Design Guide: TIDEP-01032

EtherCAT® Connected, Single-Chip, Dual-Servo Motor Drive Reference Design

TEXAS INSTRUMENTS

Description

This reference design showcases the ability of the AM243x device to support a fully-integrated real-time servo motor drive control and industrial communication path. This path extends from receiving EtherCAT® CiA402 target commands for velocity, to performing closed-loop FOC velocity control of dual connected motors, to passing the actual velocity values back up to the EtherCAT PLC.

Resources

TIDEP-01032	Design Folder
AM243x MCU+ SDK	Tool Folder
AM243x LaunchPad™	Tool Folder
BLDC BP	Tool Folder
AM243x Academy	Training Materials

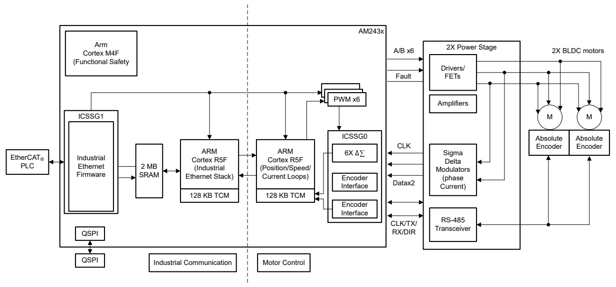
Ask our TI E2E™ support experts

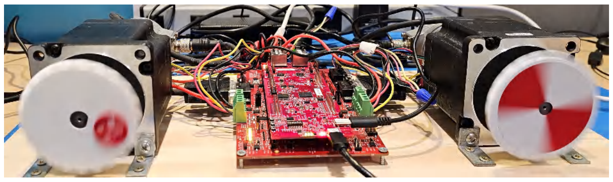
Features

- Support EtherCAT CiA402 device profile for motor velocity control
- Single-chip, dual-servo motor control
- BOOST-XL TI BoosterPack™ Plug-in Module design - 80 digital and analog I/O compatible with AM2x LaunchPad™ Development Kits
- Two axes of 3-phase BLDC motor drive with the DRV8316R 24 V, 8 A monolithic gate drive and amplifier bridges
- Two axes (6 channels) of 3-phase current feedback through AMC1035D Sigma-Delta modulator and INA241A current sense path
- Two axes of RS-485 based absolute encoder feedback supporting multiple industrial encoder standards

Applications

- Servo drive communication module
- Servo drive control module
- Servo drive position feedback
- Servo drive position sensor
- Servo drive power stage module





1 System Description

This reference design demonstrates the capacity of the AM243x device to facilitate a comprehensive real-time servo motor control and industrial communication route. This route begins with the receipt of EtherCAT CiA402 target commands for velocity, proceeds to the execution of closed-loop FOC velocity control of two connected motors, and concludes with the transmission of the actual velocity values back to the EtherCAT PLC.

1.1 Terminology

PLC	Programmable Logic Controller
FOC	Field-Oriented Control
EtherCAT	Ethernet for Control Automation Technology
CiA402	An EtherCAT Profile for Drives and Motion Control
RPM	Revolutions Per Minute
EnDAT 2.2	A digital, bidirectional interface standard for incremental and absolution position encoders
ICSS	Industrial Communication Subsystem
PRU	Programmable Real-time Unit
LP	LaunchPad™
SDFM	Sigma-Delta Filter Module
SDDF	Sigma-Delta Decimation Filtering
IPC	Inter-Processor Communication
IEP	Industrial Ethernet Peripheral
CMP	Event Comparator
ISR	Interrupt Service Routine
PWM	Pulse-Width Modulation
EPWM	Enhanced Pulse-Width Modulation

1.2 Key System Specifications

1. Supports EtherCAT CiA402 device profile for servo-motor velocity control
2. Single-chip dual-servo motor control
3. 50-kHz FOC loop for current and velocity control
4. Two axis (6 channels) of 3-phase sigma-delta modulated current feedback
5. Two axis EnDat 2.2 encoded absolute position feedback

2 System Overview

Figure 2-1 shows the setup for the TIDEP-01032 system.

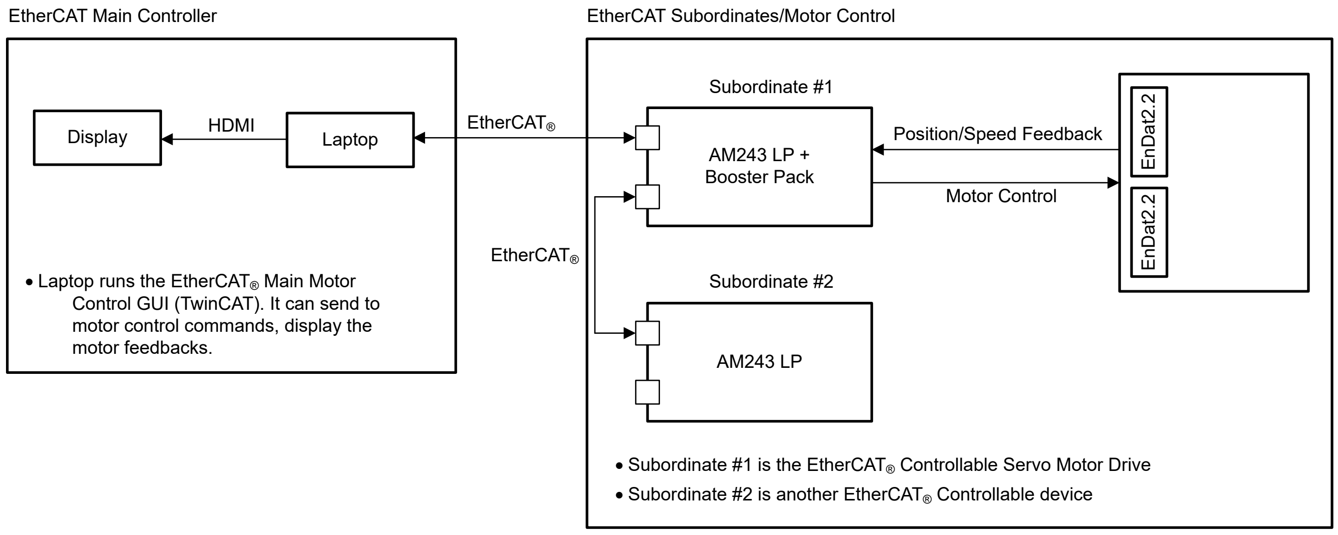

EtherCAT Main Controller
Display
HDMI
Laptop
• Laptop runs the EtherCAT® Main Motor Control GUI (TwinCAT). It can send to motor control commands, display the motor feedbacks.
EtherCAT®
EtherCAT®
EtherCAT Subordinates/Motor Control
Subordinate #1
AM243 LP + Booster Pack
Position/Speed Feedback
Motor Control
EnDat2.2
EnDat2.2
Subordinate #2
AM243 LP
• Subordinate #1 is the EtherCAT® Controllable Servo Motor Drive
• Subordinate #2 is another EtherCAT® Controllable device

Figure 2-1. System Setup

2.1 Block Diagram

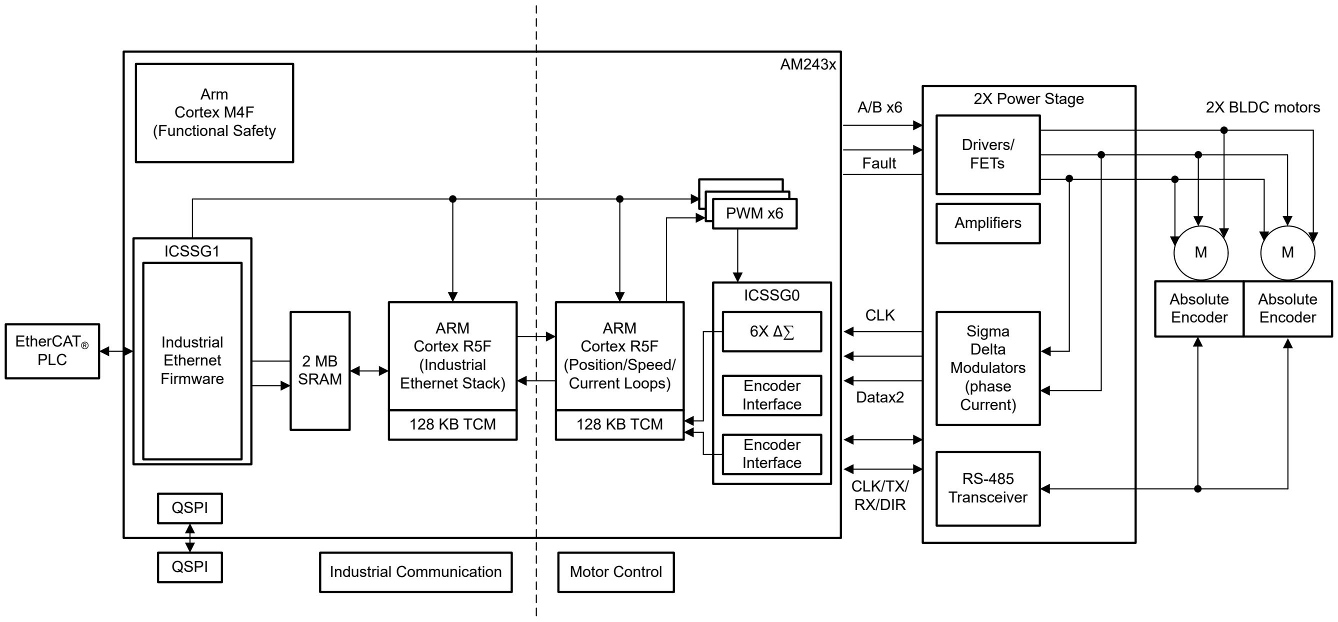


Figure 2-2. TIDEP-01032 Block Diagram

2.2 Design Considerations

The single-chip dual-axis servo motor drive implementation was architected around a central real-time path that is made up of the following:

- ICSSG1 - EtherCAT client controller firmware
- ICSSG0 - SDDF and EnDAT 2.2 decoding
  - Sigma-Delta filtering firmware with *Load Sharing* between RTU and PRU cores in PRU0 for phase current feed back from two directly connected motors
  - EnDat2.2 decoding firmware with *Loading Sharing* between RTU and PRU cores in PRU1 for angle, position, and speed feedback from two directly connected absolute encoders

- R5FSS1_0 - EtherCAT client stack implementing CiA402 using FreeRTOS
- R5FSS0_0 and R5FSS0_1 – Two independent closed-loop FOC capable of current, speed, or position closed-loop control for two directly connected motors with absolute encoders
- IPC Notify in the MCU+ SDK provides low-latency inter-core synchronization and communication
- EPWM – Six channels of enhanced PWM peripherals to generate waveforms based on the output of two FOC loops

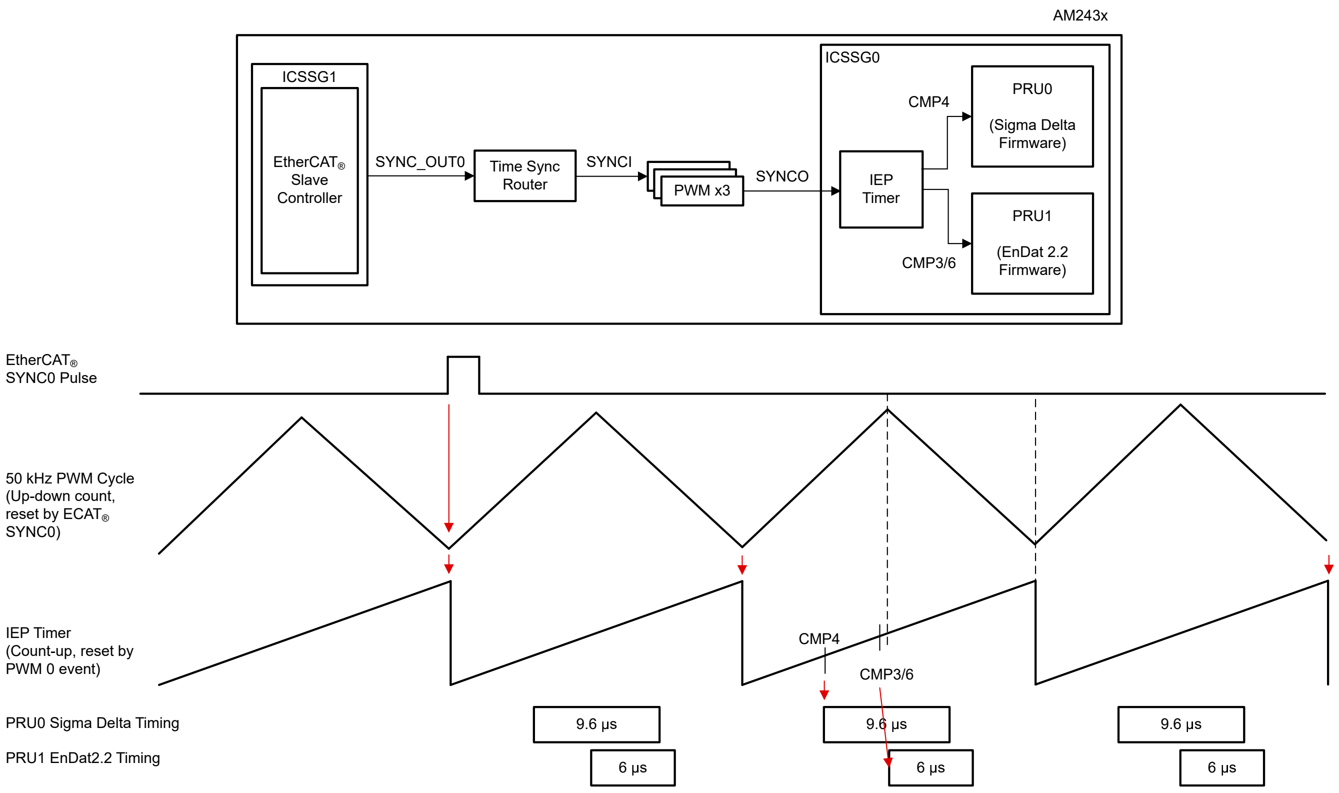


Figure 2-3. Trigger FOC Timing – 50 kHz

2.3 Highlighted Products

2.3.1 EnDAT 2.2 Interface

Table 2-1 shows the EnDAT 2.2 signal parameters.

Table 2-1. EnDAT 2.2 Signals (Two 4-wire encoders)

AM243x LP (PIN NUMBER)	BP CONNECTORS	BLDC BP	SIGNAL NAME
GPIO1_78(C16)	J8.73	VSENSOR1_SW_EN	Encoder1 enable
PRG0_PRU1_GPO0(L5)	J2.11	ENCODER_CLK1	Encoder1 clock
PRG0_PRU1_GPO2 (M2)	J7.68	ENCODER_DATA_TX_EN1	Encoder1 TX enable
PRG0_PRU1_GPO1(J2)	J7.67	ENCODER_DATA_TX1	Encoder1 TX
PRG0_PRU1_GPO13(T4)	J8.71	ENCODER_DATA_RX1	Encoder1 RX
GPIO1_77(B17)	J8.74	VSENSOR2_SW_EN	Encoder2 enable
PRG0_PRU1_GPO6(F5)	J7.69	ENCODER_CLK2	Encoder2 clock
PRG0_PRU1_GPO8(F4)	J6.57	ENCODER_DATA_TX_EN2	Encoder2 TX enable
PRG0_PRU1_GPO12(P2)	J8.72	ENCODER_DATA_TX2	Encoder2 TX
PRG0_PRU1_GPO11(P1)	J7.70	ENCODER_DATA_RX2	Encoder2 RX

EnDat 2.2 Interrupts:

- hwiPrms.intNum = ICSSG_PRU_ENDAT_INT_NUM | ICSSG_PRU_ENDAT_INT_NUM+2;
- hwiPrms.callback = &pruEncoderIrqHandler | &pruEncoderIrqHandler2;

– Motor Control Loop (FOC) one core per motor

EnDat 2.2 Input Data Buffer:

- gEndatChInfo (in .gEncChData) in R5F_0_0 TCMB

ICSSG Pin MUX:

- Mode (ICSSG_GPCFG0_REG[29-26]: PR1_PRU0_GP_MUX_SEL = 1h)
- ICSSG_SA_MX_REG[7] G_MUX_EN = 0

2.3.2 SDFM Interface

Figure 2-4 illustrates the clock source distribution and Table 2-2 shows the SDFM signals.

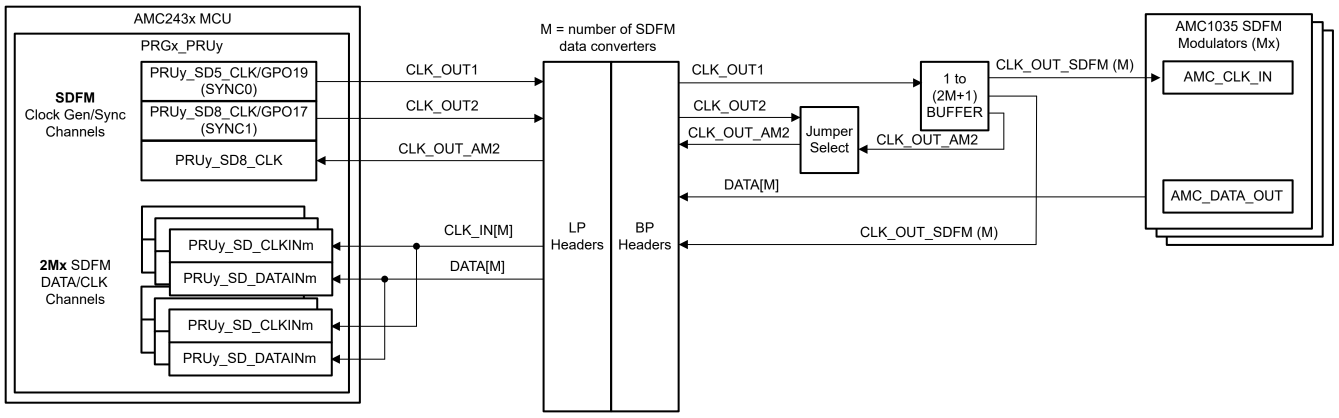


Figure 2-4. Clock Source Distribution

Table 2-2. SDFM Signals

AM243x LP (PIN NUMBER)	BP CONNECTORS	BLDC BP	SIGNAL NAME
PRG0_PRU0_GPI1(J4)	J4.32	SDFM Current High A1	SDFM Current High A1
PRG0_PRU0_GPI3(H1)	J2.19	SDFM Current High B1	SDFM Current High B1
PRG0_PRU0_GPI5(F2)	J2.13	SDFM Current High C1	SDFM Current High C1
PRG0_PRU0_GPI7(E2)	J5.44	SDFM Current High A2	SDFM Current High A2
PRG0_PRU0_GPI8(H5)	J2.15	SDFM Current High B2	SDFM Current High B2
PRG0_PRU0_GPO11(L1)	J2.12	SDFM Current High C2	SDFM Current High C2

SDFM Clock:

- PRG0_PRU0_GPO19(G2) → SYNC0 - SDFM CLOCK_OUT1(J5.45) → SDFM_CLOCK_SOURCE[1|2] (TP31)
  – AMC_CLKIN_A1, AMC_CLKIN_B1, AMC_CLKIN_C1, PR0_PRU0_SD0_CLK(J4.33), PR0_PRU0_SD1_CLK(J4.31), PR0_PRU0_SD2_CLK(J2.17)
  – AMC_CLKIN_A2, AMC_CLKIN_B2, AMC_CLKIN_C2, PR0_PRU0_SD3_CLK(J5.48), PR0_PRU0_SD6_CLK(J1.5), PR0_PRU0_SD7_CLK(J2.14)

SDFM Interrupts:

- hwiPrms.intNum = ICSSG_PRU_SDDF_INT_NUM | ICSSG_RTU_SDDF_INT_NUM;
- hwiPrms.callback = &pruSddfIrqHandler | &rtuSddfIrqHandler;

SDFM Input Data Buffers:

- gSddfChSamps[0-2] (in .gSddfChSampsRaw) in TCMB of R5F_0_0 for motor 1
- gSddfChSamps[3-5] (in .gSddfChSampsRaw) in TCMB of R5F_0_1 for motor 2

2.3.3 EPWM Interface

Table 2-3 and Table 2-4 show the *EPWM0-2 Signal Motor 1* and *EPWM3-5 Signal Motor 2* data, respectively.

Table 2-3. EPWM0-2 Signal Motor 1

AM243x LP (PIN NUMBER)	BP CONNECTORS	BLDC BP	SIGNAL NAME
GPIO1_64(B16)	J5.49	nPWM_EN_M1	DRV1 enable
GPMC0_AD8(U18)	J4.36	DRV1 EPWM High C	DRV1 PWM High C
GPMC0_AD9(U20)	J4.35	DRV1 EPWM Low C	DRV1 PWM Low C
GPMC0_AD5(T20)	J4.38	DRV1 EPWM High B	DRV1 PWM High B
GPMC0_AD6(T18)	J4.37	DRV1 EPWM Low B	DRV1 PWM Low B
GPMC0_AD3(V21)	J4.40	DRV1 EPWM High A	DRV1 PWM High A
GPMC0_AD4(U21)	J4.39	DRV1 EPWM Low A	DRV1 PWM Low A

Table 2-4. EPWM3-5 Signal Motor 2

AM243x LP (PIN NUMBER)	BP CONNECTORS	BLDC BP	SIGNAL NAME
GPIO1_65(B15)	J5.50	nPWM_EN_M2	DRV2 enable
FSI_TX0_CLK (P21)	J8.79	DRV2 EPWM High C	DRV2 PWM High C
FSI_TX0_D0(Y18)	J8.80	DRV2 EPWM Low C	DRV2 PWM Low C
TEST_LED3_RED(D1)	J8.75	DRV2 EPWM High B	DRV2 PWM High B
TEST_LED4_GREEN(F3)	J8.76	DRV2 EPWM Low B	DRV2 PWM Low B
TEST_LED1_GREEN(U19)	J8.77	DRV2 EPWM High A	DRV2 PWM High A
FSI_RX0_D1(V20)	J8.78	DRV2 EPWM Low A	DRV2 PWM Low A

EPWM Setting (**init_pwms**):

- Configure the SYNCI, SYNCO mapping to tie the three PWM groups together
- Have PWM0 SYNC from Time Sync Router 38
  – CSL_REG32_WR(CSL_TIMESYNC_EVENT_INTROUTER0_CFG_BASE + ((38 × 4) + 4), (0x10000 | 29));
- Time Sync Router input 29 (ICSSG1 IEP0 SYNC0) → Time Sync Router output 38
  – CSL_REG32_WR(CSL_CTRL_MMR0_CFG0_BASE + CSL_MAIN_CTRL_MMR_CFG0_EPWM0_CTRL, (2 << CSL_MAIN_CTRL_MMR_CFG0_EPWM0_CTRL_SYNCIN_SEL_SHIFT));
  – TIMESYNC_INTRTR0_IN_29: PRU_ICSSG1_PR1_EDC0_SYNC0_OUT_0 (IEP0 sync event 0)
  – timesync_event_introuter_out_38: epwm0_sync.input2
  – TI E2E: [FAQ] *AM64x: What is the Time Sync Router for? How do I use it?*
- Have PWM3 SYNC from Time Sync Router 39
  – CSL_REG32_WR(CSL_CTRL_MMR0_CFG0_BASE + CSL_MAIN_CTRL_MMR_CFG0_EPWM3_CTRL, (2 << CSL_MAIN_CTRL_MMR_CFG0_EPWM3_CTRL_SYNCIN_SEL_SHIFT));
- Time Sync Router input 29 (ICSSG1 IEP0 SYNC0) → Time Sync Router output 39
  – CSL_REG32_WR(CSL_TIMESYNC_EVENT_INTROUTER0_CFG_BASE + ((39 × 4) + 4), (0x10000 | 29));
  – TIMESYNC_INTRTR0_IN_29: PRU_ICSSG1_PR1_EDC0_SYNC0_OUT_0 (IEP0 sync event 0)
  – timesync_event_introuter_out_39: epwm3_sync.input2
- Have PWM6 SYNC from Time Sync Router 40
  – CSL_REG32_WR(CSL_CTRL_MMR0_CFG0_BASE + CSL_MAIN_CTRL_MMR_CFG0_EPWM6_CTRL, (2 << CSL_MAIN_CTRL_MMR_CFG0_EPWM6_CTRL_SYNCIN_SEL_SHIFT));

- Time Sync Router input 29 (ICSSG1 IEP0 SYNC0) → Time Sync Router output 40
  - CSL_REG32_WR(CSL_TIMESYNC_EVENT_INTROUTER0_CFG_BASE + ((40 * 4) + 4), (0x10000 | 29));
  - TIMESYNC_INTRTR0_IN_29: PRU_ICSSG1_PR1_EDC0_SYNC0_OUT_0 (IEP0 sync event 0)
  - timesync_event_introuter_out_40: epwm6_sync.input2
  - Force SW sync for EPWM0. Other PWMs are synchronized through hardware synchronized daisy-chain
    - Epwm_tbTriggerSwSync(gEpwm0BaseAddr);
      - HW_WR_FIELD16(((gEpwm0BaseAddr + PWMSS_EPWM_OFFSET) + PWMSS_EPWM_TBCTL), PWMSS_EPWM_TBCTL_SWFSYNC, (uint16_t)PWMSS_EPWM_TBCTL_SWFSYNC_FORCE_SYNC);
- Set EPWM to 50 kHz:
  - appEpwmCfg.epwmOutFreq = gEpwmOutFreq;
  - App_epwmConfig(&appEpwmCfg, &epwm2PrdVal, &epwm2CmpAVal);

EPWM0 Interrupt:

- hwiPrms.intNum = EPWM0_INTR;
- hwiPrms.callback = &App_epwmIntrISR;

EPWM0 Output Data:

- gEpwmPrdVal

2.3.4 ICSS-PRU IEP

The following parameters apply for the IEP CMP setup:

- 50-kHz EPWM cycle time
- 50-kHz FOC loop update (In EnDAT ISR)
- SYNC_OUT0 from EtherCAT Client (ICSSG1) to Sync EPWM clock
- PRU_ICSSG0 IEP0 period is set to 6000 (300000000/50000)
  - It is also the EPWM period
  - PRU_ICSSG IEP0 based address is 0x3002E000
- Setup CMP4 to trigger Sigma Delta encoded current feedback data sampling:
  - One CMP4 in per IEP or EPWM period: 10us (defined in gTestSdfmPrms.firstSampTrigTime)
  - Set in initPruSddf
- Set up CMP3 and CMP6 to trigger EnDAT 2.2 encoded position feedback data sampling:
  - One CMP3 and CMP6 in per IEP or EPWM period: 3000 ns (defined in endat_periodic_interface.cmp3 and endat_periodic_interface.cmp6)
  - Set in endat_config_periodic_mode

2.3.5 EtherCAT CiA402 Velocity Control

Target Velocity (EtherCAT controller → EtherCAT subordinate → AL243x LP):

- gCurTargetVelocity[3] (in .gEtherCatCia402) in non-cached on chip RAM
- Set by EtherCAT controller via GUI
- Transmitted to EtherCAT subordinate
- Saved to gCurTargetVelocity[3] via EC_SLV_APP_CSV
- Used in pruEncoderIrqHandler or pruEncoderIrqHandler2 for velocity control

Actual Velocity (EtherCAT controller ← EtherCAT subordinate ← AL243x LP):

- gCurActualVelocity[3] (in .gEtherCatCia402) in non-cached on chip RAM
- Saved to gCurTargetVelocity[3] by pruEncoderIrqHandler or pruEncoderIrqHandler2 for actual velocity
- Transmitted to EtherCAT controller via EC_SLV_APP_CSV
- Displayed in GUI by EtherCAT controller

3 System Design

R5F_0_1 Initialization

Use the following steps to initialize **R5F_0_0 for motor 1** (single_chip_servo_remote_core_start):

1. Set up GPIO pin direction and initial values (init_gpio_state)
2. Disable EPWM (enable_pwm_buffers)
3. Set up EPWM frequency and interrupt for motor 1 (init_pwms)
4. Set up ICSSG0 PRU1 for EnDAT 2.2 motor 1 (channel 0, init_encoder)
   - Configure g_mux_en to 1 in ICSSG_SA_MX_REG Register. HW_WR_REG32((CSL_PRU_ICSSG0_PR1_CFG_SLV_BASE+0x40), (0x80))
   - Register & enable ICSSG EnDat PRU FW interrupt
   - – Interrupt number: ICSSG_PRU_ENDAT_INT_NUM
     – Callback function: pruEncoderIrqHandler
   - Set up the EnDat 2.2 parameters
     – gEndat_multi_ch_mask = ENDAT_MULTI_CH0 | ENDAT_MULTI_CH2;
     – gEndat_is_multi_ch = CONFIG_ENDAT0_MODE & 1;
     – gEndat_is_load_share_mode = CONFIG_ENDAT0_MODE & 2;
   - Initialize ICSSG0 PRU1 (endat_pruss_init)
   - Initialize the encoder using the encoder driver API (endat_init)
   - Configure the encoder using the encoder driver API (endat_config_multi_channel_mask)
   - Configure Delays based on the ICSSG frequency
   - Load and run the EnDat 2.2 PRU FW to ICSSG0 PRU1 (endat_pruss_load_run_fw)
   - Check initialization ack from firmware, with a timeout of 5 second (endat_wait_initialization)
   - Set default frequency to 16 MHz for 2.2 encoders (endat_init_clock)
   - Set propagation delay to make 16 MHz work at 300-MHz PRU (endat_handle_prop_delay(priv, 265))
   - SetsEnDat 2.2 FW to periodic triggering (endat_config_periodic_trigger)
   - Configures parameters for EnDat 2.2 FW periodic mode (endat_config_periodic_mode)
     – IEP0 CMP3 event for channel 0 (3000ns from start of PWM period)
     – IEP0 CMP6 event for channel 2 (3000ns from start of PWM period)
   - Start receiving EnData 2.2 data (endat_handle_rx)
5. Set up ICSSG0 PRU0 for SDFM for motor 1 (init_sddf)
   - Initialize IEP0, configure SYNC0 SD clock (init_IEP0_SYNC)
   - Initialize ICSSG0 PRU0 (initIcss)
   - Register and enable ICSSG SDFM RTU FW interrupt
     – Interrupt number: ICSSG_RTU_SDDF_INT_NUM
     – Callback function: rtuSddfIrqHandler
   - Initialize RTU/PRU core for SDFM (initPruSddf)
     – RTU sample base address: gTestSdfmPrms.samplesBaseAddress
     – PRU sample base address: gTestSdfmPrms.samplesBaseAddress+0x80
   - Start IEP0 (start_IEP0)
   - Force SW sync for EPWM0. Other PWMs are synchronized through hardware sync daisy-chain (EPWM_tbTriggerSwSync)
   - Disable the EPWM output buffer (enable_pwm_buffers)
6. Initialize the parameters for FOC (init_pids)
7. Enable the EPWM output buffers for motor 1 (enable_pwm_buffers(TRUE))

R5F_0_1 Initialization

Use the following steps to initialize **R5F_0_1 for motor 2** (single_chip_servo_remote_core_start):

1. Set up EPWM frequency and interrupt (init_pwms) for motor 2
2. Register and enable ICSSG EnDat PRU FW interrupt for motor 2
   - Interrupt number: ICSSG_PRU_ENDAT_INT_NUM+2
   - Callback function: pruEncoderIrqHandler2
3. Register & enable ICSSG SDFM PRU FW interrupt for motor 2

    - Interrupt number: ICSSG_PRU_SDDF_INT_NUM
    - Callback function: pruSddfIrqHandler
4. Initialize the parameters for FOC (init_pids)

Set up Interrupts

Use the following instructions to set the **Interrupts** and **Handlers**:

EPWM interrupt (50 kHz), ISR - App_epwmIntrISR (Motor 1) or App_epwmIntrISR2 (Motor 2)

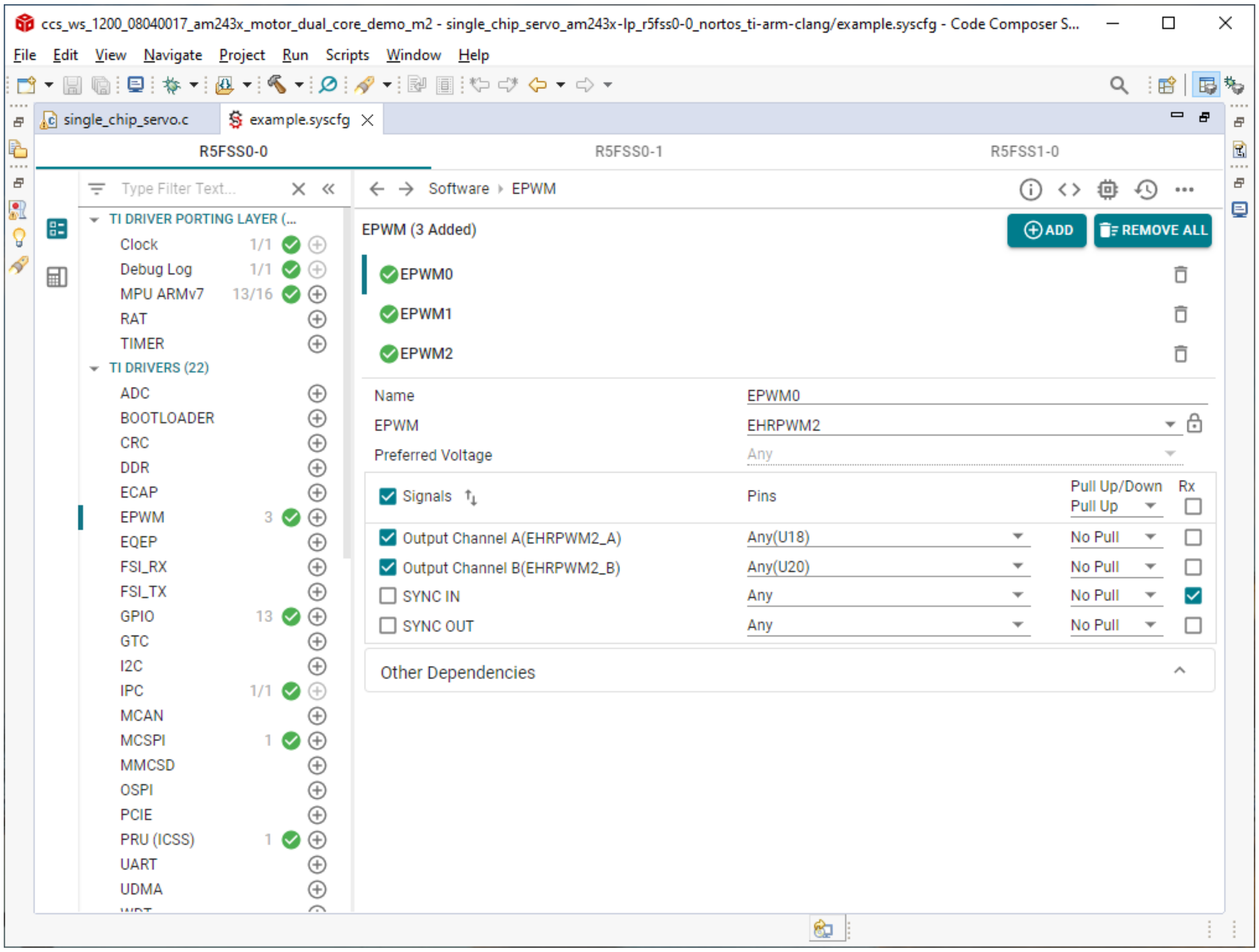
- Clear the EPWM interrupt

SDFM interrupts (50 kHz), ISR - rtuSddfIrqHandler (Motor 1) or pruSddfIrqHandler (Motor 2)

- From sample 8192 to 16384, compute the SDFM channel offsets (0–2 or 3–5). The SDFM channel offsets are used in the FOC loop when PRECOMPUTE_LEVEL == NO_PRECOMPUTE
- At sample 16384, write EPWM for Phase A, Phase B, and Phase C to lock the rotor to electrical 0 and disable SDFM interrupts
- Clear interrupt at source

EnDAT 2.2 interrupt (50 kHz), ISR – pruEncoderIrqHandler (for Motor 1)

1. Clear interrupt at source
2. For sample 0 – 8192 doing nothing
3. For sample 8193–16383:
    - Calculate mechanical and electrical angle offset (localEnDatGetSingleMulti)
4. For sample 16384:
    - Find the average of the mechanical and electrical angle offset
    - Turn off all phases
    - Save the mechanical and electrical angle offset
5. For sample after 16384:
    - Start FOC loop and unlock the rotor
    - Get the latest mechanical theta and multiturn position from the encoder (localEnDatGetSingleMulti)
    - Use calculated offset from electrical 0, 4 pole pairs
    - Running FOC loop to compute the space vector
    - Write next CMPA values. Swap cmp0 and cmp2 because the HW connect EPWM0 to Phase C and EPWM2 to Phase A
    - EPWM0 is actually uses EHRPWM2; EPWM1 is using EHRPWM1 and EPWM2 is using EHRPWM0
    - See the EPWM settings in example_syscfg of single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang for details



ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2 - single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang/example.syscfg - Code Composer S...

File Edit View Navigate Project Run Scripts Window Help

single_chip_servo.c | example.syscfg

R5FSS0-0 | R5FSS0-1 | R5FSS1-0

Software › EPWM

TI DRIVER PORTING LAYER (...

Clock	1/1
Debug Log	1/1
MPU ARMv7	13/16
RAT	
TIMER	

TI DRIVERS (22)

ADC	
BOOTLOADER	
CRC	
DDR	
ECAP	
EPWM	3
EQEP	
FSI_RX	
FSI_TX	
GPIO	13
GTC	
I2C	
IPC	1/1
MCAN	
MCSPI	1
MMCSD	
OSPI	
PCIE	
PRU (ICSS)	1
UART	
UDMA	

EPWM (3 Added) ADD REMOVE ALL

- EPWM0
- EPWM1
- EPWM2

Name	EPWM0
EPWM	EHRPWM2
Preferred Voltage	Any

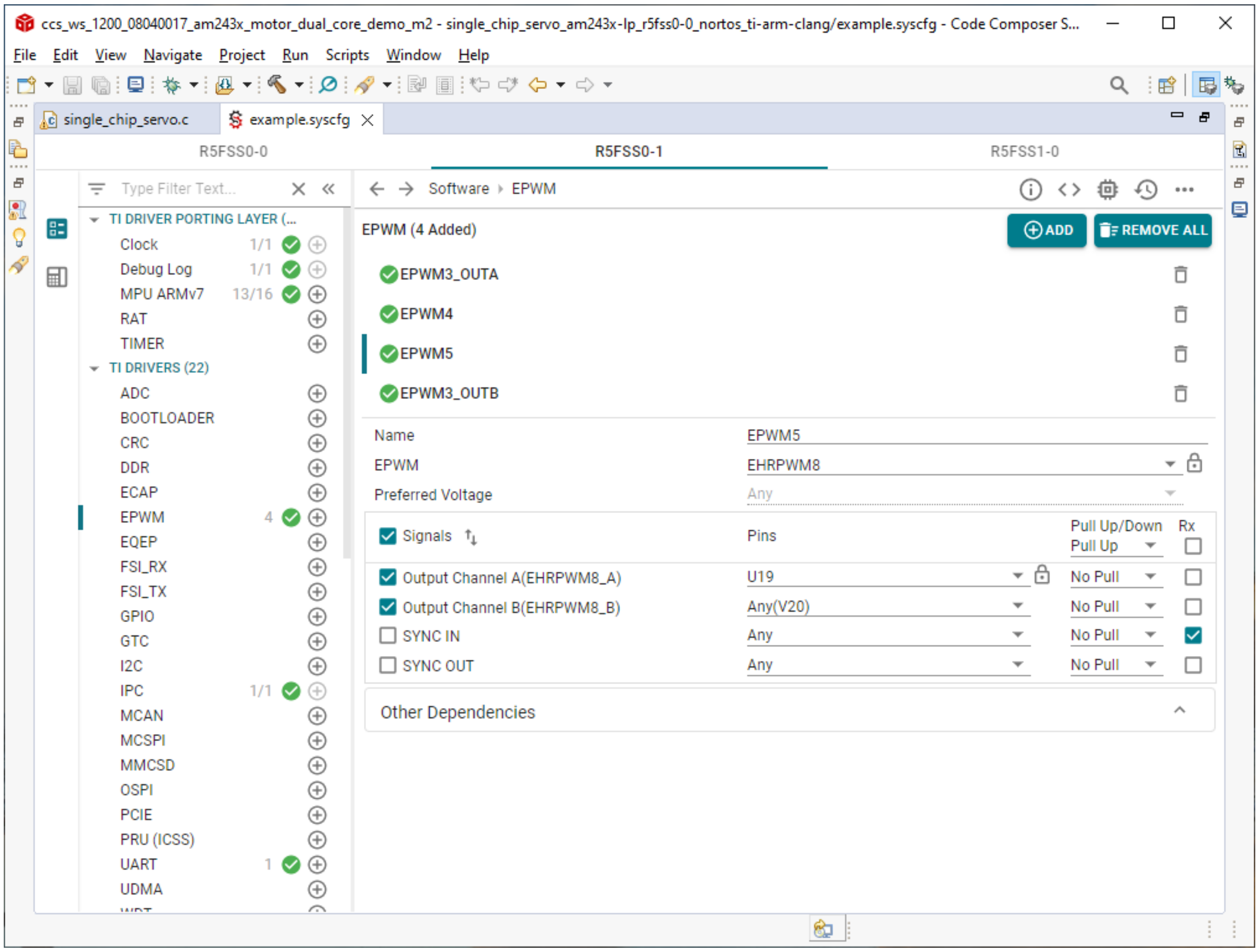
Signals	Pins	Pull Up/Down	Rx
✓ Signals		Pull Up	☐
✓ Output Channel A(EHRPWM2_A)	Any(U18)	No Pull	☐
✓ Output Channel B(EHRPWM2_B)	Any(U20)	No Pull	☐
☐ SYNC IN	Any	No Pull	✓
☐ SYNC OUT	Any	No Pull	☐

Other Dependencies

Figure 3-1. EPWM Settings for Motor 1

EnDAT 2.2 interrupt (50 kHz), ISR – pruEncoderIrqHandler (for Motor 2)

- Clear interrupt at source
- For sample 0–8192 doing nothing
- For sample 8193–16383:
  - Calculate mechanical and electrical angle offset (localEnDatGetSingleMulti)
- For sample 16384:
  - Find the average of the mechanical and electrical angle offset
  - Turn off all phases
  - Save the mechanical and electrical angle offset
- For sample after 16384:
  - Start FOC loop and unlock the rotor
  - Get the latest mechanical theta and multi-turn position from the encoder (localEnDatGetSingleMulti)
  - Use calculated offset from electrical 0, 4 pole pairs
  - Running FOC loop to compute the space vector
  - Write next CMPA values. Swap cmp0 and cmp2 because the HW connect EPWM0 to Phase C and EPWM5 to Phase A
  - EPWM0 is actually uses EHRPWM5; EPWM1 is using EHRPWM4 and EPWM2 is using EHRPWM3
  - See the EPWM settings in example_syscfg of single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang for details


ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2 - single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang/example.syscfg - Code Composer S...
File Edit View Navigate Project Run Scripts Window Help
single_chip_servo.c
example.syscfg
R5FSS0-0
R5FSS0-1
R5FSS1-0
Type Filter Text...
Software › EPWM
ADD
REMOVE ALL

TI DRIVER PORTING LAYER (...
- Clock 1/1
- Debug Log 1/1
- MPU ARMv7 13/16
- RAT
- TIMER

TI DRIVERS (22)
- ADC
- BOOTLOADER
- CRC
- DDR
- ECAP
- EPWM 4
- EQEP
- FSI_RX
- FSI_TX
- GPIO
- GTC
- I2C
- IPC 1/1
- MCAN
- MCSPI
- MMCSD
- OSPI
- PCIE
- PRU (ICSS)
- UART 1
- UDMA

EPWM (4 Added)
- EPWM3_OUTA
- EPWM4
- EPWM5
- EPWM3_OUTB

Name	EPWM5
EPWM	EHRPWM8
Preferred Voltage	Any

Signals	Pins	Pull Up/Down	Rx
✓ Signals		Pull Up	☐
✓ Output Channel A(EHRPWM8_A)	U19	No Pull	☐
✓ Output Channel B(EHRPWM8_B)	Any(V20)	No Pull	☐
☐ SYNC IN	Any	No Pull	✓
☐ SYNC OUT	Any	No Pull	☐

Other Dependencies

Figure 3-2. EPWM Settings for Motor 2

4 Hardware, Software, Testing Requirements, and Test Results

4.1 Hardware Requirements

The following equipment is required to test this reference design:

- One Microsoft® Windows® personal computer (PC) with TwinCAT automation software installed
- One LP-AM243 Evaluation board | TI.com
- One BP-AM2BLDCSERVO — AM2x Brushless-DC (BLDC) Servo Motor BoosterPack
- Two BLY342D-48V-3200 Anaheim Automation 3-phase Brushless DC motors
- Two ROQ-437 EnDat2.2 Encoders with cables

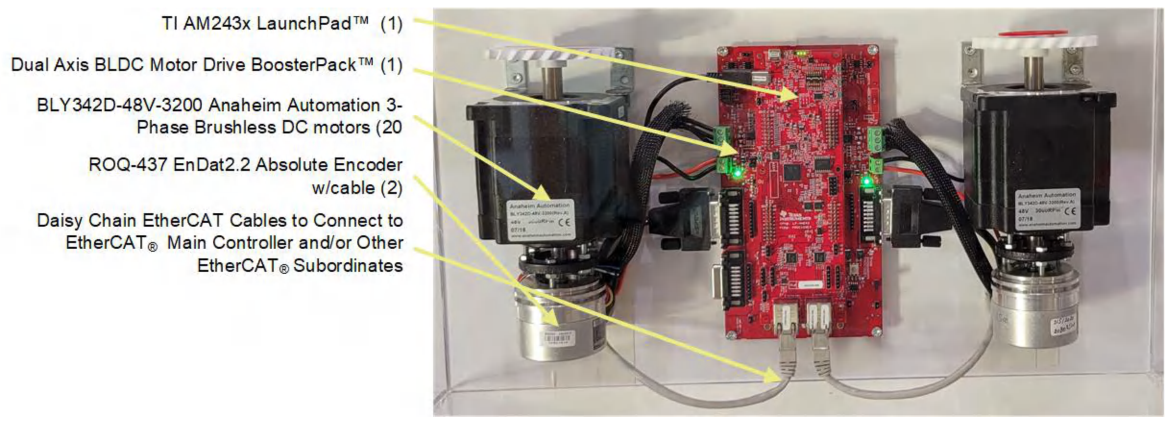

TI AM243x LaunchPad™ (1)
Dual Axis BLDC Motor Drive BoosterPack™ (1)
BLY342D-48V-3200 Anaheim Automation 3-Phase Brushless DC motors (20
ROQ-437 EnDat2.2 Absolute Encoder w/cable (2)
Daisy Chain EtherCAT Cables to Connect to EtherCAT® Main Controller and/or Other EtherCAT® Subordinates

Figure 4-1. System Hardware Configuration

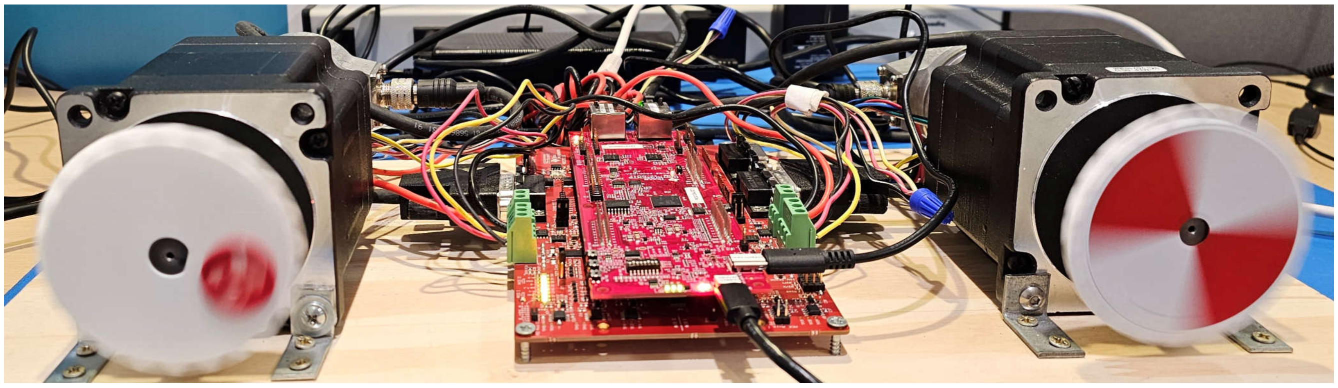


Figure 4-2. Dual Motor Drive System Setup

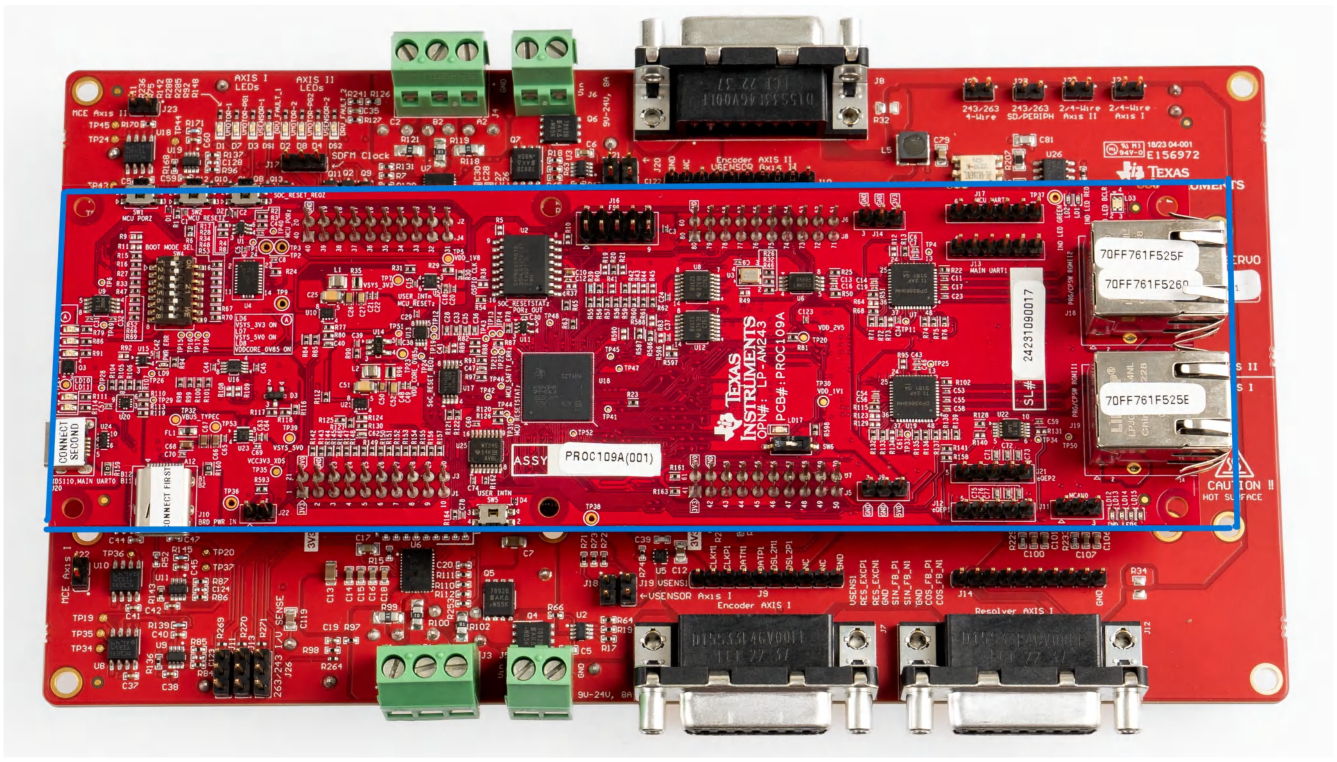


Figure 4-3. AM243x LP Rev A and BLDC BP E2

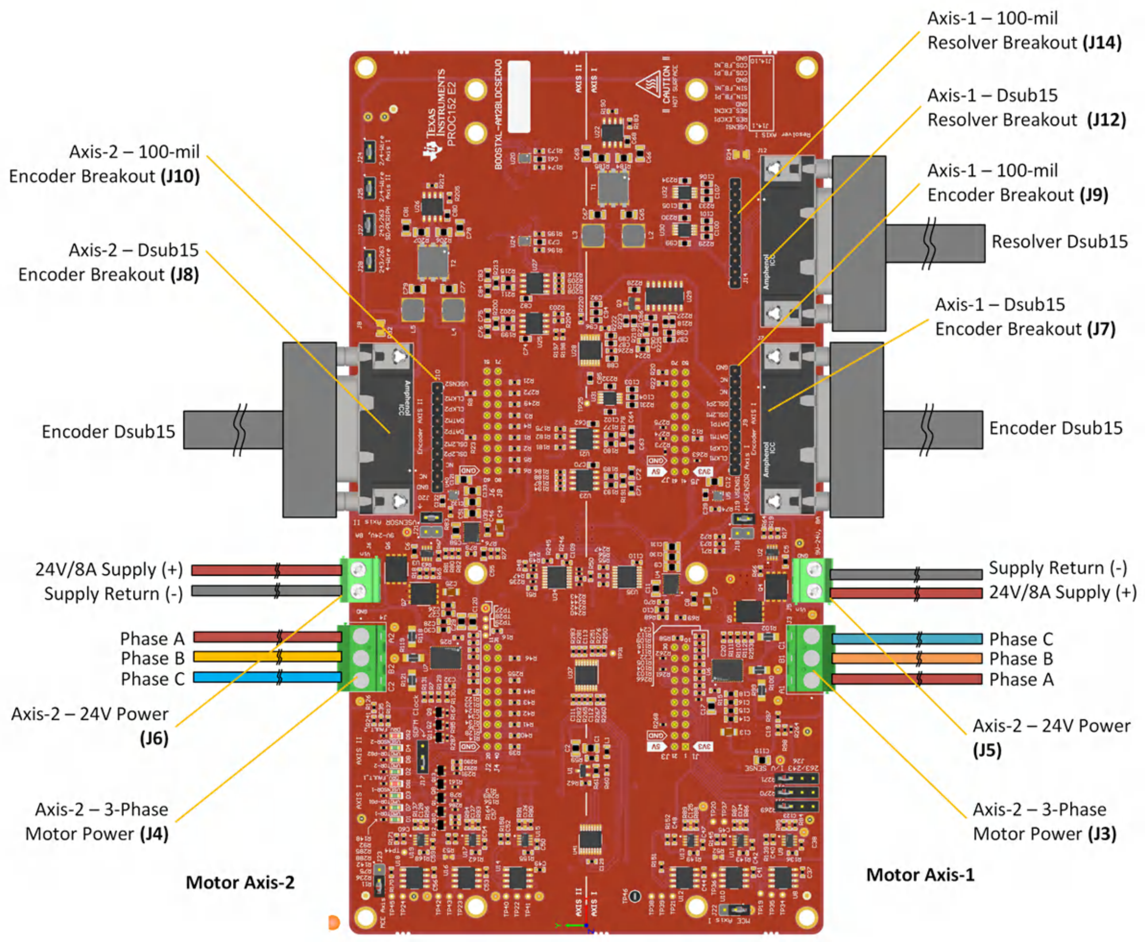
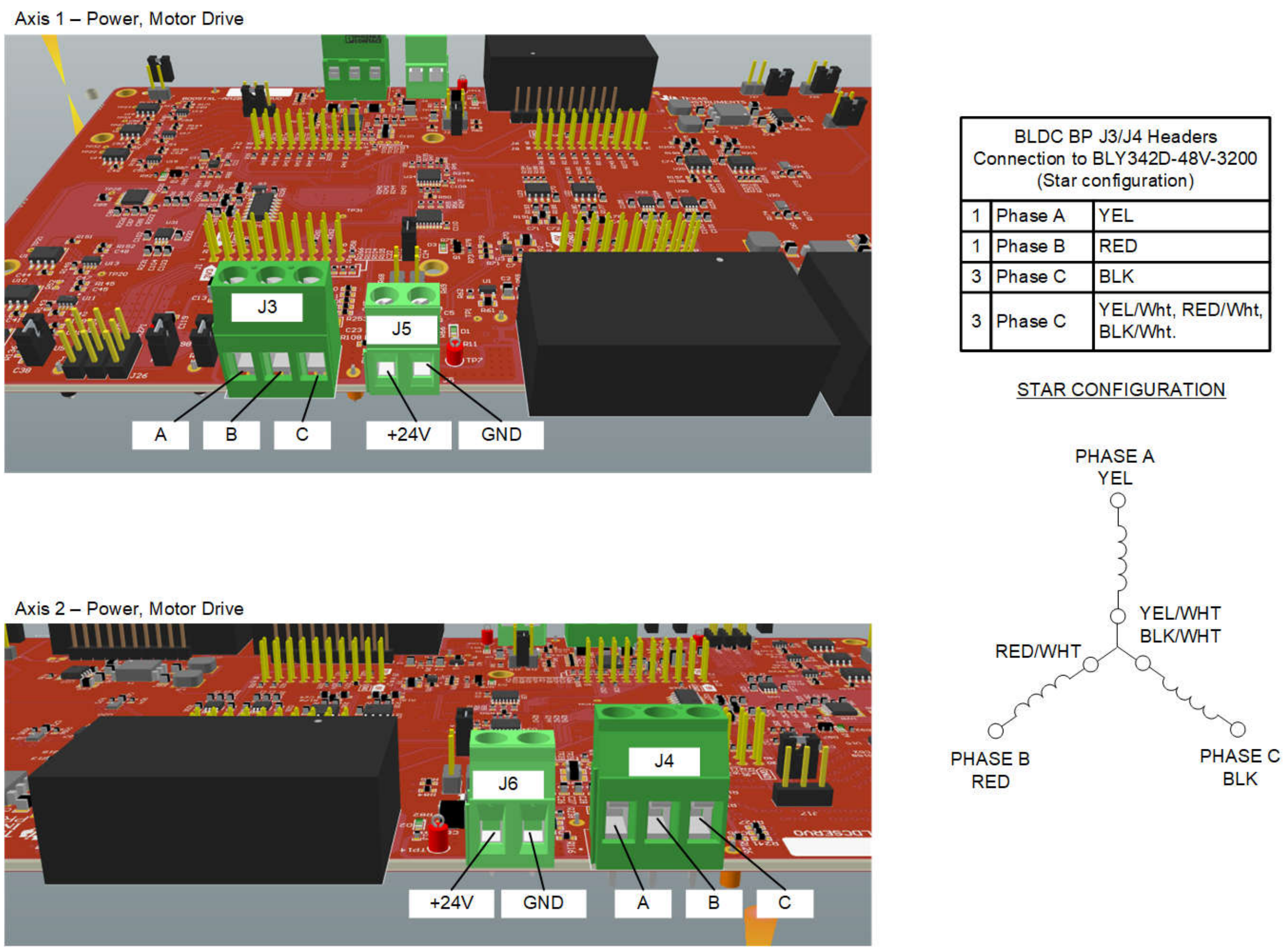

Axis-1 – 100-mil Resolver Breakout **(J14)**
Axis-1 – Dsub15 Resolver Breakout **(J12)**
Axis-2 – 100-mil Encoder Breakout **(J10)**
Axis-1 – 100-mil Encoder Breakout **(J9)**
Axis-2 – Dsub15 Encoder Breakout **(J8)**
Resolver Dsub15
Axis-1 – Dsub15 Encoder Breakout **(J7)**
Encoder Dsub15
Encoder Dsub15
24V/8A Supply (+)
Supply Return (-)
Supply Return (-)
24V/8A Supply (+)
Phase A
Phase B
Phase C
Phase C
Phase B
Phase A
Axis-2 – 24V Power **(J6)**
Axis-2 – 24V Power **(J5)**
Axis-2 – 3-Phase Motor Power **(J4)**
Axis-2 – 3-Phase Motor Power **(J3)**
Motor Axis-2
Motor Axis-1

Figure 4-4. BLDC BP E2 Connectors for Motor 1 and Motor 2



BLDC BP J3/J4 Headers Connection to BLY342D-48V-3200 (Star configuration)		
1	Phase A	YEL
1	Phase B	RED
3	Phase C	BLK
3	Phase C	YEL/Wht, RED/Wht, BLK/Wht.

Figure 4-5. Star Configuration for Motor 1 and Motor 2

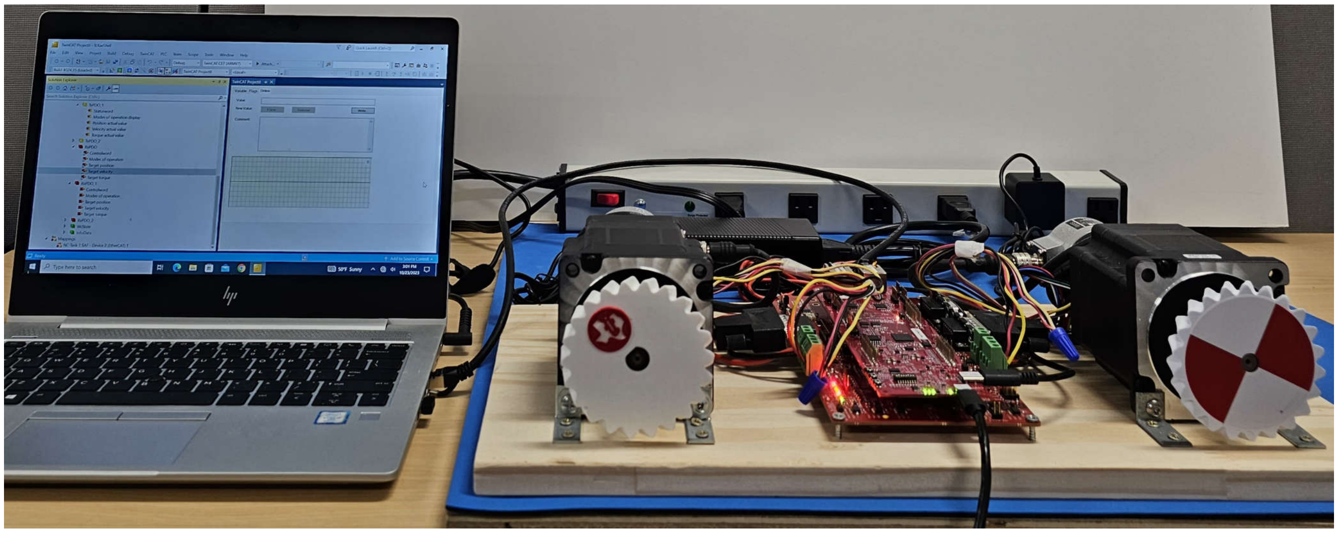
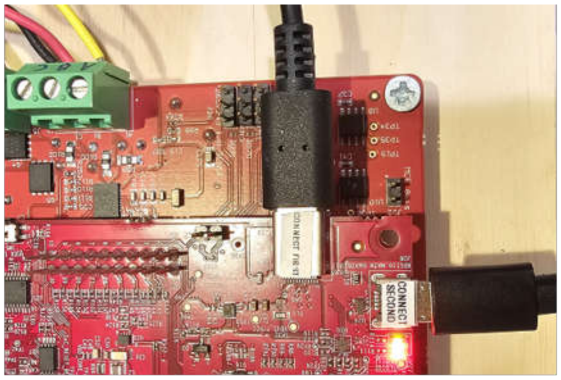


Figure 4-6. EtherCAT Connected Motor Control Setup



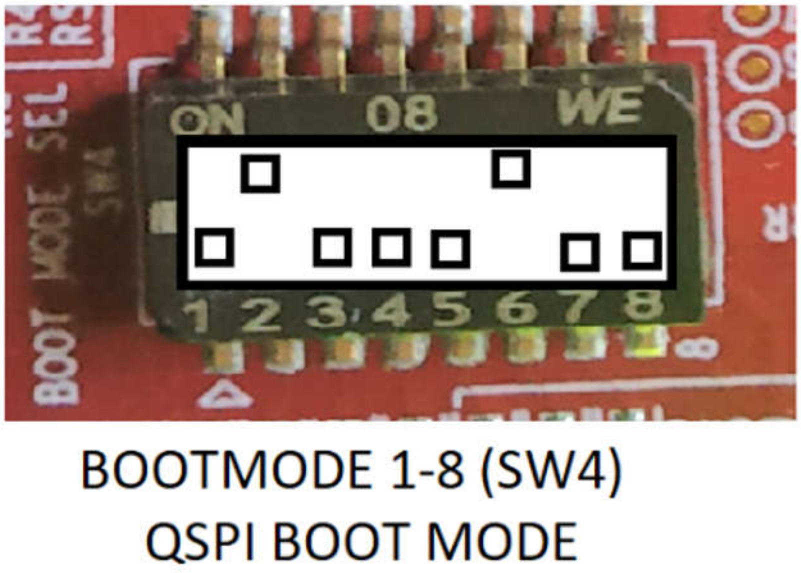


Figure 4-7. AM243x LP Power, JTAG, and Boot Mode

4.2 Software Requirements

To explore the reference design source code, download and install the Motor Control SDK 09.01.00.xx at `C:\ti\motor_control_sdk_am243x_09_01_00_xx`. In Motor Control SDK 09.01.00, the tidep_01032_dual_motor_drive holds the source code of the reference design.

Import and build the system project from C:\ti\motor_control_sdk_am243x_09_01_00_xx\examples\tidep_01032_dual_motor_drive\single_chip_servo\am243x-lp\system_freertos_nortos. The project has 3 sub-projects:

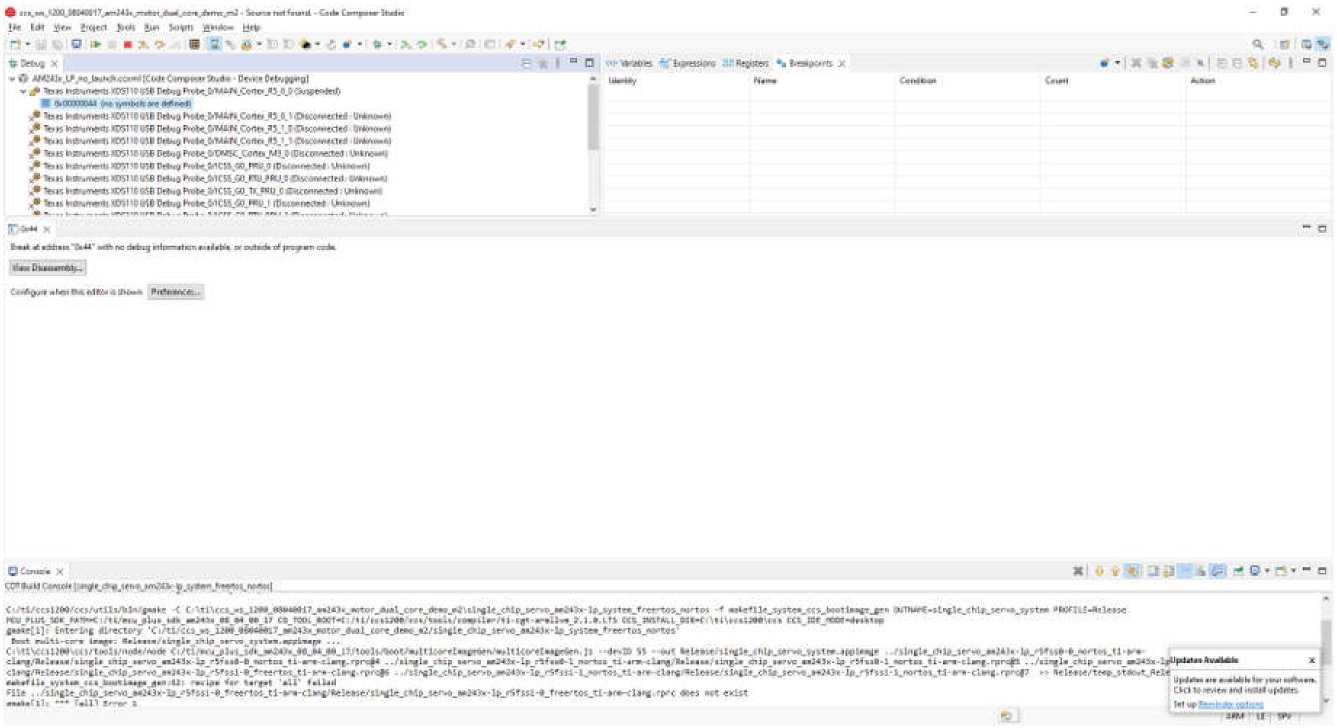
1. `C:\ti\motor_control_sdk_am243x_09_01_00_xx\examples\tidep_01032_dual_motor_drive\single_chip_servo\am243x-lp\system_freertos_nortos \r5fss0-0_nortos` (single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang: motor drive code for axle 1)
2. `C:\ti\motor_control_sdk_am243x_09_01_00_xx\examples\tidep_01032_dual_motor_drive\r5fss0-1_nortos` (single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang: motor drive code for axle 2)
3. `C:\ti\motor_control_sdk_am243x_09_01_00_xx\examples\tidep_01032_dual_motor_drive\r5fss1-0_freertos` (ethercat_slave_cia402_demo_am243x-lp_r5fss1-0_freertos_ti-arm-clang: EtherCAT CiA402 client code)

4.3 Test Setup

This section provides instructions for how to load and run the test software:

After importing and building the system project, the executable binary files for R5F_0_0, R5F_0_1 and R5F_1_0 appear in the CCS workspace directory: `C:\ti\ccs_ws_1250_am243x_mcsdk_09.01.00.01`

1. Connect to the target AM243x LP using the target configuration file
2. Load and Run Motor Control 1 – R5F_0_0
    - Halt R5F_0_0
    - Load and run single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang
    - Motor 1 is supposed to start spinning at 120 RPM



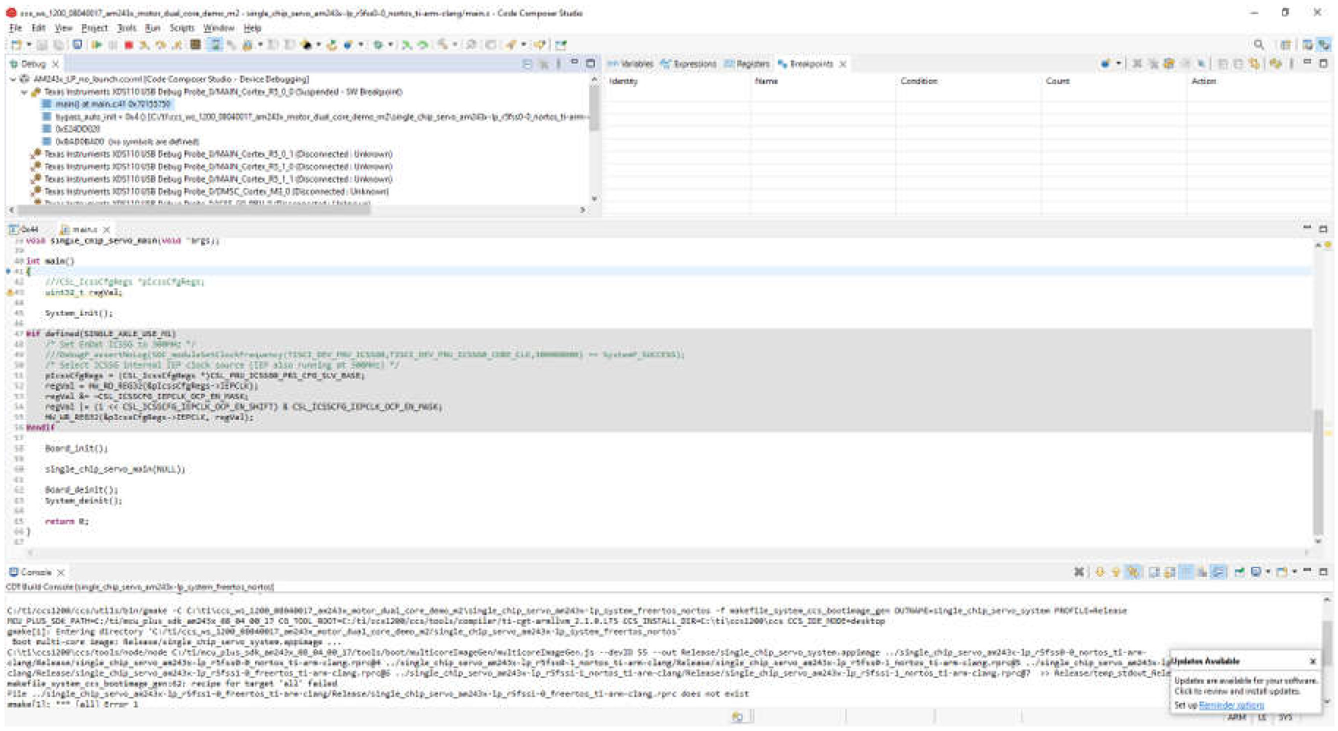
```
AM243x_LP_no_launch.ccxml [Code Composer Studio - Device Debugging]
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_0_0 (Suspended)
    0x00000044 (no symbols are defined)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_0_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_1_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_1_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/DMSC_Cortex_M3_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_PRU_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_RTU_PRU_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_TX_PRU_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_PRU_1 (Disconnected : Unknown)
```

Break at address "0x44" with no debug information available, or outside of program code.

Updates are available for your software. Click to review and install updates.

```
CDT Build Console [single_chip_servo_am243x-lp_system_freertos_nortos]

C:/ti/ccs1200/ccs/utils/bin/gmake -C C:\ti\ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2\single_chip_servo_am243x-lp_system_freertos_nortos -f makefile_system_ccs_bootimage_gen OUTNAME=single_chip_servo_system PROFILE=Release
MCU_PLUS_SDK_PATH=C:/ti/mcu_plus_sdk_am243x_08_04_00_17 CG_TOOL_ROOT=C:/ti/ccs1200/ccs/tools/compiler/ti-cgt-armllvm_2.1.0.LTS CCS_INSTALL_DIR=C:\ti\ccs1200\ccs CCS_IDE_MODE=desktop
gmake[1]: Entering directory 'C:/ti/ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2/single_chip_servo_am243x-lp_system_freertos_nortos'
 Boot multi-core image: Release/single_chip_servo_system.appimage ...
C:\ti\ccs1200\ccs/tools/node/node C:/ti/mcu_plus_sdk_am243x_08_04_00_17/tools/boot/multicoreImageGen/multicoreImageGen.js --devID 55 --out Release/single_chip_servo_system.appimage ../single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-
clang/Release/single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang.rprc@4 ../single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang.rprc@5 ../single_chip_servo_am243x-lp
clang/Release/single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang.rprc@6 ../single_chip_servo_am243x-lp_r5fss1-1_nortos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss1-1_nortos_ti-arm-clang.rprc@7  >> Release/temp_stdout_Rele
makefile_system_ccs_bootimage_gen:42: recipe for target 'all' failed
File ../single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang.rprc does not exist
gmake[1]: *** [all] Error 1
```

Figure 4-8. Connect to R5F_0_0



ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2 - single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang/main.c - Code Composer Studio

```
AM243x_LP_no_launch.ccxml [Code Composer Studio - Device Debugging]
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_0_0 (Suspended - SW Breakpoint)
    main() at main.c:41 0x70155790
    bypass_auto_init + 0x4 () [C:/ti/ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2\single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-
    0xE3400020
    0xBADDBADD (no symbols are defined)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_0_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_1_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_1_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/DMSC_Cortex_M3_0 (Disconnected : Unknown)
```

```c
void single_chip_servo_main(void *args);

int main()
{
    //CSL_IcssCfgRegs *pIcssCfgRegs;
    uint32_t regVal;

    System_init();

#if defined(SINGLE_AXLE_USE_M1)
    /* Set EnDat ICSSG to 300MHz */
    //DebugP_assertNoLog(SOC_moduleSetClockFrequency(TISCI_DEV_PRU_ICSSG0,TISCI_DEV_PRU_ICSSG0_CORE_CLK,300000000) == SystemP_SUCCESS);
    /* Select ICSSG internal IEP clock source (IEP also running at 300MHz) */
    pIcssCfgRegs = (CSL_IcssCfgRegs *)CSL_PRU_ICSSG0_PR1_CFG_SLV_BASE;
    regVal = HW_RD_REG32(&pIcssCfgRegs->IEPCLK);
    regVal &= ~CSL_ICSSCFG_IEPCLK_OCP_EN_MASK;
    regVal |= (1 << CSL_ICSSCFG_IEPCLK_OCP_EN_SHIFT) & CSL_ICSSCFG_IEPCLK_OCP_EN_MASK;
    HW_WR_REG32(&pIcssCfgRegs->IEPCLK, regVal);
#endif

    Board_init();

    single_chip_servo_main(NULL);

    Board_deinit();
    System_deinit();

    return 0;
}
```

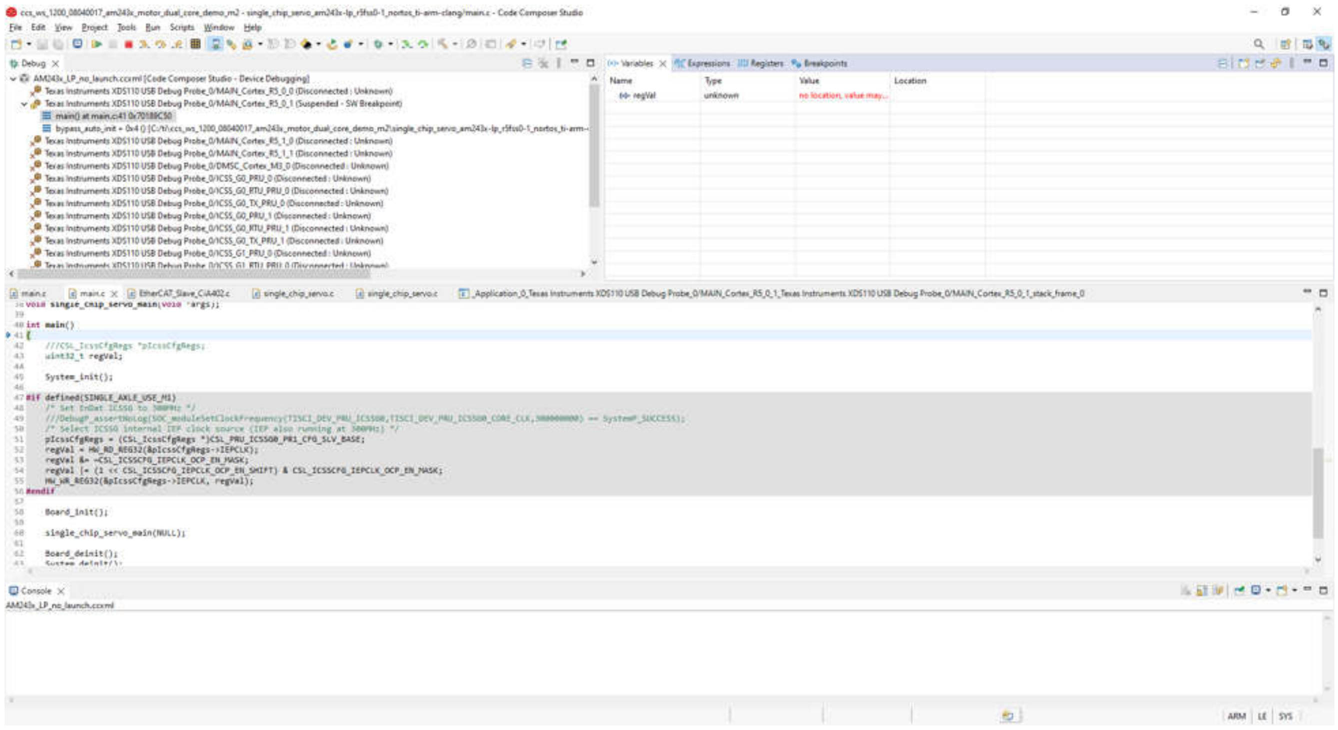
CDT Build Console [single_chip_servo_am243x-lp_system_freertos_nortos]

```
C:/ti/ccs1200/ccs/utils/bin/gmake -C C:\ti\ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2\single_chip_servo_am243x-lp_system_freertos_nortos -f makefile_system_ccs_bootimage_gen OUTNAME=single_chip_servo_system PROFILE=Release
MCU_PLUS_SDK_PATH=C:/ti/mcu_plus_sdk_am243x_08_04_00_17 CG_TOOL_ROOT=C:/ti/ccs1200/ccs/tools/compiler/ti-cgt-armllvm_2.1.0.LTS CCS_INSTALL_DIR=C:\ti\ccs1200\ccs CCS_IDE_MODE=desktop
gmake[1]: Entering directory 'C:/ti/ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2/single_chip_servo_am243x-lp_system_freertos_nortos'
 Boot multi-core image: Release/single_chip_servo_system.appimage ...
C:\ti\ccs1200\ccs/tools/node/node C:/ti/mcu_plus_sdk_am243x_08_04_00_17/tools/boot/multicoreImageGen/multicoreImageGen.js --devID 55 --out Release/single_chip_servo_system.appimage ../single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-
clang/Release/single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang.rprc@4 ../single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang.rprc@5 ../single_chip_servo_am243x-1p
clang/Release/single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang.rprc@6 ../single_chip_servo_am243x-lp_r5fss1-1_nortos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss1-1_nortos_ti-arm-clang.rprc@7  >> Release/temp_stdout_Rele
makefile_system_ccs_bootimage_gen:62: recipe for target 'all' failed
File ../single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang.rprc does not exist
gmake[1]: *** [all] Error 1
```

Updates Available
Updates are available for your software. Click to review and install updates.
Set up Reminder options

Figure 4-9. Load and Run the R5F_0_0

3. Load and Run Motor Control 2 – R5F_0_1
   - Halt R5F_0_1
   - Load and run single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang
   - Motor 2 is supposed to start spinning at 120 RPM



ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2 - single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang/main.c - Code Composer Studio

```
AM243x_LP_no_launch.ccxml [Code Composer Studio - Device Debugging]
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_0_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_0_1 (Suspended - SW Breakpoint)
    main() at main.c:41 0x70189C50
    bypass_auto_init + 0x4 () [C:/ti/ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2\single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_1_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/MAIN_Cortex_R5_1_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/DMSC_Cortex_M3_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_PRU_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_RTU_PRU_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_TX_PRU_0 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_PRU_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_RTU_PRU_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G0_TX_PRU_1 (Disconnected : Unknown)
  Texas Instruments XDS110 USB Debug Probe_0/ICSS_G1_PRU_0 (Disconnected : Unknown)
```

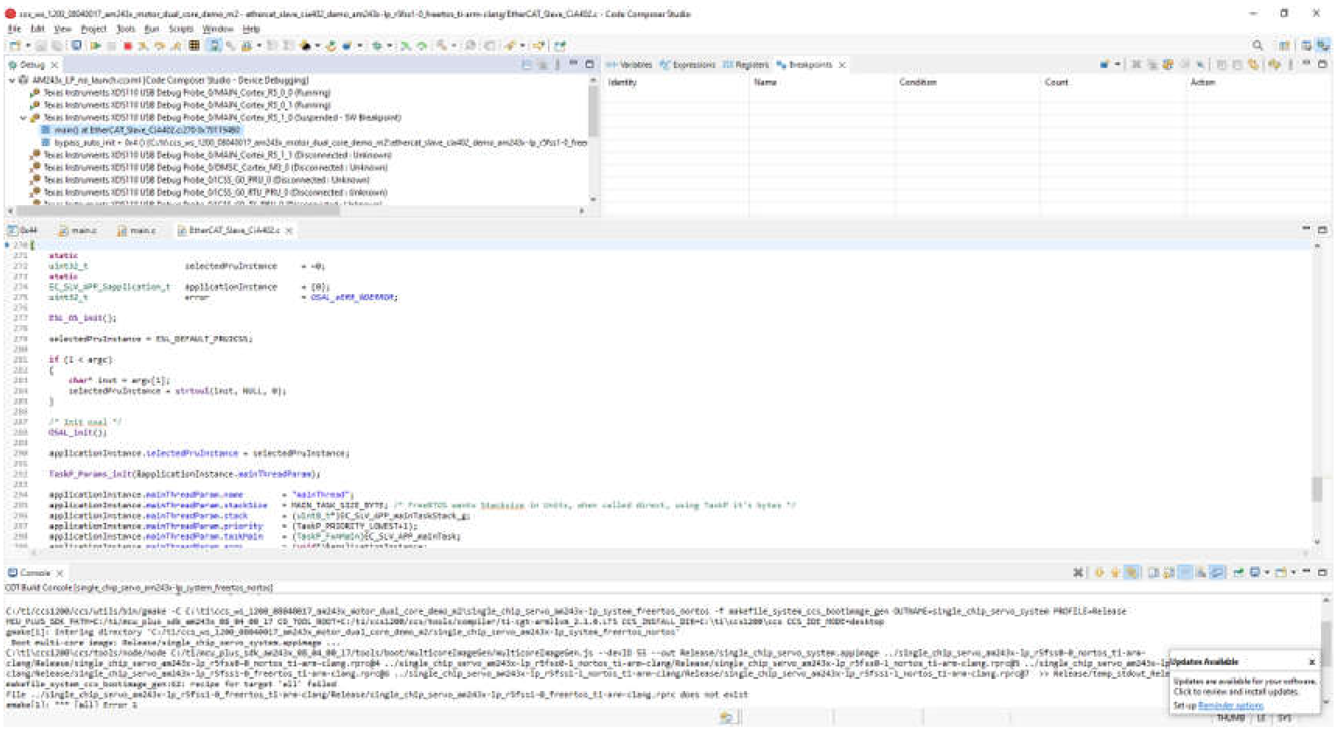
Name	Type	Value	Location
regVal	unknown	no location, value may...	

```c
void single_chip_servo_main(void *args);

int main()
{
    //CSL_IcssCfgRegs *pIcssCfgRegs;
    uint32_t regVal;

    System_init();

#if defined(SINGLE_AXLE_USE_M1)
    /* Set EnDat ICSSG to 300MHz */
    //DebugP_assertNoLog(SOC_moduleSetClockFrequency(TISCI_DEV_PRU_ICSSG0,TISCI_DEV_PRU_ICSSG0_CORE_CLK,300000000) == SystemP_SUCCESS);
    /* Select ICSSG internal IEP clock source (IEP also running at 300MHz) */
    pIcssCfgRegs = (CSL_IcssCfgRegs *)CSL_PRU_ICSSG0_PR1_CFG_SLV_BASE;
    regVal = HW_RD_REG32(&pIcssCfgRegs->IEPCLK);
    regVal &= ~CSL_ICSSCFG_IEPCLK_OCP_EN_MASK;
    regVal |= (1 << CSL_ICSSCFG_IEPCLK_OCP_EN_SHIFT) & CSL_ICSSCFG_IEPCLK_OCP_EN_MASK;
    HW_WR_REG32(&pIcssCfgRegs->IEPCLK, regVal);
#endif

    Board_init();

    single_chip_servo_main(NULL);

    Board_deinit();
    System_deinit();
```

Console
AM243x_LP_no_launch.ccxml

Figure 4-10. Load and Run the R5F_0_1

4. Load and Run EtherCat CiA402 Client – R5F_1_0
   - Halt R5F_1_0
   - Load and run `ethercat_slave_cia402_demo_am243x-lp_r5fss1-0_freertos_ti-arm-clang`
   - The EtherCat CiA402 client device is now ready to be detected by TwinCAT (PLC)



```
{
    static
    uint32_t                    selectedPruInstance     = ~0;
    static
    EC_SLV_APP_Sapplication_t   applicationInstance     = {0};
    uint32_t                    error                   = OSAL_eERR_NOERROR;

    ESL_OS_init();

    selectedPruInstance = ESL_DEFAULT_PRUICSS;

    if (1 < argc)
    {
        char* inst = argv[1];
        selectedPruInstance = strtoul(inst, NULL, 0);
    }

    /* Init osal */
    OSAL_init();

    applicationInstance.selectedPruInstance = selectedPruInstance;

    TaskP_Params_init(&applicationInstance.mainThreadParam);

    applicationInstance.mainThreadParam.name        = "mainThread";
    applicationInstance.mainThreadParam.stackSize   = MAIN_TASK_SIZE_BYTE; /* FreeRTOS wants Stacksize in Units, when called direct, using TaskP it's bytes */
    applicationInstance.mainThreadParam.stack       = (uint8_t*)EC_SLV_APP_mainTaskStack_g;
    applicationInstance.mainThreadParam.priority    = (TaskP_PRIORITY_LOWEST+1);
    applicationInstance.mainThreadParam.taskMain    = (TaskP_FxnMain)EC_SLV_APP_mainTask;
```

```
CDT Build Console [single_chip_servo_am243x-lp_system_freertos_nortos]

C:/ti/ccs1200/ccs/utils/bin/gmake -C C:\ti\ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2\single_chip_servo_am243x-lp_system_freertos_nortos -f makefile_system_ccs_bootimage_gen OUTNAME=single_chip_servo_system PROFILE=Release
MCU_PLUS_SDK_PATH=C:/ti/mcu_plus_sdk_am243x_08_04_00_17 CG_TOOL_ROOT=C:/ti/ccs1200/ccs/tools/compiler/ti-cgt-armllvm_2.1.0.LTS CCS_INSTALL_DIR=C:\ti\ccs1200\ccs CCS_IDE_MODE=desktop
gmake[1]: Entering directory 'C:/ti/ccs_ws_1200_08040017_am243x_motor_dual_core_demo_m2/single_chip_servo_am243x-lp_system_freertos_nortos'
 Boot multi-core image: Release/single_chip_servo_system.appimage ...
C:\ti\ccs1200\ccs/tools/node/node C:/ti/mcu_plus_sdk_am243x_08_04_00_17/tools/boot/multicoreImageGen/multicoreImageGen.js --devID 55 --out Release/single_chip_servo_system.appimage ../single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss0-0_nortos_ti-arm-clang.rprc@4 ../single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss0-1_nortos_ti-arm-clang.rprc@5 ../single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang.rprc@6 ../single_chip_servo_am243x-lp_r5fss1-1_nortos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss1-1_nortos_ti-arm-clang.rprc@7  >> Release/temp_stdout_Rele
makefile_system_ccs_bootimage_gen:62: recipe for target 'all' failed
File ../single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang/Release/single_chip_servo_am243x-lp_r5fss1-0_freertos_ti-arm-clang.rprc does not exist
gmake[1]: *** [all] Error 1
```

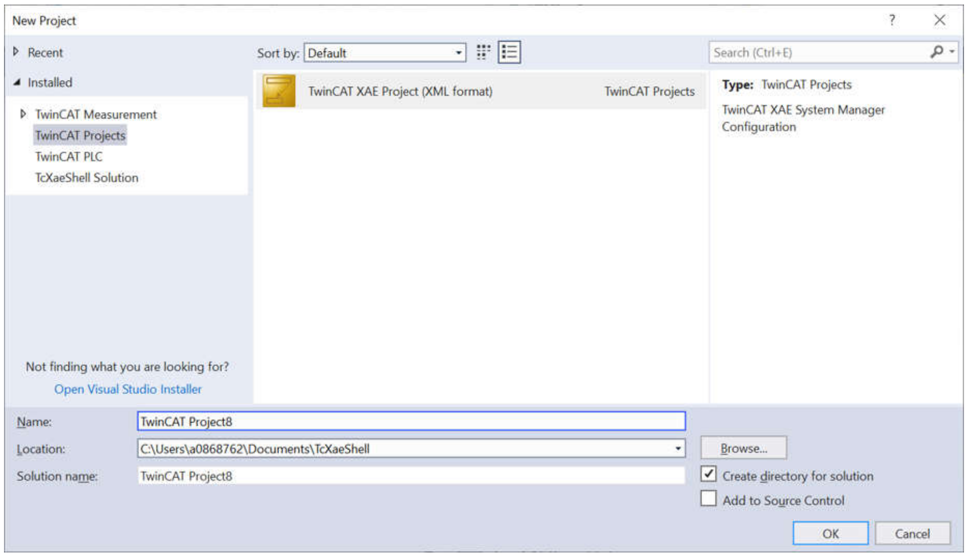
Updates Available
Updates are available for your software. Click to review and install updates.
Set up Reminder options

Figure 4-11. Load and Run the R5F_1_0

4.4 Test Results

To evaluate the reference design, complete the following steps:

1. Download and install the TwinCAT on your Windows PC
2. Launch the TwinCAT automation software
3. Create an EtherCAT Project as shown in the TwinCAT software GUI:



Type: TwinCAT Projects
TwinCAT XAE System Manager Configuration

Name: TwinCAT Project8
Location: C:\Users\a0868762\Documents\TcXaeShell
Solution name: TwinCAT Project8
[✓] Create directory for solution
[ ] Add to Source Control

Figure 4-12. Creating an EtherCAT® Project in TwinCAT

4. EtherCAT CiA402 – Scan the device by right clicking on *Devices* → *Scan* …). Use the following images to step through the process.

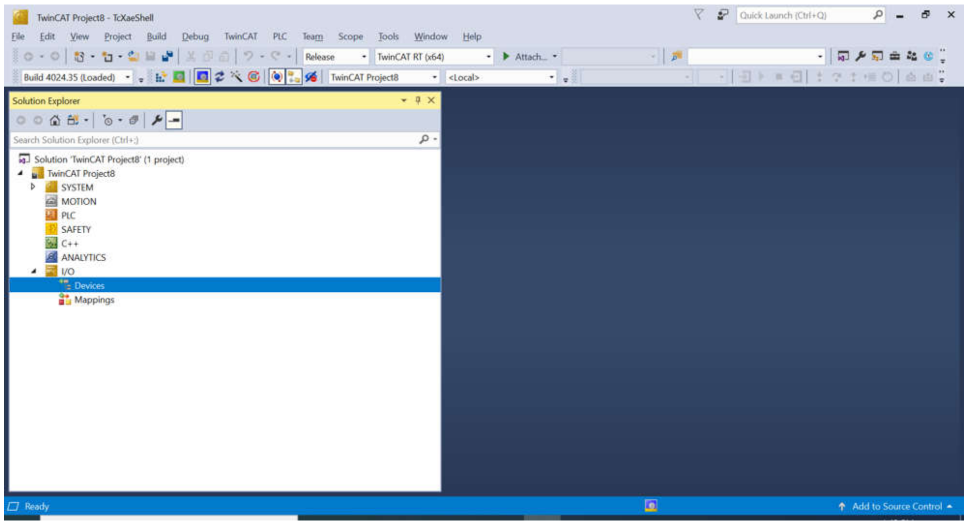
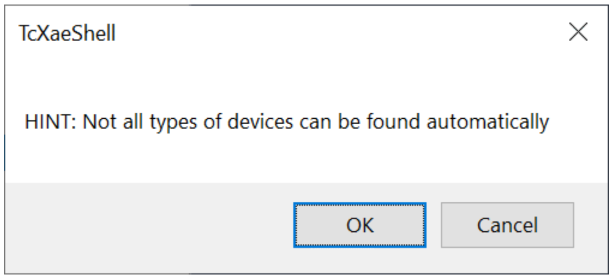
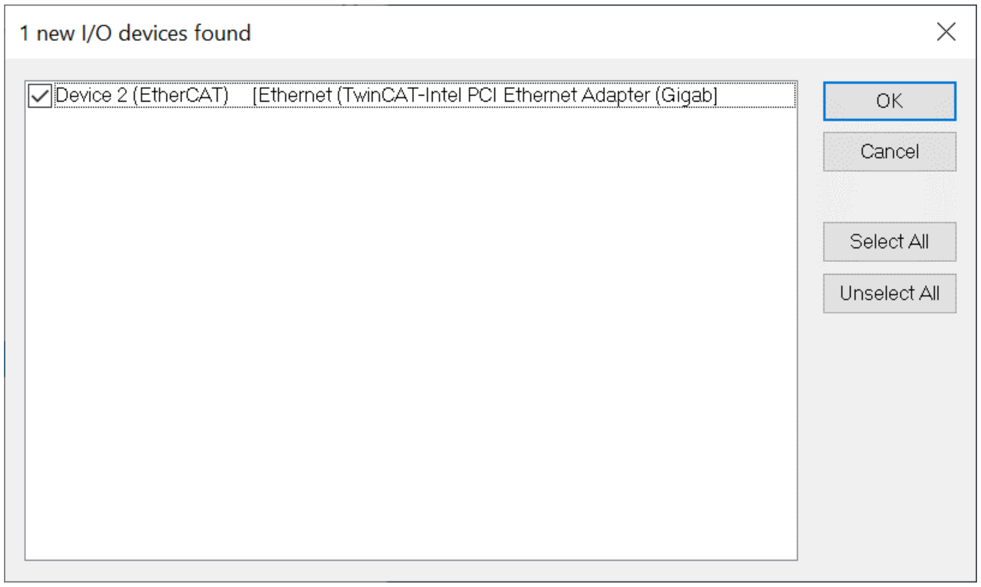


Figure 4-13. Scan for EtherCAT® device in TwinCAT


TcXaeShell

HINT: Not all types of devices can be found automatically

Figure 4-14. Scan for EtherCAT® device in TwinCAT (2)


1 new I/O devices found

☑ Device 2 (EtherCAT) [Ethernet (TwinCAT-Intel PCI Ethernet Adapter (Gigab)]

Figure 4-15. Scan for EtherCAT® device in TwinCAT (3)

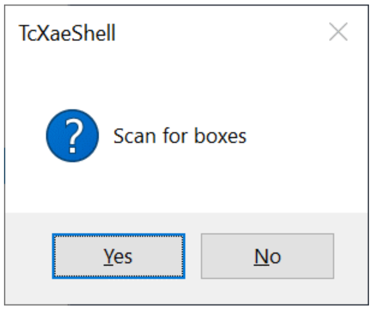
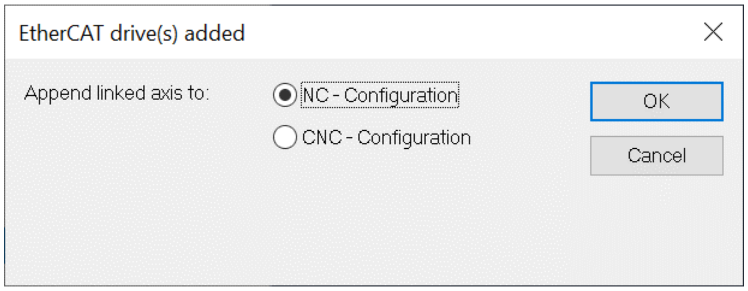


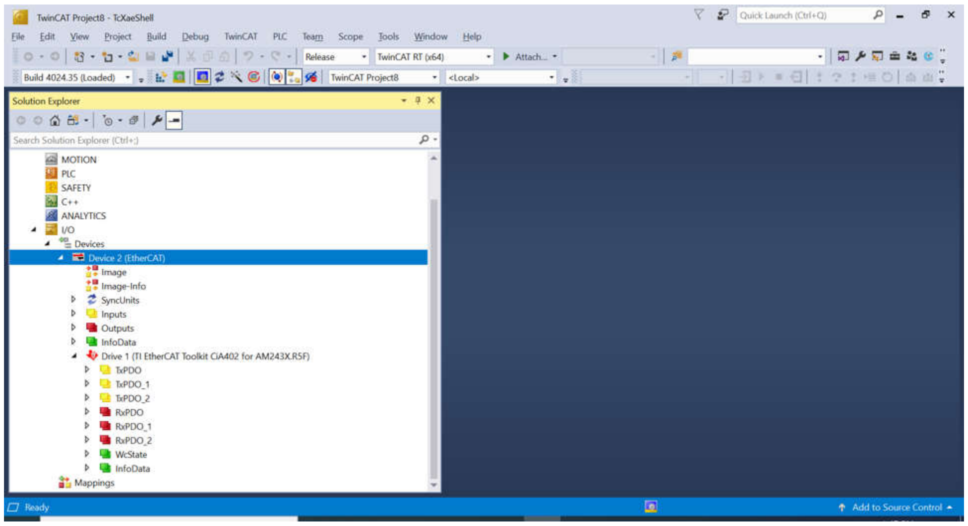
Figure 4-16. Scan for EtherCAT® device in TwinCAT (4)


EtherCAT drive(s) added

Append linked axis to: [✓ NC - Configuration] CNC - Configuration

Figure 4-17. Scan for EtherCAT® device in TwinCAT (5)

5. EtherCAT CiA402 – Device 1 (TI EtherCAT Toolkit CiA402 for AM243X.R5F) Found


TwinCAT Project8 - TcXaeShell

Build 4024.35 (Loaded)

Release

TwinCAT RT (x64)

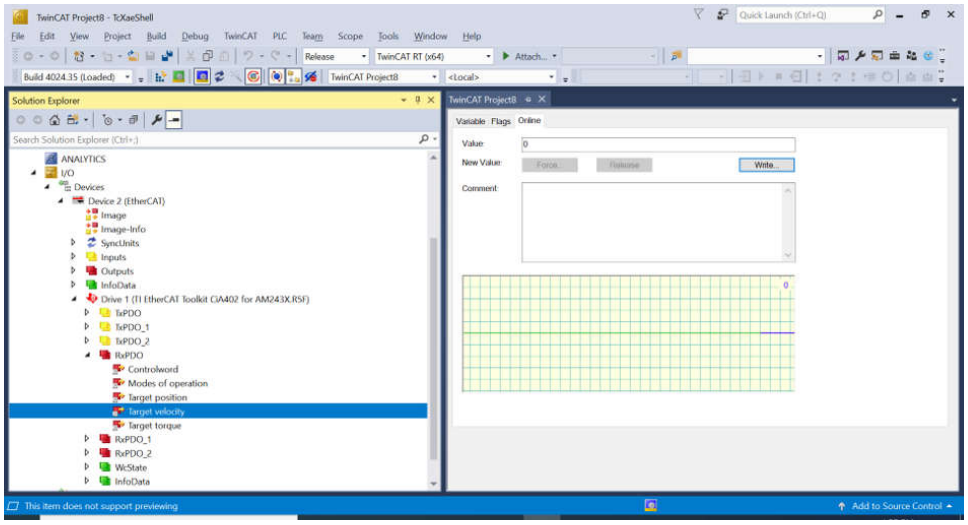
TwinCAT Project8

```
MOTION
PLC
SAFETY
C++
ANALYTICS
I/O
  Devices
    Device 2 (EtherCAT)
      Image
      Image-Info
      SyncUnits
      Inputs
      Outputs
      InfoData
      Drive 1 (TI EtherCAT Toolkit CiA402 for AM243X.R5F)
        TxPDO
        TxPDO_1
        TxPDO_2
        RxPDO
        RxPDO_1
        RxPDO_2
        WcState
        InfoData
  Mappings
```

Ready

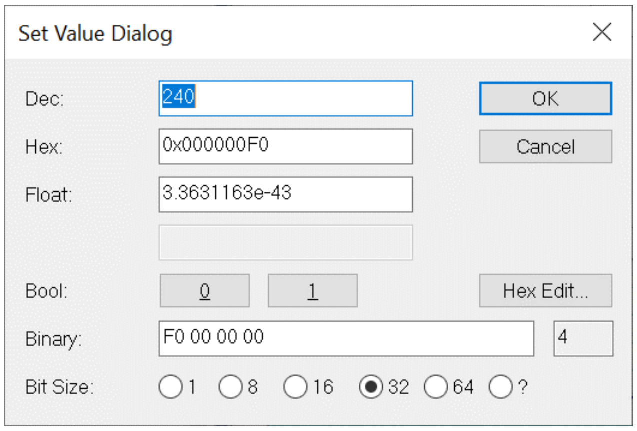
Figure 4-18. An EtherCAT® device is found by TwinCAT

6. EtherCAT CiA402 – Change RxPDO (Motor 1) *Target velocity* to 240 (240 RPM)



Value: 0

Figure 4-19. Change the Target Velocity for Motor 1 in TwinCAT


Set Value Dialog

Dec: 240

Hex: 0x000000F0

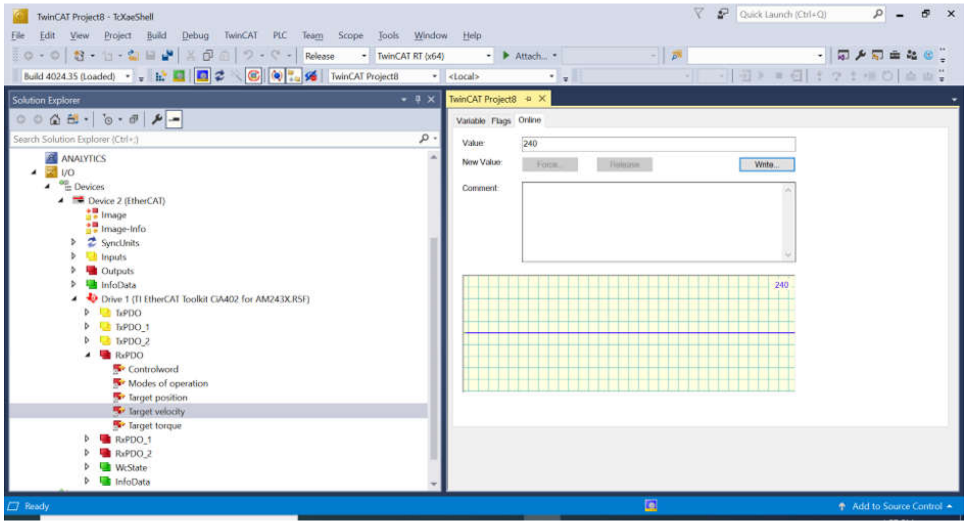
Float: 3.3631163e-43

Bool: 0 1

Binary: F0 00 00 00 4

Bit Size: 1 8 16 [✓ 32] 64 ?

Figure 4-20. Change the Target Velocity for Motor 1 in TwinCAT (2)



Value: 240

Figure 4-21. Change the Target Velocity for Motor 1 in TwinCAT (3)

7. EtherCAT CiA402 – Change RxPDO (Motor 1) *Modes of Operation* to "9" (*Cyclic synchronous velocity mode*)

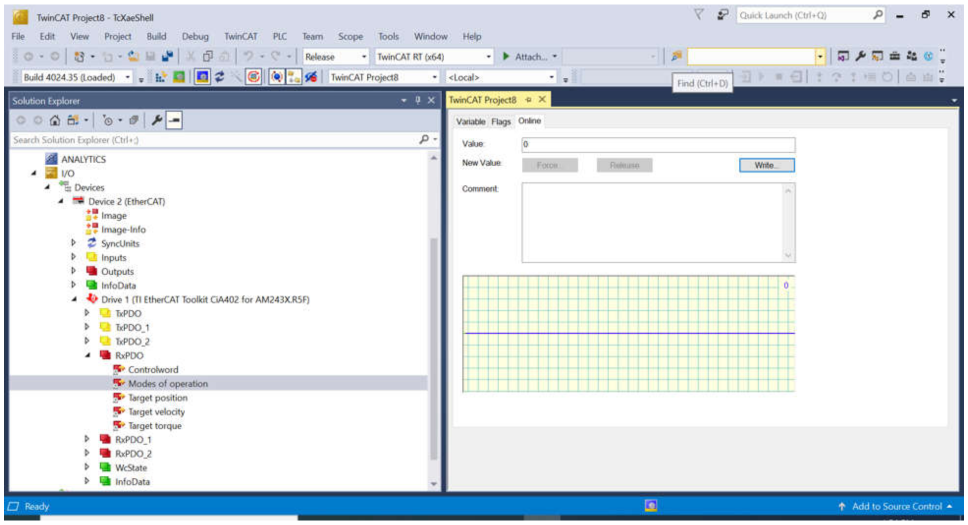
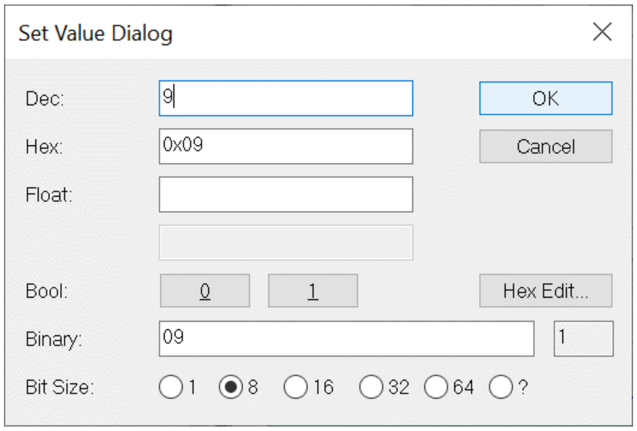


Figure 4-22. Change the Operation Mode for Motor 1 in TwinCAT


Set Value Dialog

Dec: 9

Hex: 0x09

Float:

Bool: 0 1

Binary: 09 1

Bit Size: ○ 1 [● 8] ○ 16 ○ 32 ○ 64 ○ ?

Figure 4-23. Change the Operation Mode for Motor 1 in TwinCAT (2)

8. EtherCAT CiA402 – Change RxPDO (Motor 1) *Controlword* to "15" (*Switch On* | *Enable Voltage* | *Quick Stop* | *Enable Operation*)

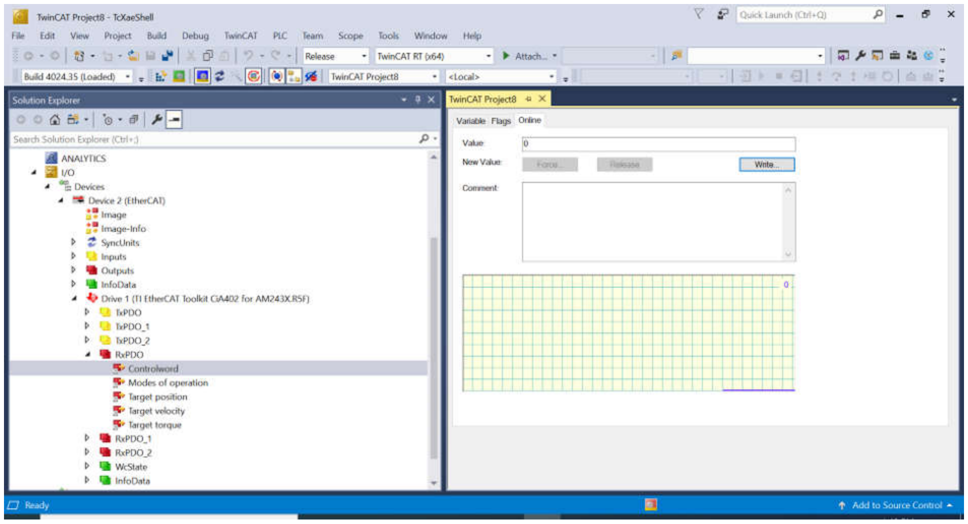
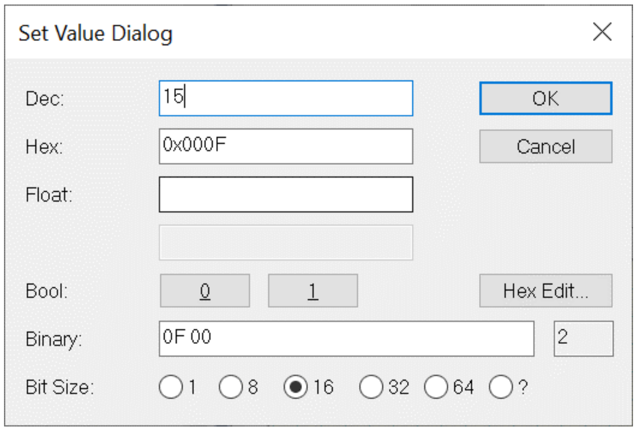


Figure 4-24. Change the Controlword for Motor 1 in TwinCAT


Set Value Dialog

Dec: 15

Hex: 0x000F

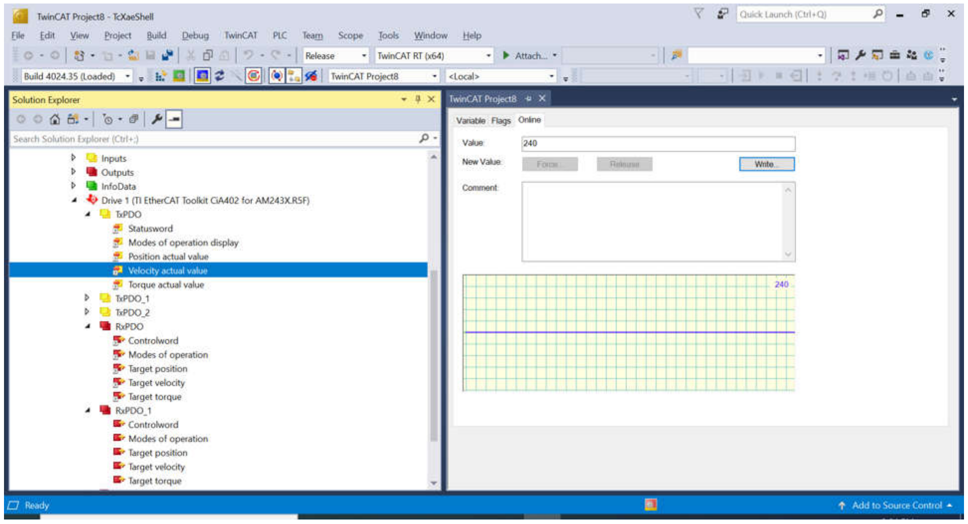
Float:

Binary: 0F 00 | 2

Bit Size: ( ) 1 ( ) 8 (● 16) ( ) 32 ( ) 64 ( ) ?

Figure 4-25. Change the Controlword for Motor 1 in TwinCAT (2)

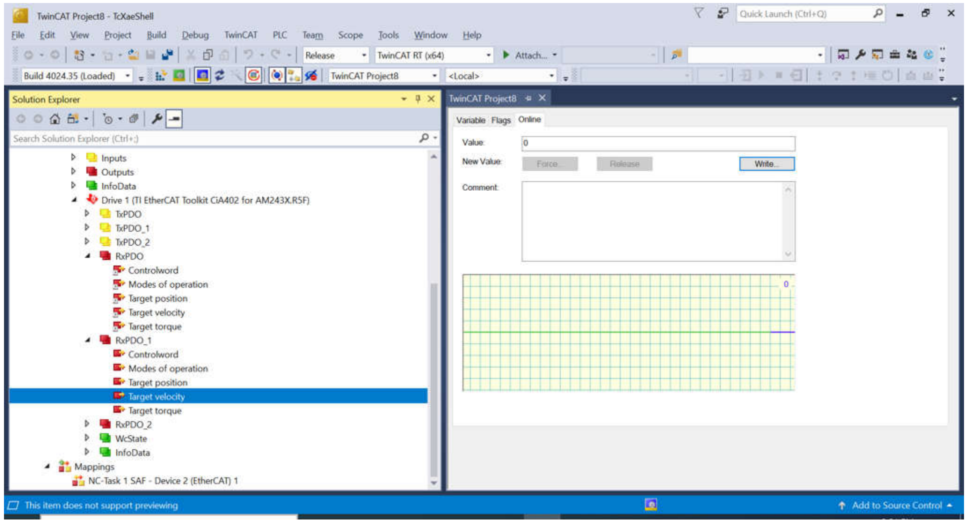
9. After changing this setting, the speed of Motor 1 changes from 120 RPM to 240 RPM
10. EtherCAT CiA402 – Check TxPDO (Motor 1), make sure *Velocity actual value* is 240 (240 RPM)



Value: 240

Figure 4-26. Check the Actual Velocity for Motor 1 in TwinCAT

11. EtherCAT CiA402 – Change RxPDO_1 (Motor 2) *Target velocity* to 180 (180 RPM)



Value: 0

Figure 4-27. Change the Target Velocity for Motor 2 in TwinCAT

12. EtherCAT CiA402 – Change RxPDO_1 (Motor 2) *Modes of Operation* to "9" (*Cyclic synchronous velocity mode*)

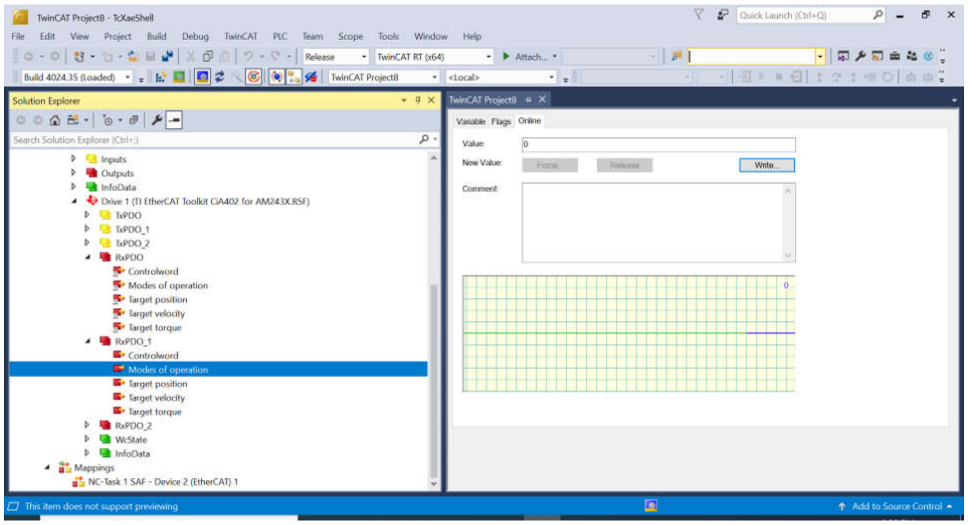


Figure 4-28. Change the Operation Mode for Motor 2 in TwinCAT

13. EtherCAT CiA402 – Change RxPDO_1 (Motor 2) *Controlword* to "15" (*Switch On* | *Enable Voltage* | *Quick Stop* | *Enable Operation*)

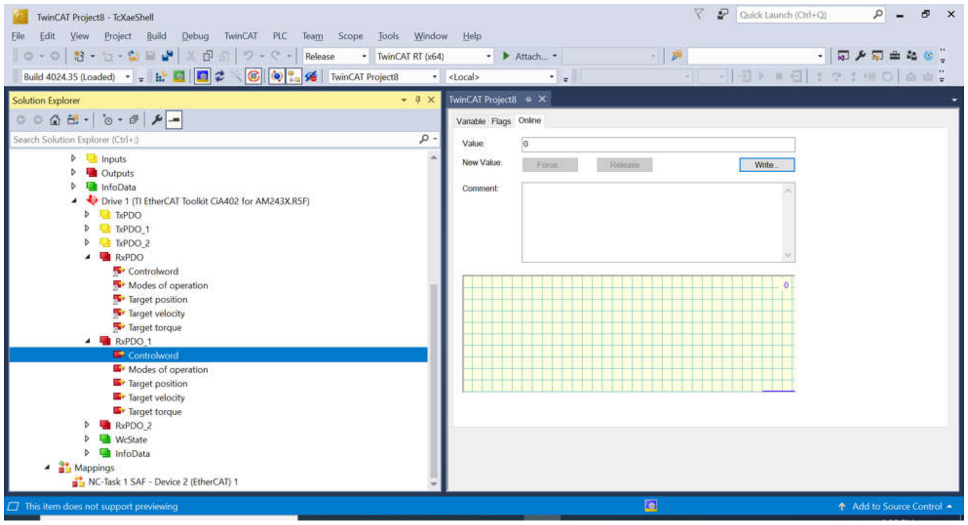


Figure 4-29. Change the Controlword for Motor 2 in TwinCAT

14. After the previous change, the target speed of motor 2 is 180 RPM

15. EtherCAT CiA402 – Check TxPDO1 (Motor 2), make sure *Velocity actual value* is 180 (180 RPM)

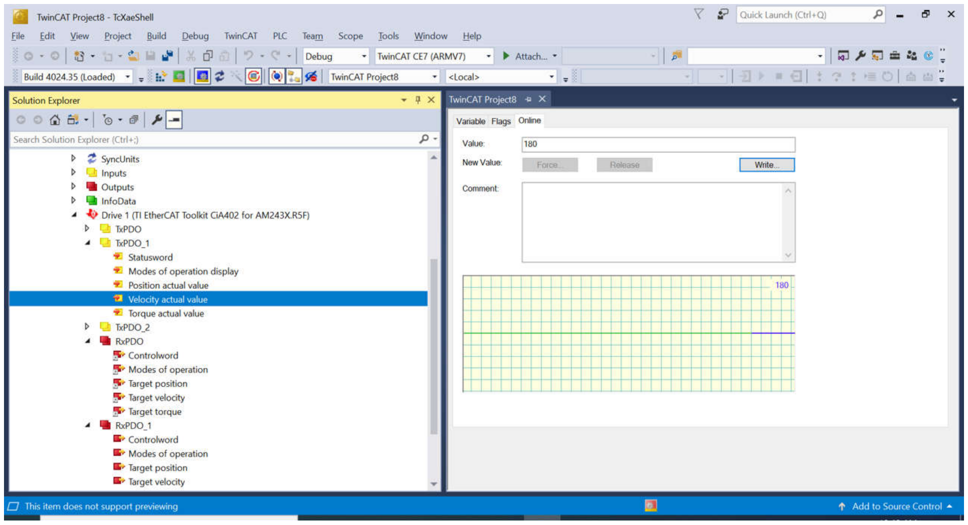


Figure 4-30. Check the Actual Velocity for Motor 2 in TwinCAT

5 Design and Documentation Support

5.1 Design Files

5.1.1 Schematics

There are two schematics related to this reference design:

To download the BLDC BoosterPack schematics, see the design files at BP-AM2BLDCSERVO Design Package.

To download the AM243x LaunchPad schematics, see the design files at LP-AM243 Design Package.

5.1.2 BOM

To download the bill of materials (BOM) for BLDC BP, see the design files at BP-AM2BLDCSERVO Design Package page

To download the bill of materials (BOM) for AM243x LP, see the design files at LP-AM243 Design Package

5.2 Tools and Software

Tools

CCSTUDIO	Code Composer Studio™ integrated development environment (IDE): download CCS 12.5.0 Version for Windows or Linux
ARM-CGT-CLANG	Arm® code generation tools - compiler: download TI ARM CLANG 3.2.0.LTS for Windows or Linux
SYSCONFIG	Standalone desktop version of SysConfig: download SysConfig 1.18.0 for Windows or Linux

Software

AM243x Motor Control SDK	Motor Control SDK Windows Installer
AM243x Industrial Communication SDK	Industrial Communications SDK Windows Installer
AM243x MCU+ SDK	MCU PLUS SDK Windows Installer

5.3 Documentation Support

1. Texas Instruments, *AM64x /AM243x Processors Silicon* Technical Reference Manual
2. Texas Instruments, *AM2x BLDC Servo Motor BoosterPack (BPAM2BLDCSERVO)* EVM User's Guide

5.4 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

5.5 Trademarks

LaunchPad™, TI E2E™, and BoosterPack™, and are trademarks of Texas Instruments.
EtherCAT® is a registered trademark of Beckhoff Automation GmbH.
Microsoft® and Windows® are registered trademarks of Microsoft Corporation.
All trademarks are the property of their respective owners.

6 About the Author

MING WEI (MGTS) is a senior software engineer at Sitara MCU, where he develops and supports the Processor SDK RTOS/MCU+ SDK/Motor Control SDK for Sitara MPU/MCU and the DSP families of SOC devices. Ming brings his extensive experiences and knowledge in motor control, real-time systems, signal processing, and code optimization to this role. Ming earned B.Sc., M.Sc., and Ph.D. in computer sciences from Xi’an Jiao-tong University and University of North Texas respectively.