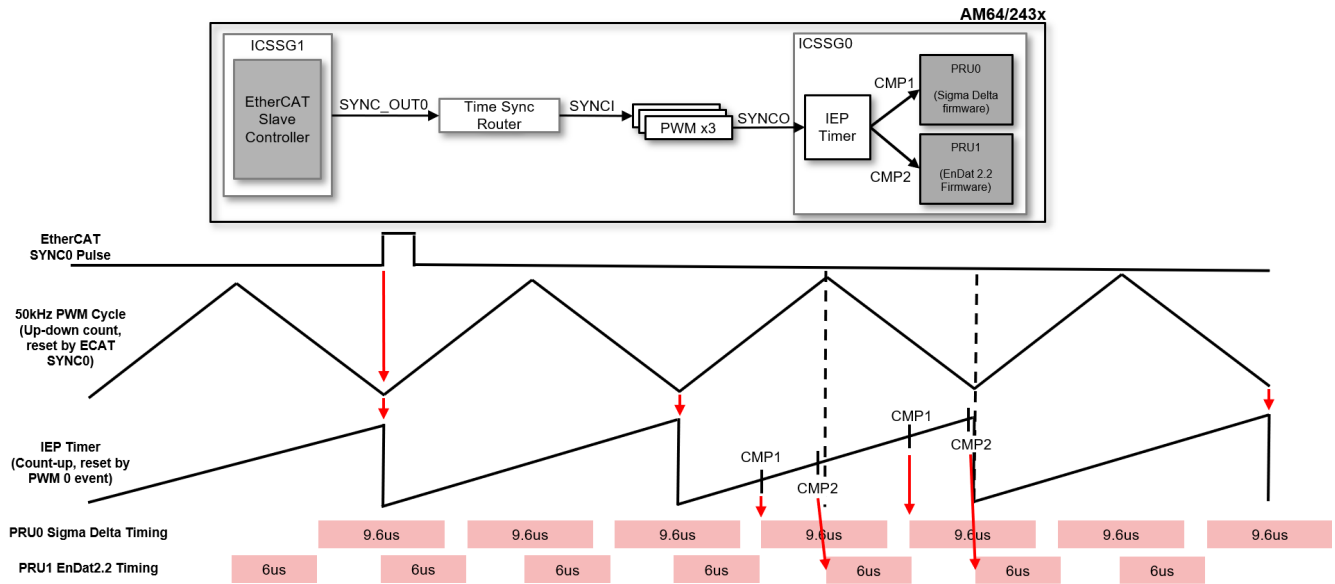


图 9-2. 同步技术

- 技术显示在下图中，结果显示在“基准测试结果”部分中

## 10 基准测试结果

### 10.1 电机控制 R5F 处理时间

测量用时大约 30 秒（大约 300 万次迭代），同时电机在环路中旋转。最大值用于所有顶层报告的结果。

GPIO 输出切换是在 180ns 时测量的。以下详细结果表的“标准化的最大值”行中的最大测量时间中减去了此值。

含 2 个更新的 50KHz PWM 周期（100KHz 处理周期时间）。

编译标志：-mcpu=cortex-r5 -mfloat-abi=hard -mfpu=vfpv3-d16 -mlittle-endian -O3

设置在“settings.h”中使用：

表 10-1. 编译选项

BUILDLEVEL	OPEN_LOOP_IQ_ID	CLOSED_LOOP_IQ_ID	CLOSED_LOOP_SPEED	CLOSED_LOOP_POSITION	CLOSED_LOOP_CIA402
PID_TUNE_LEVEL	NO_TUNING	NO_TUNING	NO_TUNING	NO_TUNING	NO_TUNING
DEBUG_LEVEL	DEBUG_BUFFERS_OFF	DEBUG_BUFFERS_OFF	DEBUG_BUFFERS_OFF	DEBUG_BUFFERS_OFF	DEBUG_BUFFERS_OFF



表 10-2. R5F 电机控制处理时间

功能	电机控制 R5F 处理时间					单位
	开环 Iq/Id	闭环 Iq/Id	闭环速度	闭环位置	闭环 CiA402	
角度/位置/速度计算	132	128	156	156		ns
相电流扩展, 转换为 FP	20	180	316	236		ns
FOC 环路和 PI 控制器	180	252	340	404		ns
写至 PWM 输出并为下一周期保存状态	84	84	84	84		ns
总计 (汇总上面所添加的值)	416	644	896	880		ns
总计 (全部一次测量) (PRECOMPUTE_LEVEL = NO_PRECOMPUTE)	388	644	884	892	860	ns
总计 (全部一次测量) ADC 值和早期运行的 Clarke (PRECOMPUTE_LEVEL = PRECOMPUTE_CLARKE)	396	484	740	772		ns

有关详细的电机控制 R5F 处理时间, 请参阅节 14

## 10.2 用于进入 R5F ISR 的触发点/捕获点

实验显示, EnDat 位置锁存器在距所需触发点 10ns 以内时显示在逻辑分析仪上。观察值距所需捕获点 0ns 到 8ns。图 10-1 显示了触发点/捕获点:

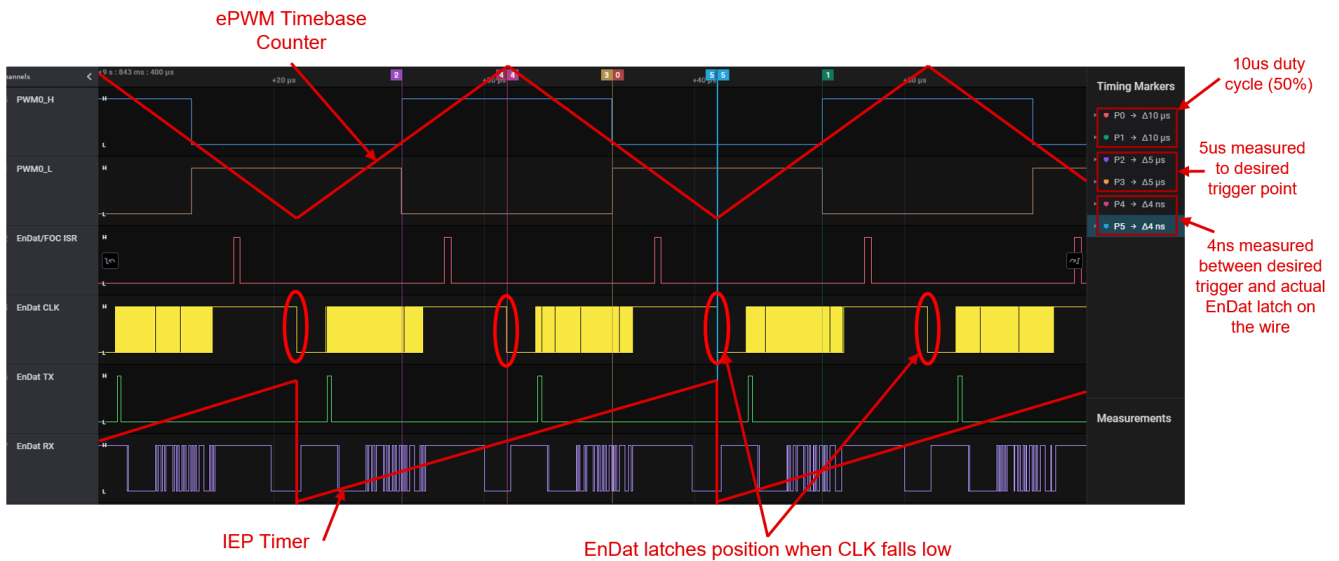


图 10-1. 触发点和捕获点

## 11 关于演示的详细用户指南

### 11.1 步骤 0. 获取软件并编译

- 有关详细信息，请参阅以下链接：[节 12](#)

### 11.2 步骤 1. 开始使用硬件

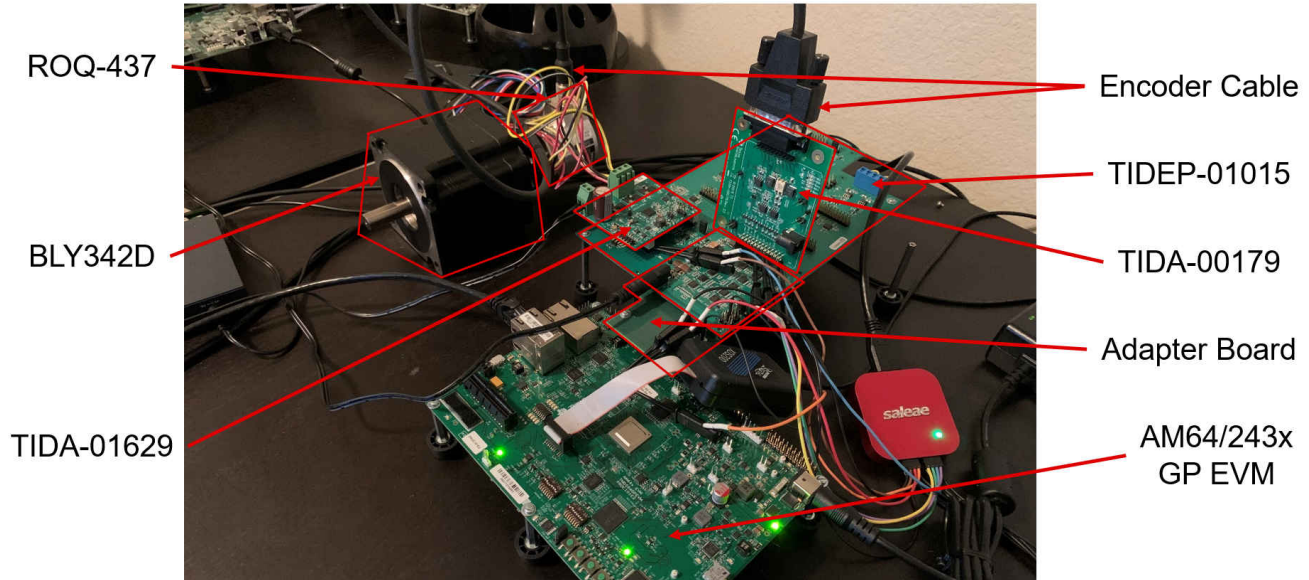


图 11-1. 硬件设置

- 将 5 个 PCB 插接在一起，具体操作请参考上面的图片和下面的说明
  - 如上图所示，适配器板、TIDEP-01015 板和 TMDS243GPEVM 只能通过一种方式连接在一起。将这三个板插接在一起，并根据需要添加所有支架
  - TIDA-01629 功率级板应插入离 TMDS243GPEVM 最近的第一个 BoosterPack 插槽中，并且螺丝接线端应面向板的外部。这第一个插槽对应于 PWM0-2 以及 SD 通道 0-2
    - 不应在此板上填充跳线。
  - TIDA-00179 编码器收发器板应通过直角连接器和离 TMDS243GPEVM 最近的 DB-15 连接器插入离 TMDS243GPEVM 最近的第一个编码器插槽中。第一个插槽对应于 EnDat 通道 0
    - 不应在此板上填充跳线
- 根据下面的红色矩形中所示及表格中所述，配置 TIDEP-01015 板上的跳线。另请注意，J7 中的螺丝接线端连接器需要短路，如[图 11-2](#)中所示及表格中所述（此连接器启用 STO 类型功能，但当前被旁路）。

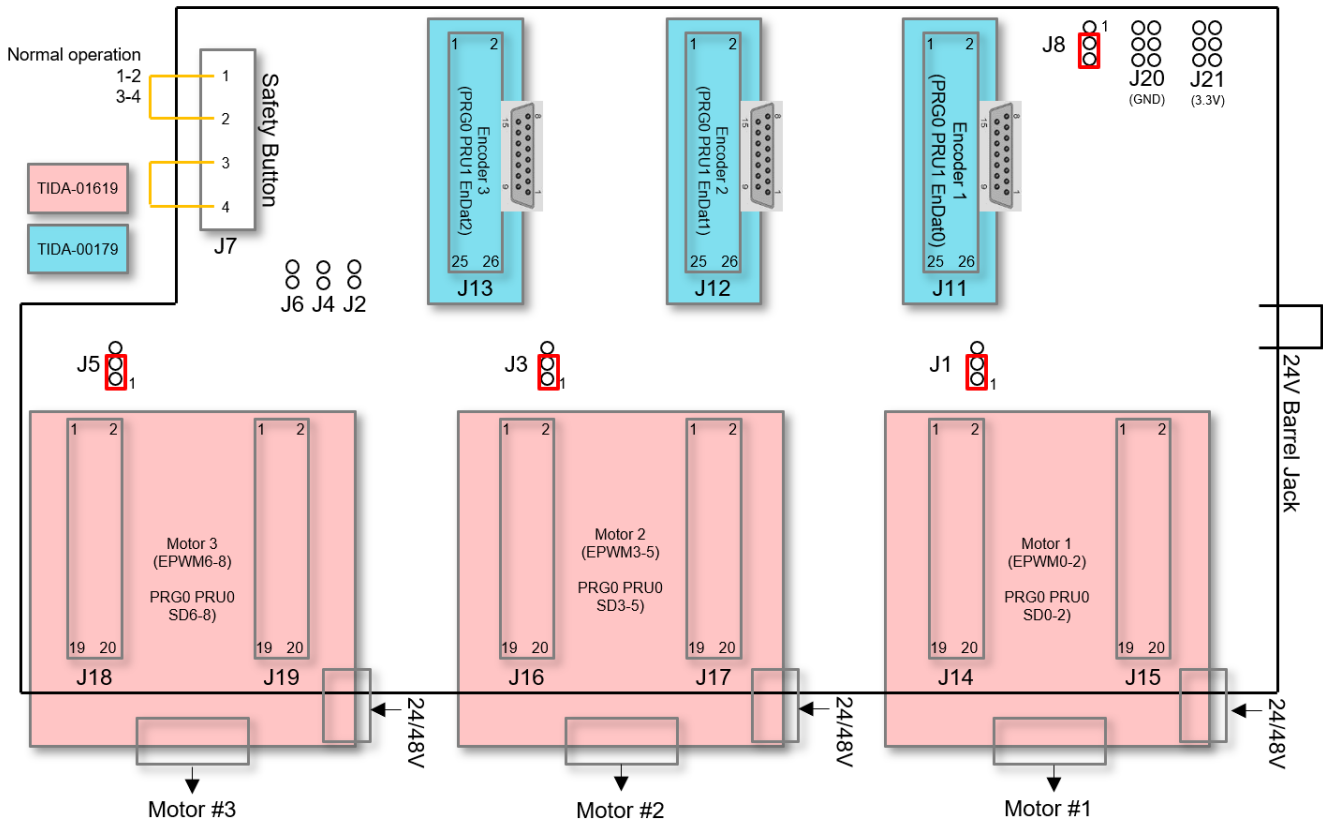


图 11-2. TIDEP-01015 跳线设置

• 跳线设置表

表 11-1. 跳线设置表

位号	说明	设置	注意事项
J1、J3、J5	FOC 环路反馈源	(1,2) : 电流 fb (默认) 2,3 : 电压 fb 开路 : 开环	引脚编号从底部引脚开始 (与板上大多数其他引脚组相反)
J2、J4、J6	ADC 选择	闭路 : SAR 开路 : $\Sigma$ - $\Delta$ (默认)	
J7	安全转矩关闭开关输入	(1,2)(3,4)	短路的引脚 1,2 和引脚 3,4 是正常工作模式。
J8	$\Sigma$ - $\Delta$ 时钟选择	(1,2) : ECAP PWM (默认) 2,3 : 硬件延迟时钟 开路 : 无效	
J20	GND		
J21	3V3_D		

- 将电机电线连接到 J6 中的 TIDA-01629 功率级螺丝接线端。一定要注意，电机相电线为 6 根较粗的电线。

备注

还有 5 根较细的单色电线，它们用于应保持未连接状态的增量编码器。

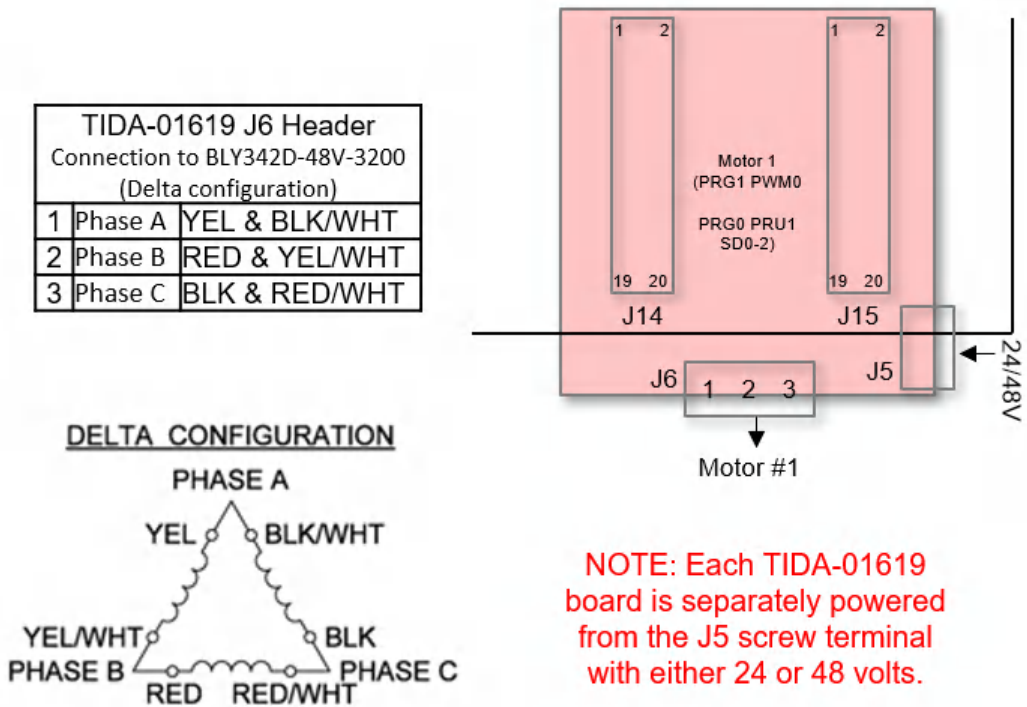


图 11-3. TIDA-01619 J6 接头 1

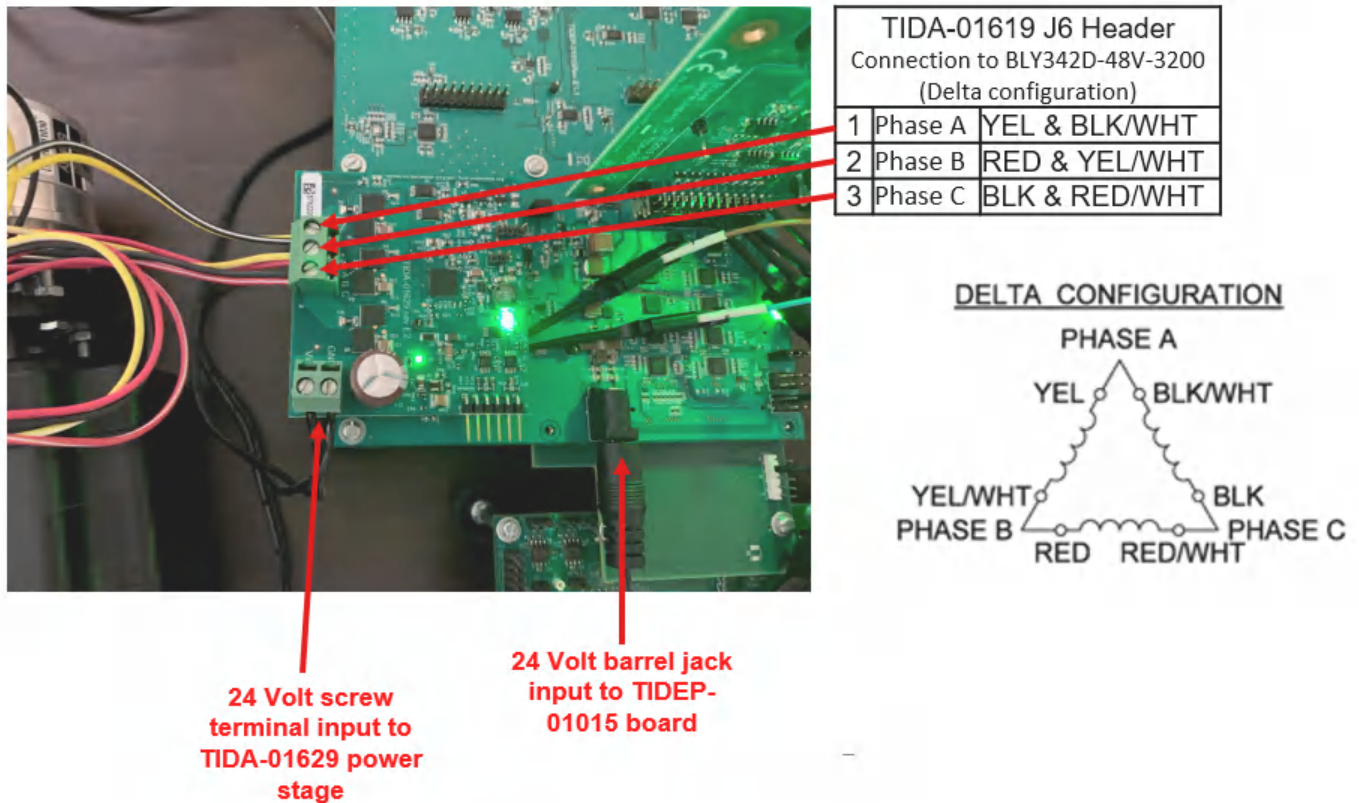


图 11-4. TIDA-01619 J6 接头 2

- 将编码器电缆从 ROQ-437 EnDat2.2 编码器连接到 TIDA-00179 编码器收发器板
- 将 24V 连接到 TIDA-01629 板的螺丝接线端以及 TIDEP-01015 板上的桶形插孔连接器 ( 2 个步骤如上图所示 )。请勿将电源连接到 TIDA-00179 板的桶形插孔，将通过 TIDEP-01015 板为该板供电

### 11.3 步骤 2.配置 ROQ437 EnDat2.2 编码器以缩短 EnDat 2.2 恢复时间 ( 只需在首次使用 ROQ437 编码器时执行一次 )

在 ROQ437 中，两个位置锁存器之间的 EnDat2.1 恢复时间默认较短。EnDat2.2 可实现较短的恢复时间，但我们需要先在 ROQ437 中设置一次非易失性位，然后再运行演示的剩余部分以缩短恢复时间。这是目前的权变措施，可在以后并入演示代码中。

Heidenhain 文档中的说明 ( [https://www.heidenhain.us/wp-content/uploads/Bidirectional\\_Interface\\_for\\_Position\\_Encoders-1.pdf](https://www.heidenhain.us/wp-content/uploads/Bidirectional_Interface_for_Position_Encoders-1.pdf) )：扩展的 EnDat 接口 2.2 版在通信、命令集和时间条件方面与之前的 2.1 版兼容，但也具有一些显著的优势。例如，有了它，不必发送单独的请求，即可传输含位置值的附加信息。接口协议已扩展，并且时间条件已优化，如下所示：提高了时钟频率 ( 时钟 ) ( 16MHz )，优化了计算时间 ( 位置值采集时间在 5 $\mu$ s 内 )，极大地缩短了死区时间 ( 恢复时间 ) ( 1.25 $\mu$ s 至 3.75 $\mu$ s )，扩展了电源电压范围 ( 3.6V 至 5.25V，或者，在编码器中为 3.6V 至 14V )

- 插入所有硬件以将 EnDat2.2 编码器连接到离 TMD5243GPEVM 最近的插槽
- 将位于 “%SDK\_INSTALL\_DIR%\examples\motor\_control\endat\_diagnostic” 的 MCU+ SDK 中的 “endat\_diagnostic” 示例导入到 CCS 中
- 编译该示例并将其加载到 MAIN\_Cortex®\_R5\_0\_0 中
- 在 Code Composer Studio™ ( CCS ) 中设置端子以查看诊断输出 ( 如下所示，我的端口为 COM6，但您的端口可能不同 )：

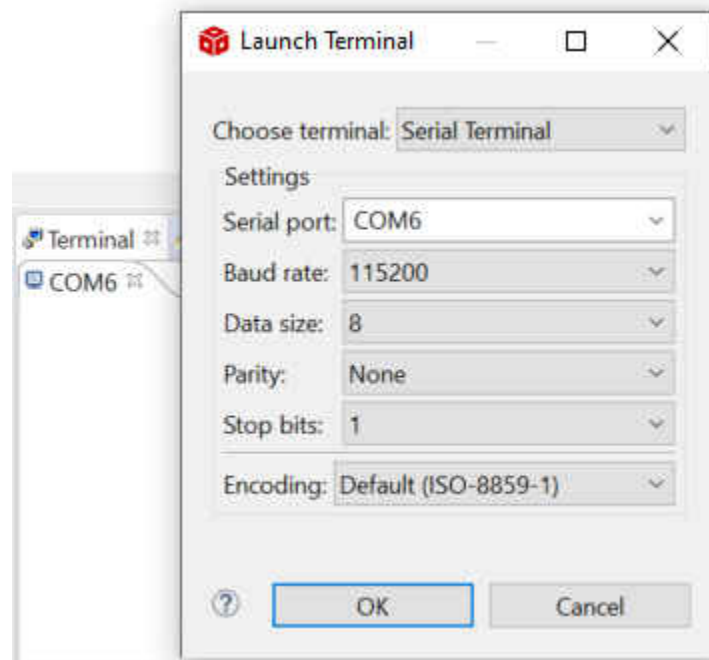


图 11-5. COM 诊断输出端口

- 点击 “Resume (F8)” 以运行诊断
- 针对多通道配置选择 “y”，针对所选通道 0 选择 “y”，并针对通道 1 和通道 2 选择 “n”。

```

COM6
EnDat firmware : 0.2.0 (release)

Multi channel configuration ? [y/N]: y
selected multi channel configuration
select channels to be used in multi channel
select channel 0 [Y/n]: y
select channel 1 [Y/n]: n
select channel 2 [Y/n]: n
channels 0 selected

CHANNEL 0
EnDat 2.2 rotary encoder ID: 1042255 01 SN: 68423189
Position: 37 bits (singleturn: 25, multiturn: 12) [resolution: 33554432 M/rev]
Propagation delay: 335ns
    
```

图 11-6. EnDat 2.2 诊断输出

- 完成这些选择之后，诊断工具应与编码器通信并返回 ID、SN 以及有关分辨率和传播延迟的信息
- 将逻辑分析仪连接到 EnDat CLK、TX 和 RX 引脚（如下所示，可通过 TIDA-00179 板的直角连接器访问这些引脚）

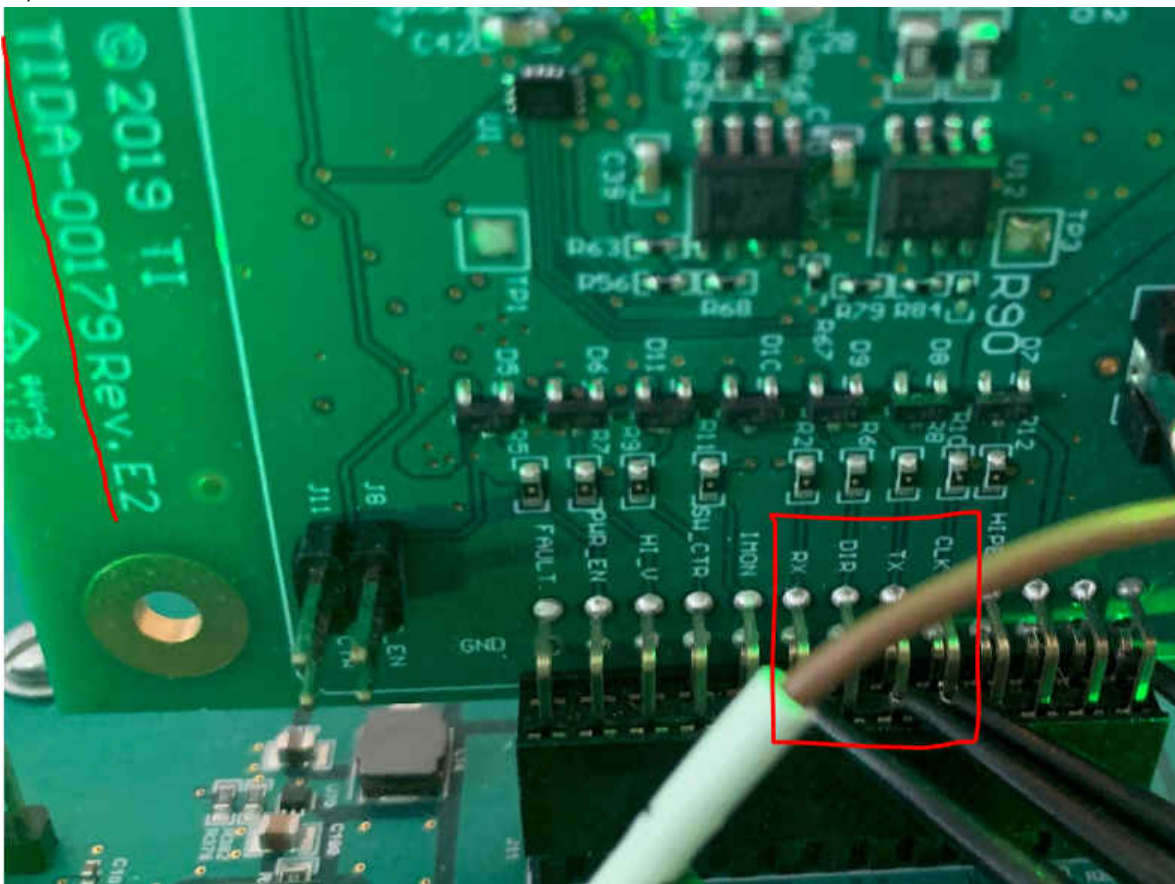


图 11-7. 测量 EnDat CLK、TX 和 RX 引脚

- 在诊断工具中选择选项“107”，并输入“8000”（以 Hz 为单位）作为频率值。这会导致程序持续向编码器请求数据，如图 11-8 中所示。

```

enter value: 107
enter frequency in Hz: 8000

press enter to stop the position display

CHANNEL 0

position,      revolution, crc errors, f1, f2
167.691802978516,      82,      1, 0, 1
167.691818237305,      82,      1, 0, 1
167.691802978516,      82,      1, 0, 1
167.691802978516,      82,      1, 0, 1
167.691787719727,      82,      1, 0, 1
167.691787719727,      82,      1, 0, 1
    
```

图 11-8. EnDat 2.2 位置输出

- 在仿真位置环路运行时运行逻辑分析仪，并查看 RX 信号中的恢复时间（在 CLK 停止后，RX 信号保持高电平的时间）：



图 11-9. RX 信号 (变化前)

- 要设置位以启用 EnDat2.2 恢复模式，请选择诊断选项 10，参数地址 3，及参数值 1

```

enter value: 10
enter parameter address (hex value): 3
enter parameter (hex value): 1
    
```

图 11-10. EnDat 2.2 恢复模式变化

- 然后，选择诊断选项“5”将编码器复位，以使设置生效
- 再次运行选项 107 并将频率设置为 8000，以确保设置保存下来并且恢复时间少于 3.75μs：

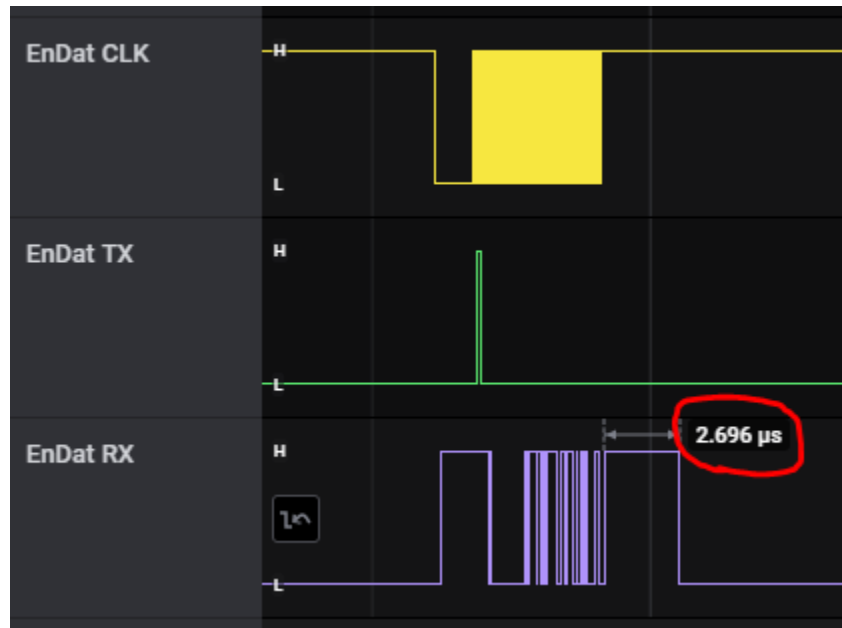


图 11-11. RX 信号 (变化后)

- 如果恢复时间仍为 20μs 左右，则您可能需要再试一次之前的步骤（选项 10，地址 3，值 1，然后使用选项 5 进行复位）

此时，应将您的编码器配置为在 EnDat2.2 恢复模式下运行，以便在演示的剩余部分中全程使用所需的严格计时。

### 11.4 步骤 3. 开环 Iq 控制 ( BUILDLEVEL == OPEN\_LOOP\_IQ\_ID )

此编译级别可验证  $\Sigma$ - $\Delta$  是否在运行并且它的直流失调电压是否设置正确，EnDat 机械  $\theta$  失调电压根据 EnDat 反馈计算，并且 PWM 路径正常运行以使电机在开环控制模式下旋转。

1. 转至 settings.h 文件，并更改定义以匹配以下内容（开环控制，并启用调试缓冲区）：

```
#define BUILDLEVEL OPEN_LOOP_IQ_ID
#define PID_TUNE_LEVEL NO_TUNING
#define DEBUG_LEVEL DEBUG_BUFFERS_ON
#define DEBUG_WRAP_TYPE DEBUG_BUFFER_SINGLE
#define PRECOMPUTE_LEVEL NO_PRECOMPUTE

/* The number of ISR cycles to use the same target from
#define CYCLES_PER_TARGET 6000
#define TARGET_BUFF_SIZE CYCLES_PER_TARGET * 8

/* Iq testing value for open loop */
#define IQ_TESTING 0.2
/* Max speed change request allowed between updates */
#define MAX_SPD_CHANGE 0.12
/* Max position change request allowed between updates
#define MAX_POS_CHANGE 0.03 /* 500RPMs
```

图 11-12. 开环 IQ ID - BUILDLEVEL

2. 在调试模式下编译项目，然后将其加载到 MAIN\_Cortex\_R5\_0\_0 中。



3. 加载完项目后，打开“single\_chip\_servo.c”文件并找到以下行，然后点击右键并选择“Run to Line”。

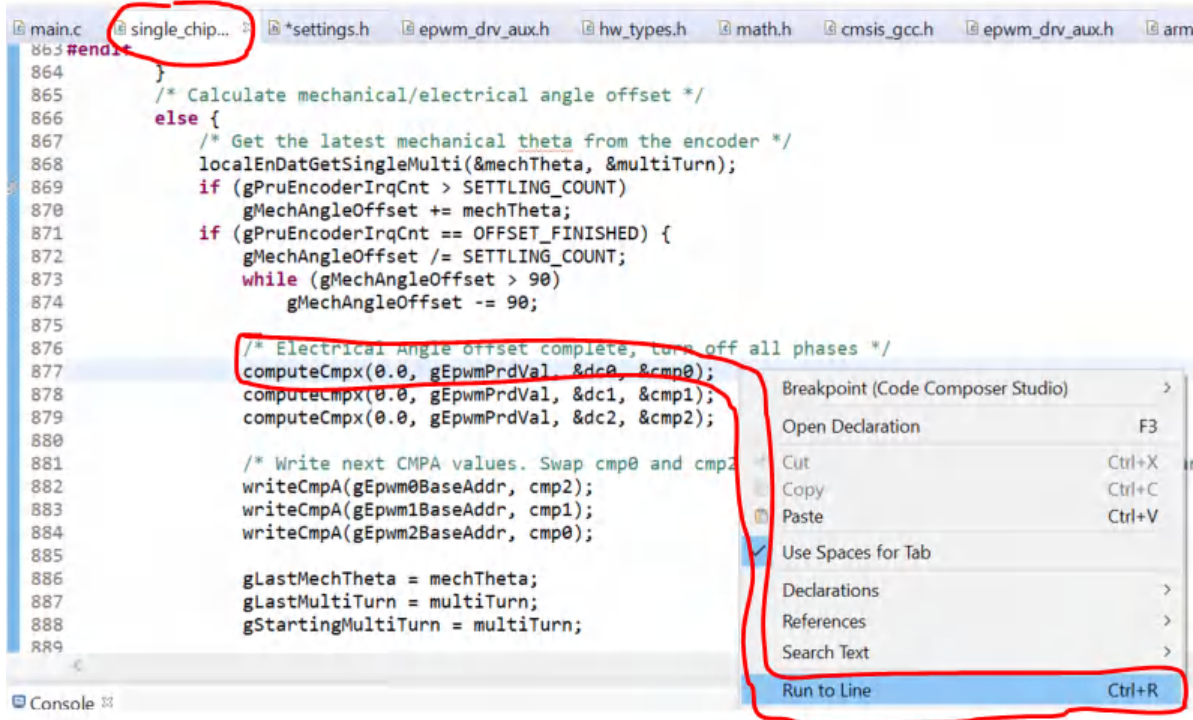


图 11-13. 开环 IQ ID - 运行到行

4. 在内核达到该行并停止之前，电机轴应锁定到位（电角为 0）。
5. 一旦内核达到该行，程序已将所有相设置为 0,0,0，对所有三个相电流读取 SETTLING\_COUNT 次（默认为 8192）并据其计算直流失调电压。一旦发现  $\Sigma$ - $\Delta$  偏移量，程序会通过将所有相设置为 1,0,0 来将轴以 0 电角锁定。一旦锁定，程序便会为 EnDat2.2 机械角反馈值读取 SETTLING\_COUNT 次（8192），并计算机械  $\theta$  和电  $\theta$  之间的偏移量。
6. 要确保反馈路径（ $\Sigma$ - $\Delta$  和 EnDat2.2）正常运行，请打开“Expressions”窗口并观察“gSddfChOffsets”和“gMechAngleOffset”。
  - a.  $\Sigma$ - $\Delta$  偏移量应介于 -1000 至 1000 之间。如果偏移量不在此范围内，或卡在中标度或满量程值（131072 或 272144）处，则需要检查您的跳线设置是否选择  $\Sigma$ - $\Delta$ （J2、J4、J6 保持开路）和全部 3 个相电流（J1、J3 和 J5 连接引脚 1 和 2（离功率级最近的两个引脚））。如果所有跳线设置均正确无误，但您的一个通道仍卡在此范围外，则可能存在硬件问题（迄今已在一个板上的单通道上观察到）。
  - b. gMechAngleOffset 的值应介于 0 和 90 之间，并表示编码器机械角和电机绕组内电角之间的偏移量。此值应在演示的多次运行之间保持相对稳定。如果您注意到此值在不同运行中是不同的，则电机和编码器轴之间的耦合器可能已松散，而这将导致打滑。如果运行演示时耦合器中打滑，则为 Park 变换提供的电角将是错误的，并且电机不会平滑旋转（在所测试的第一个电机上观察到了这一点）。

gSddfChOffsets	float[3]	[589.728882, -271.831177, -292.681152]
[0]	float	589.728882
[1]	float	-271.831177
[2]	float	-292.681152
gMechAngleOffset	float	50.8770294
Add new expression		

图 11-14. 开环 IQ ID - 电角

7. 一旦在上面确认了  $\Sigma$ - $\Delta$  和 EnDat2.2 路径，则您可以恢复内核，使其开始运行开环控制，电机应该开始旋转。
8. 在电机旋转时，您可以查看调试输出：
  - a. 连接到 OTHER R5F 内核“R5\_1\_0”并加载 SYMBOLS ONLY：

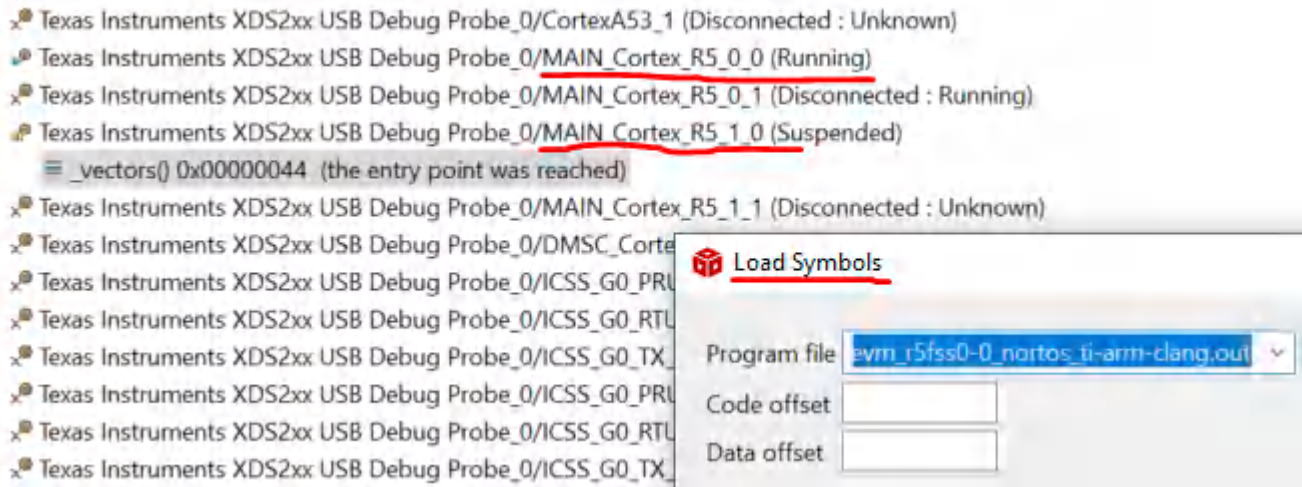


图 11-15. 开环 IQ ID - 加载符号

b. 导入三个 CCS 图以显示在开环控制模式下使用的相 A、B 和 C PWM 输出值。

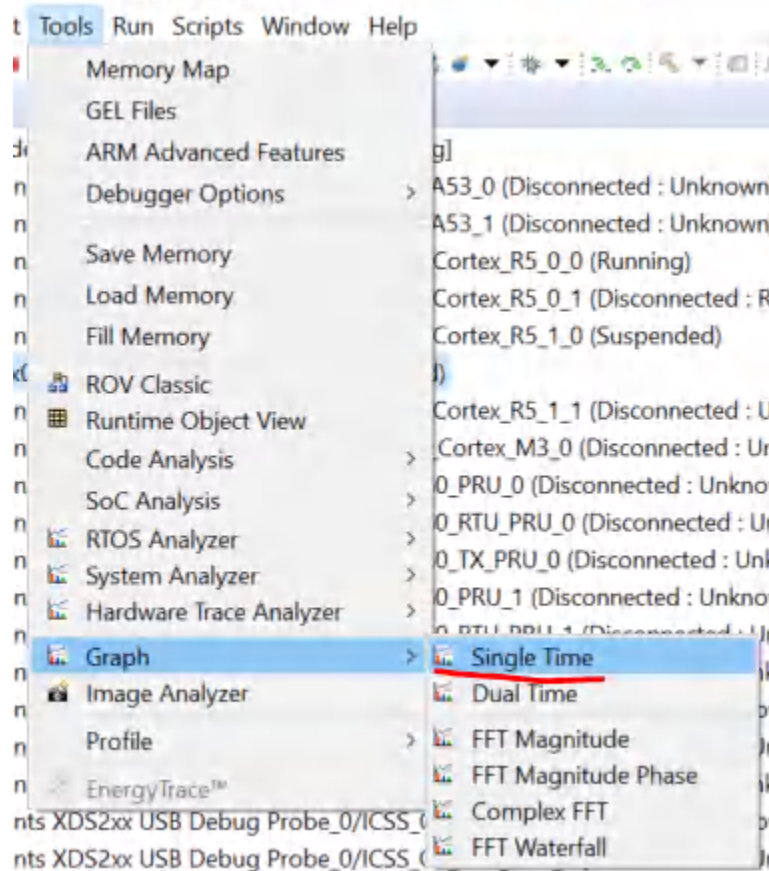


图 11-16. 开环 IQ ID - PWM 图 1

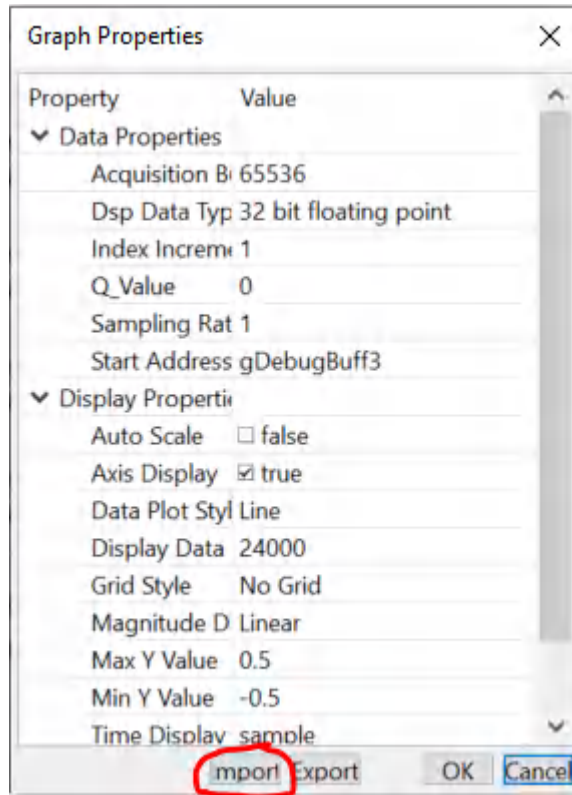


图 11-17. 开环 IQ ID - PWM 图 2



图 11-18. 开环 IQ ID - PWM 图 3

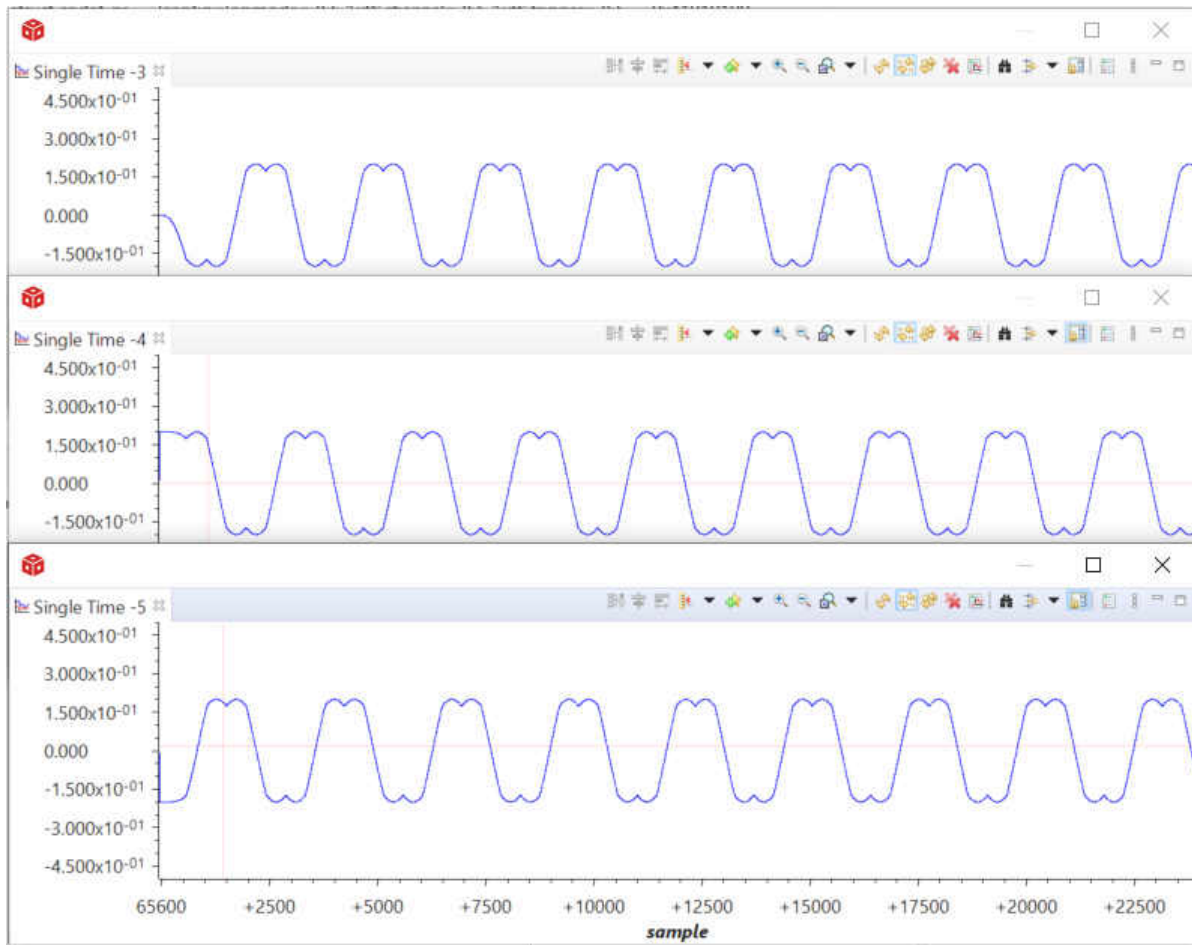


图 11-19. 开环 IQ ID - PWM 图 4

- c. 这会显示处于开环控制模式下时输出的空间矢量调制 PWM 值。您可以看出，三个相是上图中的每个偏移量。如果您想要在 Matlab 或 Excel 中将这此数据一起操作或为其绘图，则可以通过 CCS 内存浏览器转储这此数据。
- d. 开环控制演示到这里就结束了，现在已完成对反馈路径和输出路径的验证（假设上面显示的您的偏移量值看起来正常并且您的电机在旋转）。
- e. 要在开环控制模式下停止电机旋转，您可以关闭 EVM。

## 11.5 步骤 4. 闭环 Iq/I<sub>d</sub> 控制 ( BUILDLEVEL == CLOSED\_LOOP\_IQ\_ID )

此编译级别是使用相电流和角度/位置反馈关闭环路的第一个级别。

1. 转至 settings.h 文件，并更改定义以匹配以下内容（闭环电流控制，启用调试缓冲区，每个设定点具有 8000 个周期）：

```
#define BUILDLEVEL CLOSED_LOOP_IQ_ID
#define PID_TUNE_LEVEL NO_TUNING
#define DEBUG_LEVEL DEBUG_BUFFERS_ON
#define DEBUG_WRAP_TYPE DEBUG_BUFFER_SINGLE
#define PRECOMPUTE_LEVEL NO_PRECOMPUTE

/* The number of ISR cycles to use the same target from */
#define CYCLES_PER_TARGET 8000
#define TARGET_BUFF_SIZE CYCLES_PER_TARGET * 8

/* Iq testing value for open loop */
#define IQ_TESTING 0.2
/* Max speed change request allowed between updates */
#define MAX_SPD_CHANGE 0.12
/* Max position change request allowed between updates */
#define MAX_POS_CHANGE 0.03 /* 500RPMs
```

图 11-20. 闭环 IQ ID - BUILDLEVEL

2. 如果需要，请转至 single\_chip\_servo.c 文件并修改 Iq 设定点数组。默认值如下所示。

```
volatile float gIqArray[8] = {0,0.5,1.0,1.5,0,-.5,-1,0};
```

3. 在调试模式下编译项目，然后将其加载到 MAIN\_Cortex\_R5\_0\_0 中。
4. 加载完项目后，打开“single\_chip\_servo.c”文件并找到以下行，然后点击右键并选择“Run to Line”。

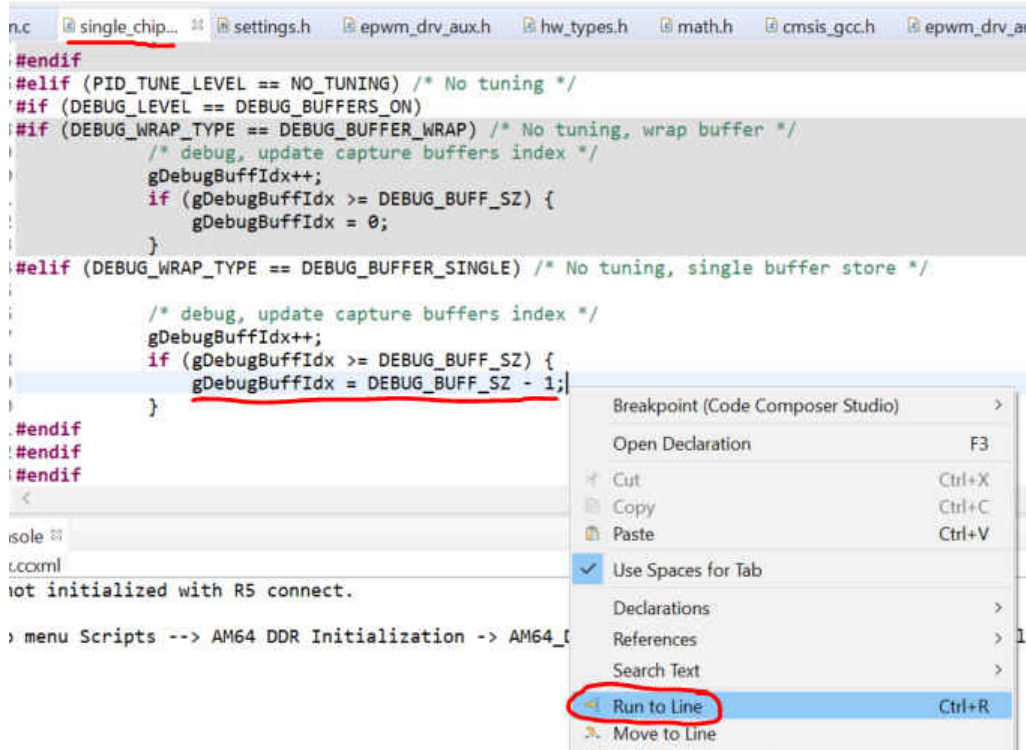


图 11-21. 闭环 IQ ID - 运行到行

5. 运行到此行将使控制代码可以运行八个不同的设定点并填充调试缓冲区，以便可以查看此图。

6. 通过导入以下各个图来查看输出：

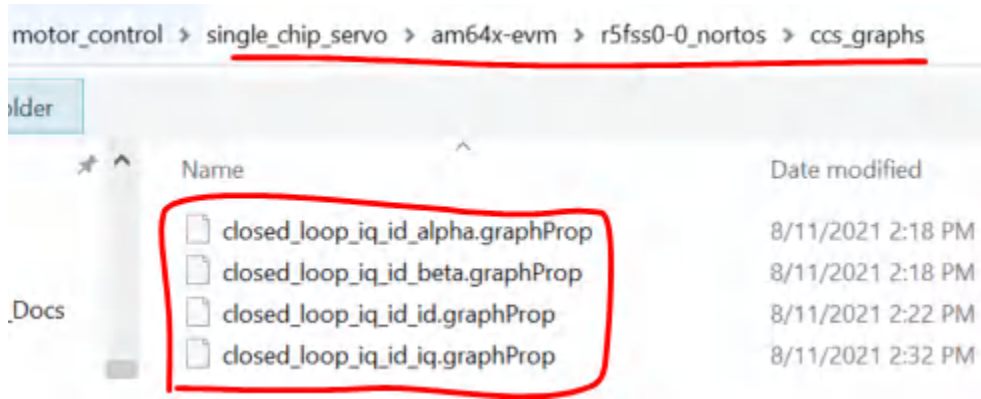


图 11-22. 闭环 IQ ID -  $\alpha$ 、 $\beta$ 、IQ 和 ID 图 1

7. 以下所示为示例输出（自上而下为  $\alpha$ 、 $\beta$ 、Iq 和 Id 图）。请注意 Iq 图与设定点数组中上述设定点的匹配程度，根据请求，该 Id 被控制为接近于 0：

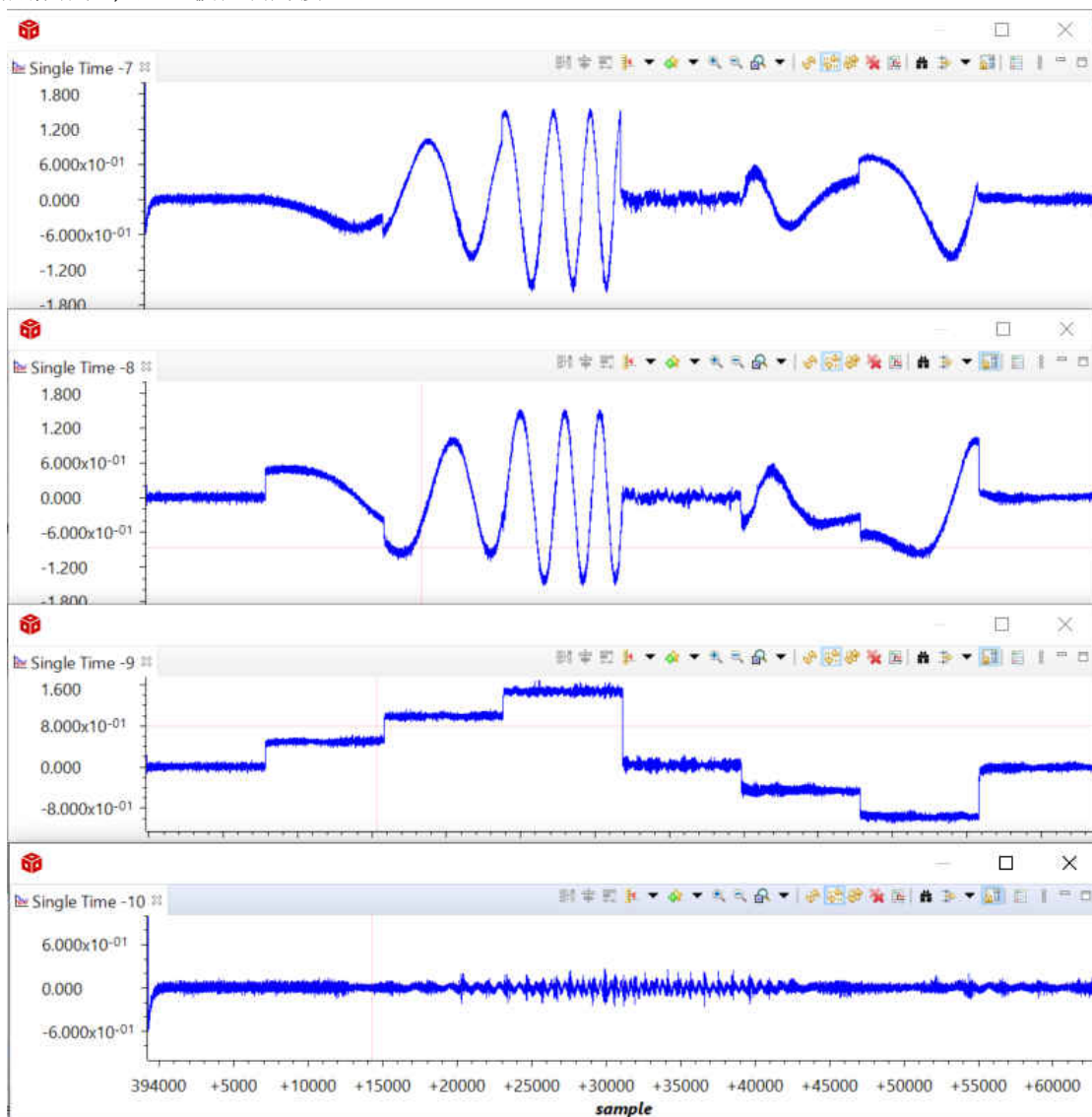


图 11-23. 闭环 IQ ID -  $\alpha$ 、 $\beta$ 、IQ 和 ID 图 2

## 11.6 步骤 5. 闭环速度控制 ( BUILDLEVEL == CLOSED\_LOOP\_SPEED )

除电流环路以外，此编译级别还可以关闭速度环路。

1. 转至 settings.h 文件，并更改定义以匹配以下内容（闭环速度控制，启用调试缓冲区，每个设定点具有 8000 个周期）：

```
#define BUILDLEVEL          CLOSED_LOOP_SPEED
#define PID_TUNE_LEVEL     NO_TUNING
#define DEBUG_LEVEL        DEBUG_BUFFERS_ON
#define DEBUG_WRAP_TYPE    DEBUG_BUFFER_SINGLE
#define PRECOMPUTE_LEVEL   NO_PRECOMPUTE

/* The number of ISR cycles to use the same target froi
#define CYCLES_PER_TARGET   8000
#define TARGET_BUFF_SIZE   CYCLES_PER_TARGET * 8

/* Iq testing value for open loop */
#define IQ_TESTING          0.2
/* Max speed change request allowed between updates */
#define MAX_SPD_CHANGE      0.12
/* Max position change request allowed between updates
#define MAX_POS_CHANGE      0.03 /* 500RPMs
```

图 11-24. 闭环速度 - BUILDLEVEL

2. 如果需要，请转至 single\_chip\_servo.c 文件并修改速度设定点数组。默认值如下所示。

```
volatile float gSpdArray[8] = {0,500,750,-500,-750,0,0,0};
```

3. 在调试模式下编译项目，然后将其加载到 MAIN\_Cortex\_R5\_0\_0 中。
4. 加载完项目后，打开“single\_chip\_servo.c”文件并找到以下行，然后点击右键并选择“Run to Line”。

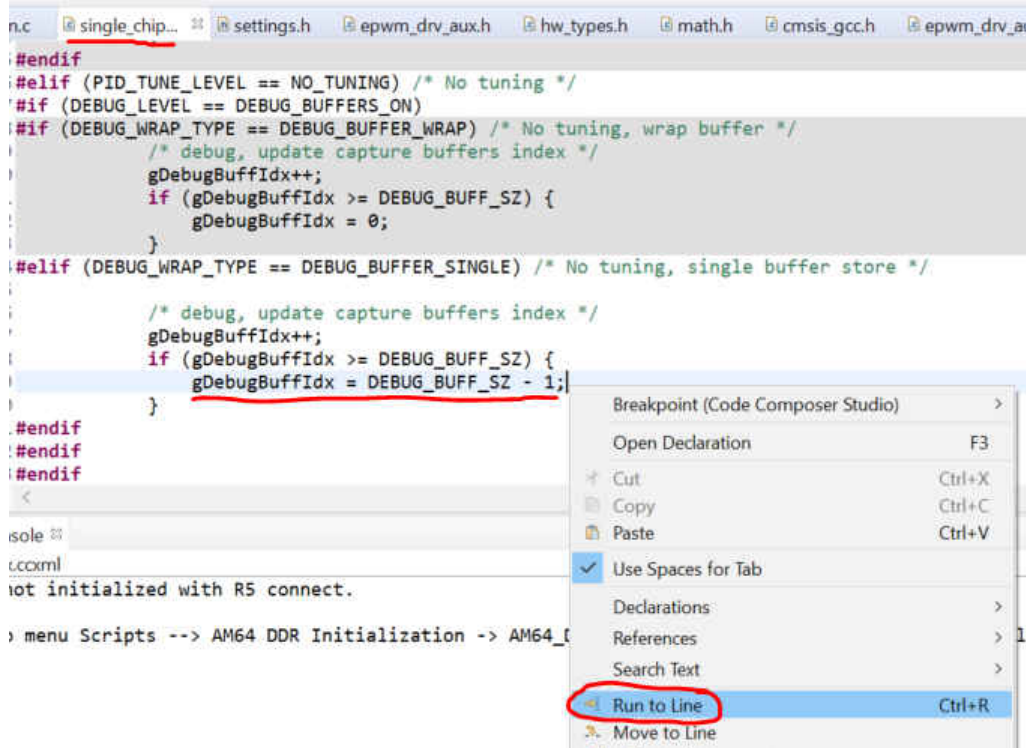


图 11-25. 闭环速度 - 运行到行

5. 运行到此行将使控制代码可以运行八个不同的设定点并填充调试缓冲区，以便可以查看此图。

6. 通过导入以下各个图来查看输出：




	closed_loop_speed_id.graphProp	8/11/2021 3:03 PM
	closed_loop_speed_iq.graphProp	8/11/2021 3:03 PM
	closed_loop_speed_speed.graphProp	8/11/2021 3:03 PM

图 11-26. 闭环速度 - ID、IQ、速度图 1

7. 以下所示为示例输出（自上而下为 Iq、Id 和速度图）。请注意在速度出现斜坡且趋近于 0 并保持稳定期间，Iq 斜坡是怎样的以及它是怎样保持高水平的。底部的速度图显示，速度出现斜坡且达到请求值，然后保持稳定。斜坡斜率由 settings.h 文件中的 MAX\_SPD\_CHANGE 定义确定。您将会注意到，在斜坡中，速度从 750 到 -500，再到 -750，-500 基本会被跳过，原因是斜坡不够陡峭，无法在允许的周期内达到该值。

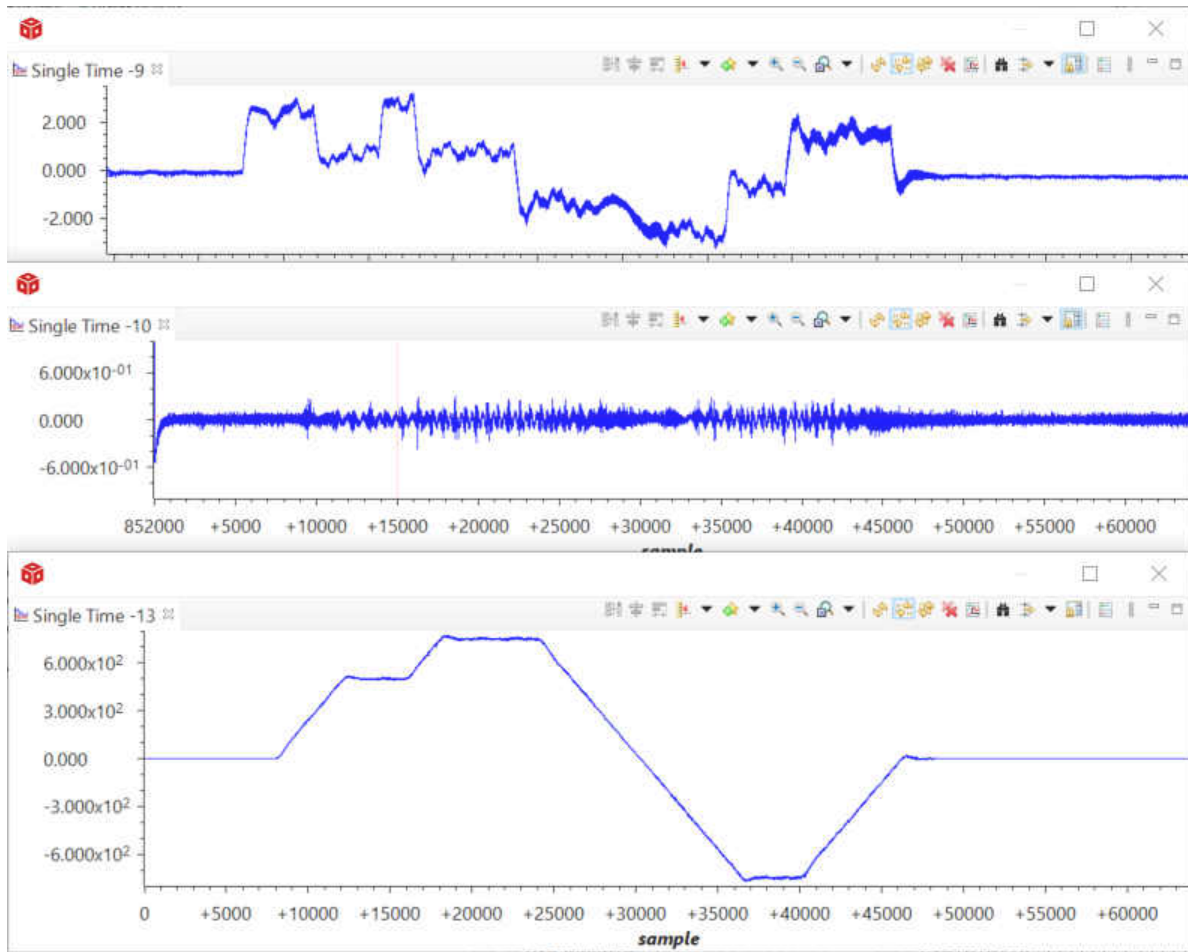


图 11-27. 闭环速度 - ID、IQ、速度图 2



## 11.7 步骤 6. 闭环位置控制 ( BUILDLEVEL == CLOSED\_LOOP\_POSITION )

除速度和电流环路外，此编译级别还可关闭位置环路。

1. 转至 `settings.h` 文件，并更改定义以匹配以下内容（闭环位置控制，启用调试缓冲区，每个设定点具有 8000 个周期）：

```
#define BUILDLEVEL          CLOSED_LOOP_POSITION
#define PID_TUNE_LEVEL      NO_TUNING
#define DEBUG_LEVEL         DEBUG_BUFFERS_ON
#define DEBUG_WRAP_TYPE    DEBUG_BUFFER_SINGLE
#define PRECOMPUTE_LEVEL   NO_PRECOMPUTE

/* The number of ISR cycles to use the same target for
#define CYCLES_PER_TARGET  8000
#define TARGET_BUFF_SIZE   CYCLES_PER_TARGET * 8

/* Iq testing value for open loop */
#define IQ_TESTING         0.2
/* Max speed change request allowed between updates */
#define MAX_SPD_CHANGE     0.12
/* Max position change request allowed between updates
#define MAX_POS_CHANGE     0.03      /* 500RPMs
```

图 11-28. 闭环位置 - BUILDLEVEL

2. 如果需要，转至 `single_chip_servo.c` 文件并修改速度设定点数组。默认值如下所示。

```
volatile float gPosArray[8] = {180,180,359.5,359.5,0.5,0.5,180,180};
```

- a. 值将重复，以便为电机留出在设定点周期数量增加前转至该位置的时间。
3. 加载完项目后，打开“`single_chip_servo.c`”文件并找到以下行，然后点击右键并选择“Run to Line”。

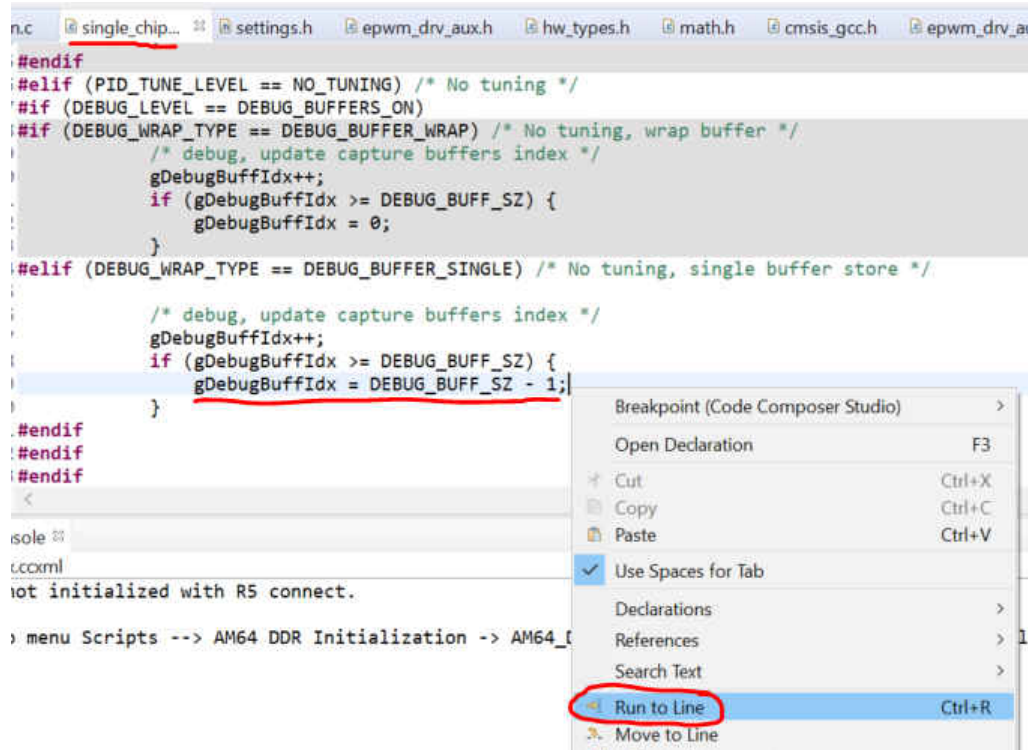


图 11-29. 闭环位置 - 运行到行

4. 运行到此行将使控制代码可以运行八个不同的设定点并填充调试缓冲区，以便可以查看此图。

5. 通过导入以下各个图来查看输出：



	closed_loop_position_position.graphProp	8/11/2021 6:55 PM
	closed_loop_position_speed.graphProp	8/11/2021 6:54 PM

图 11-30. 闭环位置 - 位置和速度图 1

6. 以下所示为示例输出（自上而下为速度和位置图）。请注意速度是如何在位置变化期间变为  $+500\text{RPM}$  并在到达该位置前保持不变的。位置斜坡斜率由 `settings.h` 文件中的 `MAX_POS_CHANGE` 定义确定。



图 11-31. 闭环位置 - 位置和速度图 2

7. 我们难以在上面的 CCS 图中了解位置的精度，因此，您也可以使用 CCS 内存浏览器将 “gDebugBuff5” 转储到一个文本文件中，以便在 Excel 或 Matlab 中查看。完成此操作后，您可以在位置稳定下来之后查看位置的实际精度：通常为请求目标  $\pm 0.001^\circ$ （如两个请求位置的图 11-32 中所示）：

179.999435	359.500366
179.999512	359.500305
179.999481	359.500305
179.999512	359.500305
179.999496	359.500336
179.999466	359.500305
179.999512	359.500305
179.999512	359.500305
179.999527	359.500244
179.999512	359.500244
179.999542	359.500244
179.999512	359.500244
179.999512	359.500244
179.999557	359.500214
179.999512	359.500214
179.999542	359.500244
179.999588	359.500275
179.999557	359.500214
179.999557	359.500275
179.999588	359.500214
179.999588	359.500244
179.999557	359.500153
179.999542	359.500214
179.999542	359.500214
179.999619	359.500183
179.999588	359.500214
179.999542	359.500153
179.999557	359.500183
179.999557	359.500122
179.999588	359.500153
179.999512	359.500153
179.999557	359.500092
179.999619	359.500153
179.999542	359.500092
179.999619	359.500153
179.999557	359.500122
179.999619	359.500031
179.99968	359.500122
179.999603	359.500122
179.999649	359.500122
179.999619	359.500092
179.999649	359.500061
179.999603	359.500031
179.99968	359.500031
179.999649	359.5
179.999619	359.5
179.99968	359.500031
179.99968	359.5
179.99968	359.500031
179.999649	359.5
179.999664	359.499939
179.999695	359.499969

图 11-32. 闭环位置 - 位置控制精度

## 12 使用 MCU+SDK 08.00.00.21 & CCS 10.3.1 进行编译

1. 从 c:\ti 下载用于 AM64x 的 MCU+SDK 08.00.00.21 并安装。
2. 下载 ( ) 并将其解压缩到 <DEMO\_INSTALL ROOT>。
3. 复制与覆盖：
  - a. 将 <DEMO\_INSTALL ROOT>\mcu\_plus\_sdk\_servo\_drive\_code\examples\motor\_control\sddf\ 复制到 C:\ti\mcu\_plus\_sdk\_am64x\_08\_00\_00\_21\examples\motor\_control
  - b. 将 <DEMO\_INSTALL ROOT>\mcu\_plus\_sdk\_servo\_drive\_code\examples\motor\_control\single\_chip\_servo\ 复制到 C:\ti\mcu\_plus\_sdk\_am64x\_08\_00\_00\_21\examples\motor\_control
  - c. 将 <DEMO\_INSTALL ROOT>\mcu\_plus\_sdk\_servo\_drive\_code\source\motor\_control\ 复制到 C:\ti\mcu\_plus\_sdk\_am64x\_08\_00\_00\_21\ource\motor\_control

4. 重新编译 EnDat 库：
  - a. `cd C:\ti\mcu_plus_sdk_am64x_08_00_00_21`
  - b. `gmake -s -f makefile.am64x motorcontrol_endat_r5f.ti-arm-clang_clean`
  - c. `gmake -s -f makefile.am64x motorcontrol_endat_r5f.ti-arm-clang`
  - d. `gmake -s -f makefile.am64x motorcontrol_endat_r5f.ti-arm-clang_clean PROFILE=debug`
  - e. `gmake -s -f makefile.am64x motorcontrol_endat_r5f.ti-arm-clang PROFILE=debug`
5. 导入和编译 CCS 工程：
  - a. 导入和编译以下文件夹中的 `single_chip_servo_am64x-evm_system_freertos_nortos`：  
`C:\ti\mcu_plus_sdk_am64x_08_00_00_21\examples\motor_control\single_chip_servo\am64x-evm`。
  - b. 系统 CCS 工程将导入三个子 CCS 工程：
    - i. `single_chip_servo_am64x-evm_r5fss0-0_nortos_ti-arm-clang`
    - ii. `single_chip_servo_am64x-evm_r5fss1-0_freertos_ti-arm-clang`
    - iii. `single_chip_servo_am64x-evm_m4fss0-0_nortos_ti-arm-clang`
  - c. 您可以编译系统 CCS 工程。它将编译全部三个子 CCS 工程。
6. 加载和运行 `single_chip_servo_am64x-evm_r5fss0-0_nortos_ti-arm-clang`：
  - a. 使用 AM64x/AM243x EVM 上的 CCS 和 JTAG 加载和运行 R5F\_0\_0 中的 `single_chip_servo_am64x-evm_r5fss0-0_nortos_ti-arm-clang`。
  - b. 电机应以 6000 个周期/分钟的速度启动引脚。

## 13 总结

工业 4.0 为伺服驱动器带来了全新的指导原则和系统要求，因此，设计人员选择适合当前和未来伺服驱动器需求的解决方案非常重要。精度、连接、效率和安全是主要考虑因素。Sitara AM243x MCU 等器件包含多达 4 个 800MHz 的 R5F、工业通信子系统和实时控制外设，非常适合用于互联高端电机驱动系统。此应用手册提供分步指南来演示伺服驱动器应用中的 AM243x，并帮助用户了解有关实现和性能的详细信息。下面总结了关键特性：

- 能够灵活地支持多个常见的工业通信协议
- AM243x/AM64x 具有可轻松在系统事件之间实现同步的独特同步架构
- AM243x/AM64x 具有可确保功能安全的专用 M4F 内核
- 通常情况下实现的控制环路频率为 8KHz 至 20KHz，相比之下，使用 800Mhz 的专用 R5F 内核可实现更高的完整电机控制环路频率（50KHz），这要归功于 Arm® R5F 内核的低延迟及更高的确定性和可靠性。
- 使用高效的 MCU+ SDK，利用单周期 TCM 和大型片上 RAM 来打造高效的无 DDR 式电机控制系统。
- 使用 800Mhz 的专用 R5F 内核来处理工业通信堆栈，以提高可靠性并能够处理千兆位 Profinet。
- 可轻松扩展到多轴电机控制 - 轻松增添功能安全特性。这归功于专用 M4F 内核能够实现功能安全。
- FOC 环路时间从 6us 缩短至 <1us。
- PRU\_ICSSG 的  $\Sigma$ - $\Delta$  滤波功能可以处理多达 9 个通道（对于 3 个轴足够了）
- PRU\_ICSSG 的 EnDat 2.2 解码功能可以处理多达 3 个通道（对于 3 个轴足够了）
- AM243x 消耗的功率不到 1W

## 14 附录 A：详细的电机控制 R5F 处理时间

下表中列出了详细的电机控制 R5F 处理时间：

表 14-1. 角度/位置/速度计算时间

	角度/位置/速度计算					单位
	开环 Iq/Id	闭环 Iq/Id	闭环速度	闭环位置	闭环 CiA402	
最小值	252	252	268	268		ns
最大值	312	308	336	336		ns
平均值	274.462	269.271	291.303	295.503		ns
标准化的最大值	132	128	156	156		ns

表 14-2. 相电流扩展和转换时间

	相电流扩展和转换为浮点					单位
	开环 Iq/Id	闭环 Iq/Id	闭环速度	闭环位置	闭环 CiA402	
最小值	188	316	460	380		ns
最大值	200	360	496	416		ns
平均值	185.139	328.791	474.26	394.462		ns
标准化的最大值	20	180	316	236		ns

表 14-3. FOC 环路时间

	FOC ( Clarke、Sin/Cos、Park、PI 控制器、逆向 Park、空间矢量生成 )					单位
	开环 Iq/Id	闭环 Iq/Id	闭环速度	闭环位置	闭环 CiA402	
最小值	308	380	444	468		ns
最大值	360	432	520	584		ns
平均值	331.301	398.24	476.313	494.554		ns
标准化的最大值	180	252	340	404		ns

表 14-4. PWM 输出时间

	写至 PWM 输出	单位
标准化的最大值	84	ns
最小值	244	ns
平均值	252.666	ns
最大值	264	ns

表 14-5. 一次测量的完整 ISR

	一次测量的完整 ISR					单位
	开环 Iq/Id	闭环 Iq/Id	闭环速度	闭环位置	闭环 CiA402	
最小值	516	700	932	892	854	ns
最大值	568	824	1064	1072	1040	ns
平均值	535.97	741.272	983.271	947.688	888.707	ns
标准化的最大值	388	644	884	892	860	ns

表 14-6. 一次测量的完整 ISR ( 预计算的 Clarke )

	一次测量的完整 ISR ( 预计算的 Clarke )					单位
	开环 Iq/Id	闭环 Iq/Id	闭环速度	闭环位置	闭环 CiA402	
最小值	516	572	820	820		ns
最大值	576	664	920	952		ns
平均值	543.08	599.084	851.557	874.515		ns
标准化的最大值	396	484	740	772		ns

## 15 参考文献

- 德州仪器 ( TI ) : [用于执行连接、控制和通信的高度集成工业驱动器](#)
- 德州仪器 ( TI ) : [基于快速串行接口 \( FSI \) 的分布式多轴伺服驱动器参考设计](#)
- [AM2434](#) 产品文件夹页面
- [AM6442](#) 产品文件夹页面
- 德州仪器 ( TI ) : [适用于伺服驱动器且具有智能栅极驱动器的 48V/500W 三相逆变器参考设计](#)
- 德州仪器 ( TI ) : [连接至绝对位置编码器的通用数字接口参考设计](#)
- 德州仪器 ( TI ) : [在工业 4.0 伺服驱动器中应用 Sitara 处理器和微控制器](#)
- 德州仪器 ( TI ) : [Sitara™ 处理器支持的工业通信协议](#)
- 连接和控制工厂车间 , AM64x/AM243x 多协议演示视频 : ( <https://training.ti.com/connect-and-control-factory-floor> )

## 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