Application Note

How to Synchronize the Timing Between Chips With Programmable Real-Time Unit

Chen Gao, Sabari Kannan Muthalagu

ABSTRACT

In industrial motor drive system, multiple devices or chips are often required to communicate with each other in a quick speed, low-latency and synchronized manner. One typical application example is closed-loop control with two different Micro-programmed Control Unit (MCU). To receive and transmit data in every control cycle, the communication timing needs to be deterministic and synchronous.

The Industrial Ethernet Peripheral (IEP) Module which is part of the Programmable Real-time Unit and Industrial Communication Subsystem (PRU_ICSS) features an industrial Ethernet timer to perform hardware work for real-time control.

This application note describes how to use IEP timer and PRU cores to perform time synchronization between chips and transmitting data with configurable timing under control cycle via Fast Serial Interface (FSI) for closed-loop motor control.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

Copyright © 2025 Texas Instruments Incorporated

1 Introduction of AC or Servo Drive Hot-Side Control Architecture

The primary entities of an industrial motor drive are motor control, industrial communication and application to manage control and communication functions.

- The motor control portion includes the Field-oriented Control (FOC) algorithm, the motor Pulse-Width Modulation (PWM) controller and the motor current and position feedback system components which provide the normal operation for the motor.
- The industrial communication interface provides the real-time communication link and synchronized system timing with the field network.
- The application provides the high-level control of the drive managing the overall drive communications and control functions. This can include configuration, startup or shutdown, status, operation, motion control and other management functions.

These three components of the industrial drive architecture can further be implemented in a variety of configurations. One of the architectures as shown in Figure 1-1 is a two-chip design with hot-side motor control which provides a single or multi-axis drive with multi-protocol field bus and real-time Ethernet, advanced application algorithms and control performance. The hot-side Micro-programmed Control Unit (MCU) is implemented to perform FOC control with a constant cycle time. A part of the cold-side MCU is implemented to decode angle and velocity information from the motor encoder. To receive the motor angle and velocity information in every cycle time, the timing needs to be deterministic and pre-configured to match the FOC control loop scheme.

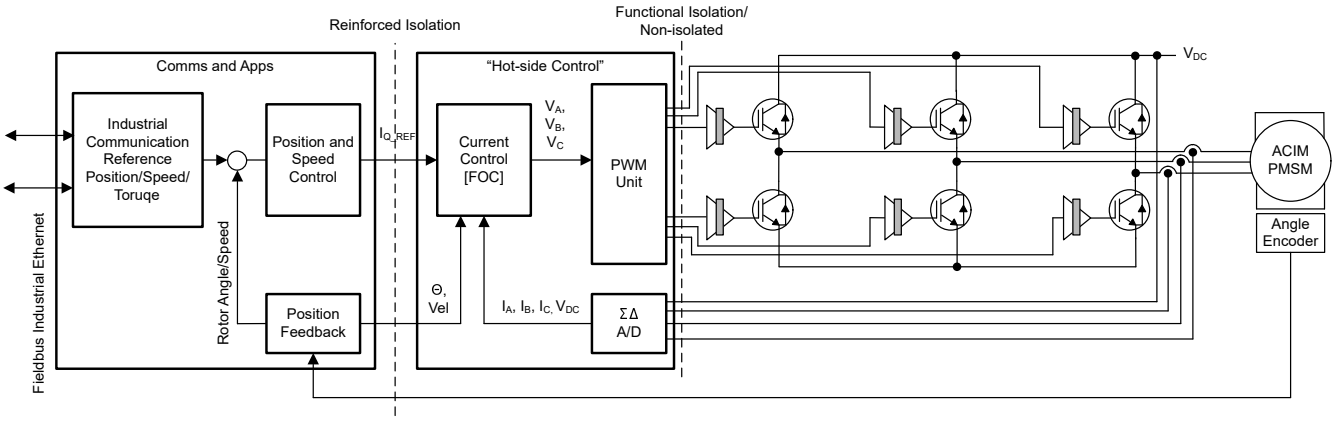

Figure 1-1. Two-Chip Design With Hot-Side Motor Control Architecture

Copyright © 2025 Texas Instruments Incorporated

2 PRU and FSI Implementation for Time Synchronization and Data Transmitting

2.1 Importance of Clock in Industrial Systems With MCUs

Clock plays a fundamental role in industrial systems that incorporate microcontroller units (MCUs), as the clock directly influences system performance. The execution speed of code within an MCU is directly proportional to the clock frequency, meaning that higher clock speeds result in faster processing and execution of instructions. Additionally, an increased clock frequency enhances data communication rates, enabling more efficient data transfer between interconnected devices.

Maintaining a stable clock frequency is critical for establishing reliable communication and accurate data exchange, especially in industrial applications where precision and timing are crucial. Many serial communication protocols depend on a consistent clock signal to facilitate error-free data transmission. However, as clock frequency increases, clock drift becomes more pronounced. The clock drift refers to the gradual deviation of a clock signal from the expected timing due to variations in oscillator stability, temperature fluctuations, and manufacturing tolerances. This drift can lead to synchronization issues, particularly in systems where multiple MCUs are deployed to perform coordinated tasks.

2.2 IEP Timer Interface

The IEP module features an industrial Ethernet timer with 16 compare events, industrial Ethernet sync generator and latch capture, industrial Ethernet watchdog timer, and a digital I/O port. The IEP functional block diagram is shown in Figure 2-1.

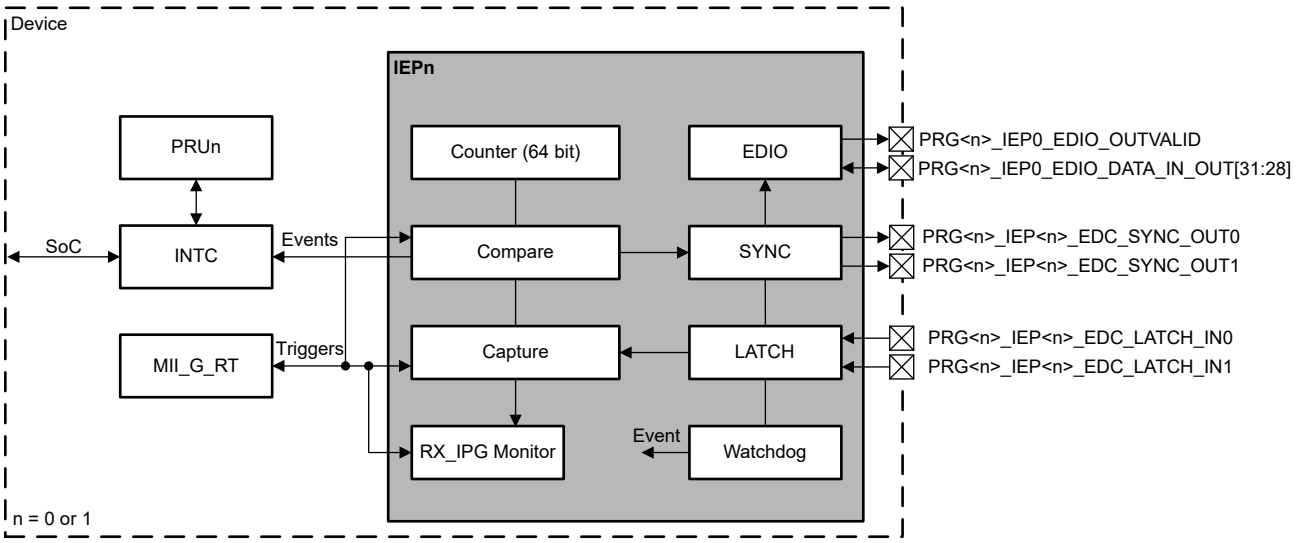

Figure 2-1. IEP Timer Functional Block Diagram

The sync block supports the generation of two synchronization signals: SYNC0 and SYNC1 which can be directly mapped to the output for external devices to use. The signals can also be used for internal synchronization within the PRU_ICSSG. The generation modes can be configured to four operation modes: cyclic mode, single shot mode, cyclic with acknowledge mode, and single shot with acknowledge mode as Figure 2-2 shown.

Copyright © 2025 Texas Instruments Incorporated

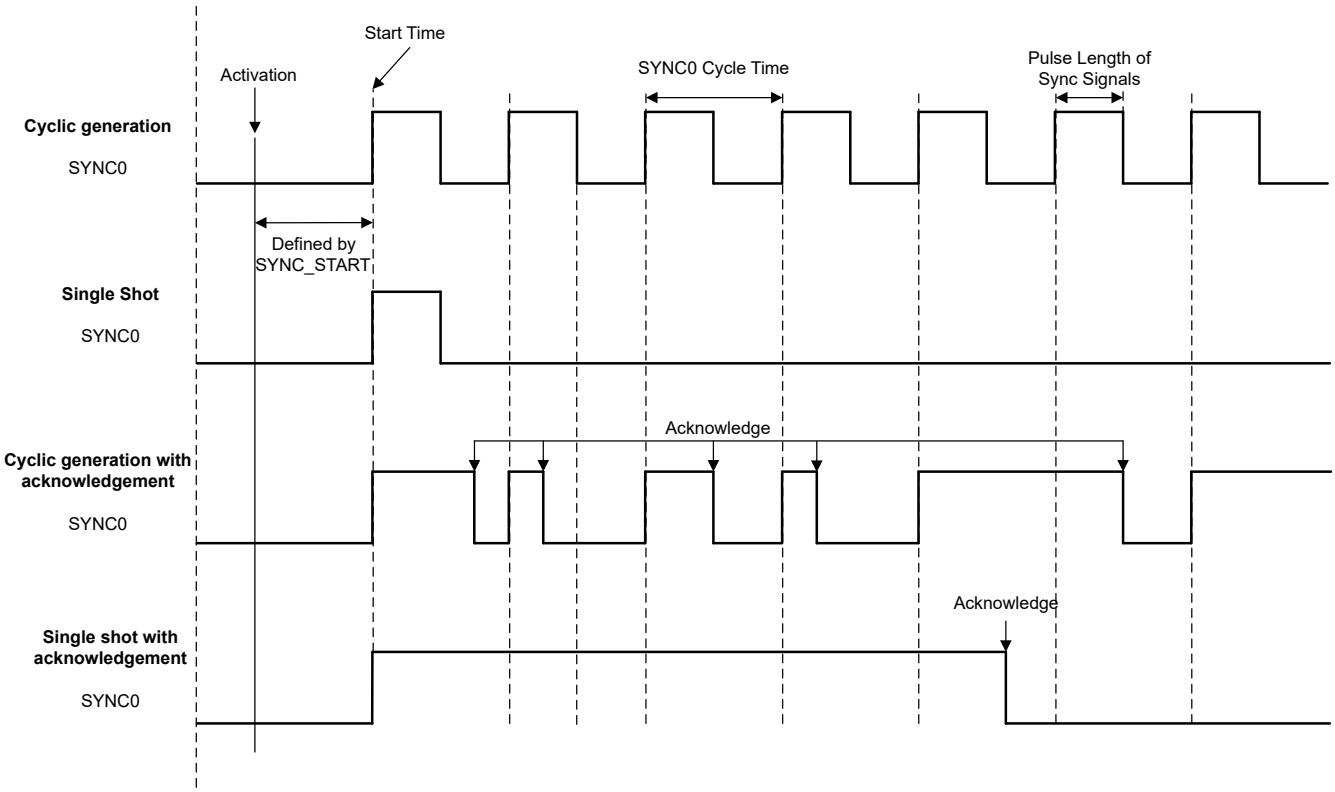

Figure 2-2. PRU_ICSSG IEP SYNC0 Signal Generation Modes

The time sync router (TSR) module is designed to send one sync signal to multiple recipients. This sync signal allows multiple peripherals or cores within the processor to synchronize the counters to a single *main clock*. TSR signals can even be routed to processor pins if the sync signal also needs to be received by a device external to the processor. The IEP latch signal is connected to the TSR output to establish a signal path from PRU GPIO to the ICSSG_IEP_Latch. *[FAQ] AM64x: What is the Time Sync Router for? How do I use it? - Processors forum - Processors - TI E2E support forums* is the E2E forum post link of the introduction of TSR module.

The controller can send cyclic SYNC0 according to the control loop frequency. And the device can capture the SYNC0 signal by IEP latch and capture modules then compensate the delay time and align the timestamping with the controller.

The IEP counters of both MCUs are synchronized using sync out signal, latch and IEP counter compensation register. Both MCUs are configured to operate the IEP count at the same frequency and with same increment value. In this setup of application note, the IEP clock frequency is set to 250MHz, with the default increment value of the IEP counter being 4.

2.3 PRU_ICSSG Task Manger

The dedicated task manager is integrated for each PRU core of the PRU_ICSSG system used for efficient switching between tasks. Each task manager works independently from the others. The task manager has two modes of operation that are general purpose mode and RX_TX mode for Ethernet purpose.

In this application, general purpose mode is used to improve the firmware efficiency and performance. This feature enables software to get preempted to do another higher priority task. The task manager can issue the hardware preemption and respond to the relative instruction by saving off the current program counter and flags and providing a new program counter (PC) to start a new task. When the firmware completes this new task, the firmware can terminate this task by issuing a dedicated instruction. When the task manager sees this execution of this instruction, the task manager can return the state of the PRU from the last saved off task. This hardware context switching block diagram is shown on Figure 2-3. The firmware is responsible for saving and restoring any internal registers that the firmware can override during a task. The data RAM or shared RAM can be used to store the register value. The multi tasks can be mapped and triggered by source event such as IEP compare

Copyright © 2025 Texas Instruments Incorporated

event, IEP capture event and interrupt controller (INTC) host event so that all the tasks timing is deterministic and pre-configured.

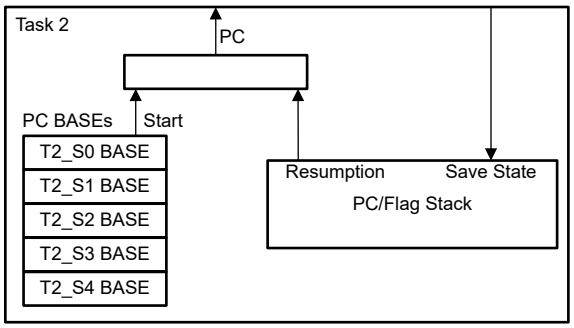

Figure 2-3. PRU_ICSSG Task Manager Hardware Context Switching Block Diagram

2.4 Fast Serial Interface

Fast Serial Interface (FSI) module is a serial communication peripheral capable of reliable high-speed communication across isolation devices. Galvanic isolation devices are used in situations where two different electronic circuits, which do not have common power and ground connections, must exchange information. Though isolation devices facilitate these signal communications, the devices can also introduce a large delay on the signal lines and add skew between the signals. The FSI is designed specifically to make sure reliable high-speed communication for system scenarios that involve communication across isolation barriers without adding components. The FSI consists of independent transmitter (FSITX) and receiver (FSIRX) cores. The FSITX and FSIRX cores are configured and operated independently. Figure 2-4 shows the high level FSI communication block diagram. With 50MHz clock and 2 data line, the communication speed of FSI can reach to 200Mbit/sec.

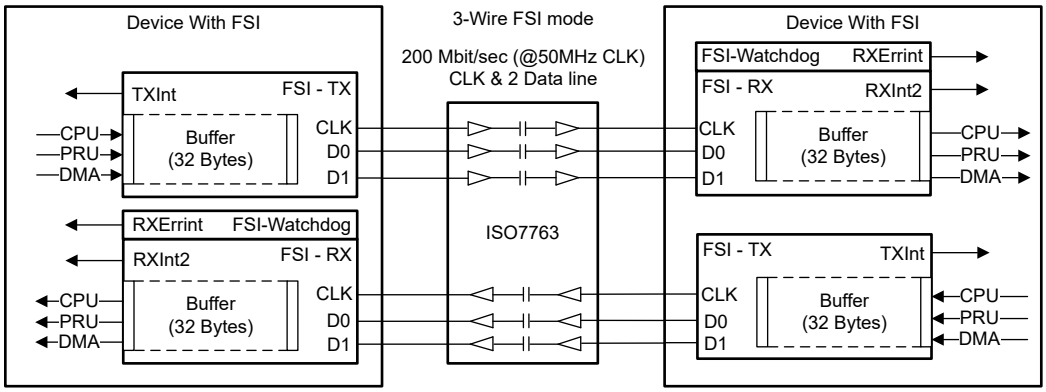

Figure 2-4. FSI High Level Communication Modules

Copyright © 2025 Texas Instruments Incorporated

2.5 Two-Chip System Scheme for Time Synchronization and Data Transmitting

Assuming that two AM243x devices with PRU_ICSSG are used for implementation for time synchronization and data transmitting. Figure 2-5 and Figure 2-6 show the system block diagram and timing for the time synchronization and data transmitting during control cycles.

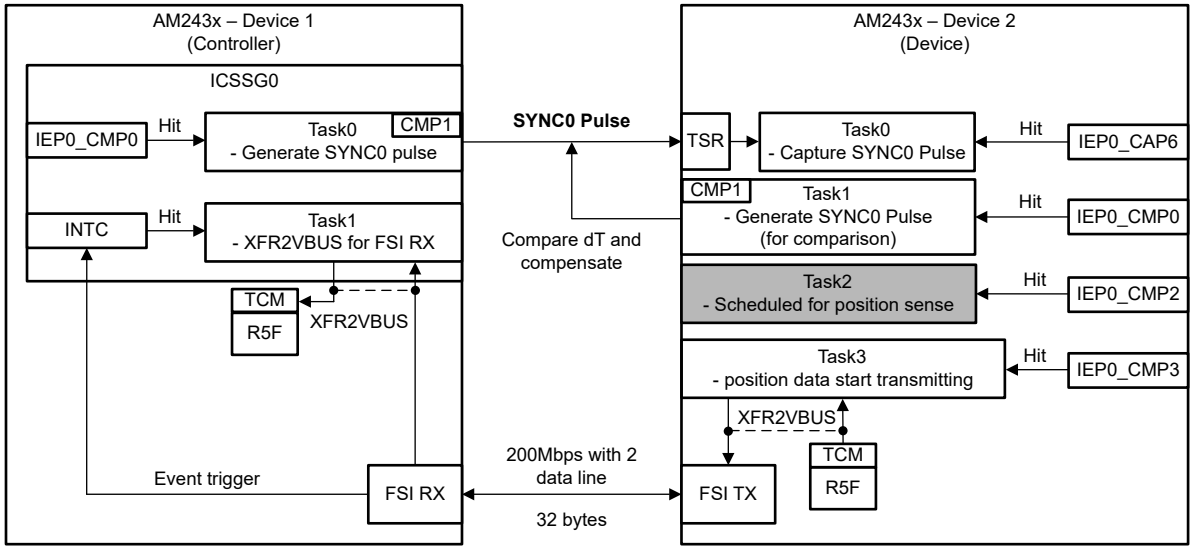

Figure 2-5. Time Synchronization and Data Transmitting Block Diagram

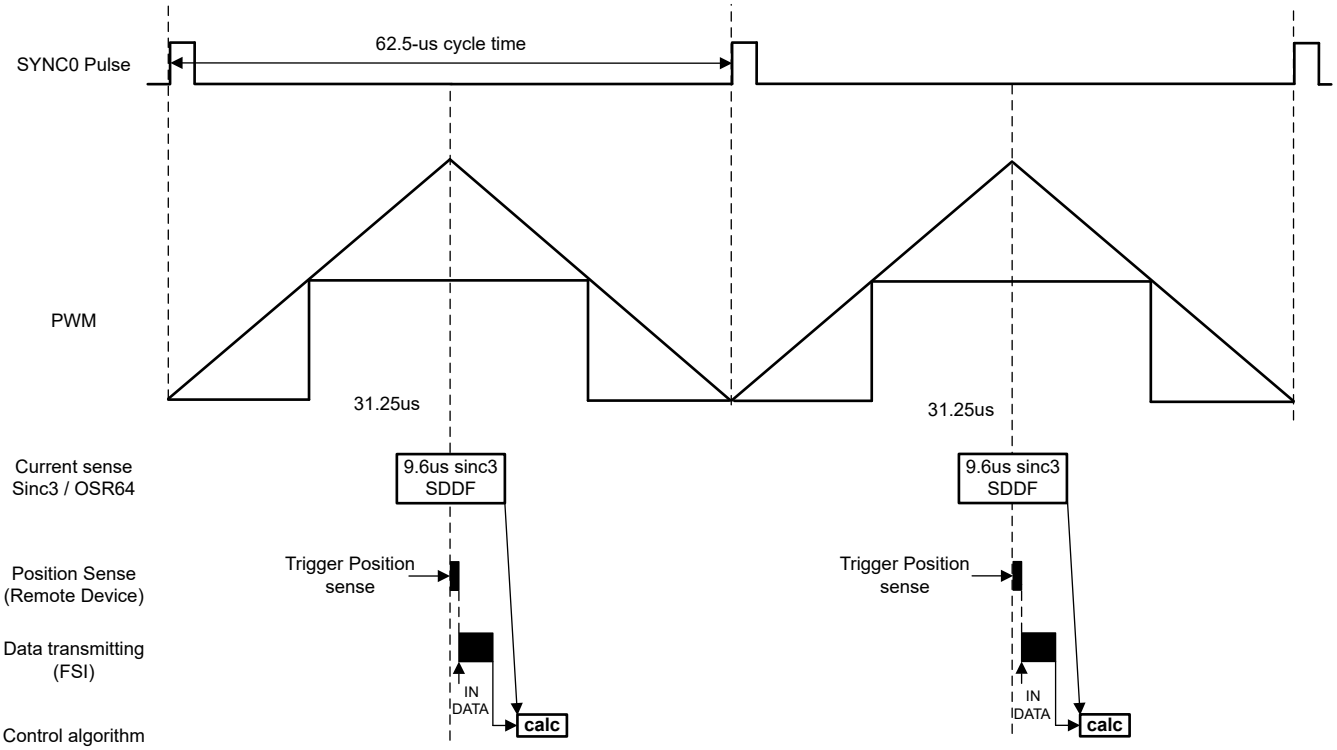

Figure 2-6. Time Synchronization and Data Transmitting Timing

Copyright © 2025 Texas Instruments Incorporated

2.5.1 Device 1 Configuration

The AM243x device 1 (controller) is implemented as the controller. The FSI configuration and PRU initialization are done in the ARM code. Following is the configuration for device 1:

2.5.1.1 Pad Configuration

AM243x pins and pads can be configured for multiple functions using pin multiplexing. In this application note, a specific pin is selected to output sync signal from ICSSG0 IEP0. Set MUXMODE bit field = 0x2 in the PADCONFIG107 register (0x000F41AC) to route the pin with ball name PRG0_PRU0_GPO19 to the signal PRG0_IEP0_EDC_SYNC_OUT0. This pin serves as the SYNC signal, generated by the IEP. Device 2 uses this signal as a reference to adjust the IEP counter, establishing synchronization with device 1.

2.5.1.2 Clock Source Configuration

Configure the CTRLMMR_ICSSG0_CLKSEL register (0x43008040), set CORE_CLKSEL bit field = 0x1 to select the ICSSG0 core clock as PLL0_HSDIV_CTRL9 and IEP_CLKSEL bit field = 0x1 to select the ICSSG0 IEP clock as PLL0_HSDIV_CTRL6.

The ICSSG0 core clock is set to 333MHz and the IEP clock is set to 250MHz. This is achieved by selecting the appropriate PLL dividers.

Configure HSDIV bit field = 0x2 in PLL0_HSDIV_CTRL9 register (0x006800a4), resulting in 333MHz for the ICSSG0 core clock.

Configure HSDIV bit field = 0x3 in PLL0_HSDIV_CTRL6 register (0x00680098), resulting in 250MHz for the ICSSG0 IEP clock.

2.5.1.3 IEP Timer Configuration

The IEP counter value can be read from the IEP_COUNT_REG0 (0x3002E010) and IEP_COUNT_REG1 (0x3002E014) registers. A compare event (CMP0 event) is generated when the IEP counter reaches the compare value set in CMP0 register.

Set the IEP counter to wrap around when the counter reaches the CMP0 value. To achieve this, set CMP0_RST_CNT_EN bit field = 0x1 in IEP_CMP_CFG_REG register (0x3002E070) register and enable CMP0 event.

Configure CMP0_0 bit field = 62500 - 4 in the IEP_CMP0_REG0 register (0x3002E078), resulting the IEP counter resets every 62500 counts, which corresponds to 62.5µs when the IEP clock source runs at 250MHz and the IEP counter is incremented by four per clock cycle.

Configure CMP1_0 bit field = 1000 - 4 in the IEP_CMP1_REG0 register (0x3002E080), to define the activation time of the sync signal to 1µs, which defines at which point in IEP counter the sync signal is activated, and enable CMP1 event.

Set the IEP counter's default increment value by configuring the DEFAULT_INC bit field = 0x4 in IEP_GLOBAL_CFG_REG register (0x3002E000), resulting the IEP counter increments four times per IEP source clock cycle.

Define the SYNC0 pulse width as 10 clock cycles by writing SYNC_HPW bit field = 10 - 1 in the IEP_SYNC_PWIDTH_REG register (0x3002E190). This determines the duration for which SYNC0 signal remains high.

Set SYNC_EN and SYNC0_EN bit fields = 0x1 and SYNC0_CYCLIC_EN bit field = 0x0 in the IEP_SYNC_CTRL_REG (0x3002E180) register to enable SYNC0 in single-shot mode.

Set the PWM_EFC_EN bit field = 0x1 in the ICSSG_SA_MX_REG register (0x30026040) to enable the IEP CMP flags to get auto hardware cleared.

Once the required register configurations are done, the CMP0 task is written.

Copyright © 2025 Texas Instruments Incorporated

2.5.1.4 Task Manager Configuration

Set the task manager to general purpose mode by configuring the TASKS_MGR_MODE bit field = 0x2 in TASKS_MGR_GLOBAL_CFG register (0x3002A000), enable task 2 sub task 1 by setting TS2_EN_S1 bit field = 0x1 and enable task 2 sub task 4 by setting TS2_EN_S4 bit field = 0x1.

Write the address of the TS2_S1 (CMP0 task) that has to be executed when a CMP0 event occurs to the TASKS_MGR_TS2_PC_S1 register (0x3002A020). The CMP0 task generates the sync signal in single shot mode. Write TS2_GEN_S1_MX bit field = 16 in TASKS_MGR_TS2_GEN_CFG1 register (0x3002A040) to configure CMP0 event as the trigger for executing the TS2_S1 task (CMP0 task).

So, the compare 0 event of IEP is configured to reset IEP timer with 62.5us cycle time and also to hit the task 0 to generate the SYNC0 pulse for synchronization. The SYNC0 activation time is triggered by IEP compare 1 event. The SYNC0 pulse generation mode is set to single-shot mode and pulse width set to 50ns. During the task 0, the SYNC0 pulse is disabled then enabled.

Write the address of the TS2_S4 (INTC task) that has to be executed when an FSI_RX interrupt event occurs to the TASKS_MGR_TS2_PC_S4 register (0x3002A02C). The INTC can generate the event to hit task 1 once the FSI RX buffer is filled up. During the task 1, the PRU firmware can move the position data received from remote device 2 to the tightly coupled memory (TCM) of R5F core via the PRU XFR2VBUS hardware accelerator. The TX write buffer and RX read buffer of the XFR2VBUS widget are all 64 bytes deep. Write TS2_GEN_S4_MX bit field = 136 in TASKS_MGR_TS2_GEN_CFG2 register (0x3002A044) to configure INTC event as the trigger for executing the TS2_S4 task (INTC task).

2.5.2 Device 2 Configuration

The AM243x device 2 (device) is implemented as the device. Similar as the controller software, the FSI configuration and PRU driver are initialized in the ARM code. Following is the configuration for device 2:

2.5.2.1 Pad Configuration

Route the pin PRG0_PRU0_GPO18 to the PRG0_IEP0_EDC_LATCH_IN0 signal by setting the MUXMODE bit field = 0x2 in the PADCONFIG106 register (0x000F41A8). The PRG0_PRU0_GPO19 pin configuration is similar to that of device 1.

2.5.2.2 Clock Configuration

The ICSSG0 core clock and IEP clock configurations are identical to those of Device 1.

2.5.2.3 IEP Timer Configuration

The compare event and SYNC0 configuration is the same as device 1, with the following additional steps. Enable capture 6 event CAP6 by setting bit 6 in the IEP_CAP_CFG_REG register (0x3002E018). The IEP counter value is captured to CAP6 register when a latch event occurs due to the sync signal from device 1. This is achieved by routing the sync signal from device 1 through the PRG0_PRU0_GPO18 to the IEP latch of device 2 using the TSR module.

2.5.2.4 TSR Configuration

Configure the TSR to route the PRG0_IEP0_EDC_LATCH_IN0 signal to PRU_ICSSG0_pr1_edc0_latch0_in_IN_0. This establishes a signal path from PRG0_PRU0_GPO18 to the ICSSG0 IEP0 latch. The routing is performed by selecting the appropriate input and output signal to the TSR. Set the MUX_CNTL bit field = 0x4 in the TIMESYNC_EVENT_INTROUTE R0 register (0x00a40024).

Copyright © 2025 Texas Instruments Incorporated

2.5.2.5 Task Manager Configuration

There are 4 tasks of the PRU firmware for the device 2 and the task manager configuration is similar to those of device 1:

- Task 0 is hit by the capture 6 event of IEP to latch the SYNC0 pulse from controller. Also compensate the incremental value of IEP to synchronize the timing with the controller. The compensation value is calculated using the following formula:

Compensation value = CAP6 – CMP1 – hardware delay

Compensation is applied based on the calculated value. If the compensation value is positive, then set CMP_INC bit field = 0x3 in the IEP_GLOBAL_CFG_REG register (0x3002E000). Write the COMPEN_CNT bit field = calculated compensation value in the IEP_COMPEN_REG register (0x3002E008). As a result, the IEP counter can be incremented by 3 for the number of times specified by the calculated compensation value. If the compensation value is negative, then set CMP_INC bit field = 0x5. Write the COMPEN_CNT bit field = absolute of the compensation value in IEP_COMPEN_REG register (0x3002E008). In this case, the IEP counter can be incremented by 5 for the number of times specified by the calculated compensation value.

- Task 1 is hit by the compare 0 event of IEP to generate the SYNC0 pulse to compare the output SYNC0 pulse of the device with the SYNC0 pulse from the controller, also the compare 0 event is used to reset the IEP timer. The SYNC0 activation time is triggered by IEP compare 1 event. The SYNC0 pulse generation mode is set to single-shot mode and pulse width set to 50ns via IEP_SYNC_CTRL_REG and IEP_SYNC_PWIDTH_REG respectively. During the task 0, the SYNC0 pulse is disabled then enabled by IEP_SYNC_CTRL_REG [0].
- Task 2 is hit by the compare 2 event of IEP for pre-scheduled. Assuming that the position sense can start from this time slot.
- Task 3 is hit by the compare 3 event of IEP to start transmitting position data via FSI interface. During the task, PRU firmware can set the FSI TX buffer, move the position data from TCM to the FSI TX buffer via the PRU XFR2VBUS hardware accelerator and start the FSI transmitting.

Copyright © 2025 Texas Instruments Incorporated

3 Verification

To verify the scheme, 2 AM243x launch Pad are used with some hardware connection including:

- Connect LP-AM243x device 1 BP.45 (SYNCOUT0) to LP-AM243x device 2 BP.8 (LATCH_IN0) by wire
- Connect device 1 J16 pin 1 (FSI RXCLK) to device 2 J16 pin 2 (FSI TXCLK) by wire
- Connect device 1 J16 pin 5 (FSI RXD0) to device 2 J16 pin 6 (FSI TXD0) by wire
- Connect device 1 J16 pin 7 (FSI RXD1) to device 2 J16 pin 8 (FSI TXD1) by wire
- Connect device 1 J16 pin 3 (GND) to device 2 J16 pin 4 (GND) by wire

Figure 3-1 shows the verification set up for the demo and Figure 3-2 is the waveform for the timing of all the signals including:

- Channel 0: **device 2** – BP.33 (PRG0_PRU0_GPO0 toggling to show the cycle time using IEP compare0 event)
- Channel 1: **device 2** – BP.32 (PRG0_PRU0_GPO1 toggling to show the task 2 triggered by IEP compare2 event used for pre-scheduled starting time of position sense)
- Channel 2: **device 2** – BP.31 (PRG0_PRU0_GPO2 toggling to show the task 3 triggered by IEP compare3 event used for pre-scheduled starting time of position data transmitting via FSI)
- Channel 3: **device 1** – J6 pin1 (FSI TX CLK) or device 2 – J16 pin 2 (FSI RX CLK)
- Channel 4: **device 1** – BP.11 (PRG0_PRU1_GPO0 toggling to show the task 1 triggered by INTC event used for move the position data from FSI RX buffer to TCM)
- Channel 5: **device 1** – BP.51 (system GPO1_20 toggling to show PRU1 interrupt once the position data movement finished)
- Channel 6: **device 1** – BP.45 (IEP0_SYNCOUT0)
- Channel 7: **device 2** – BP.45 (IEP0_SYNCOUT0)

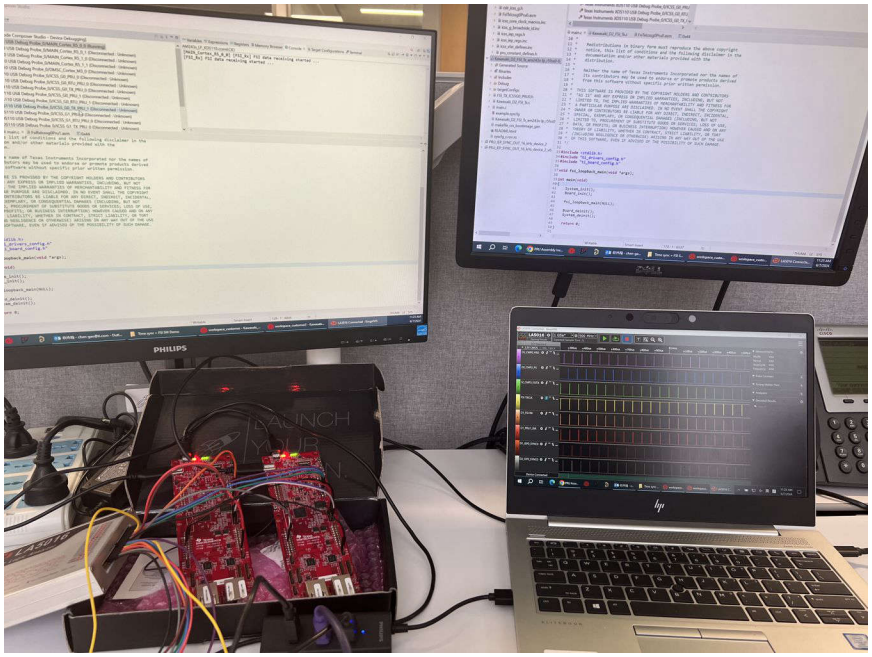

Figure 3-1. Verification Demo Set Up

Copyright © 2025 Texas Instruments Incorporated

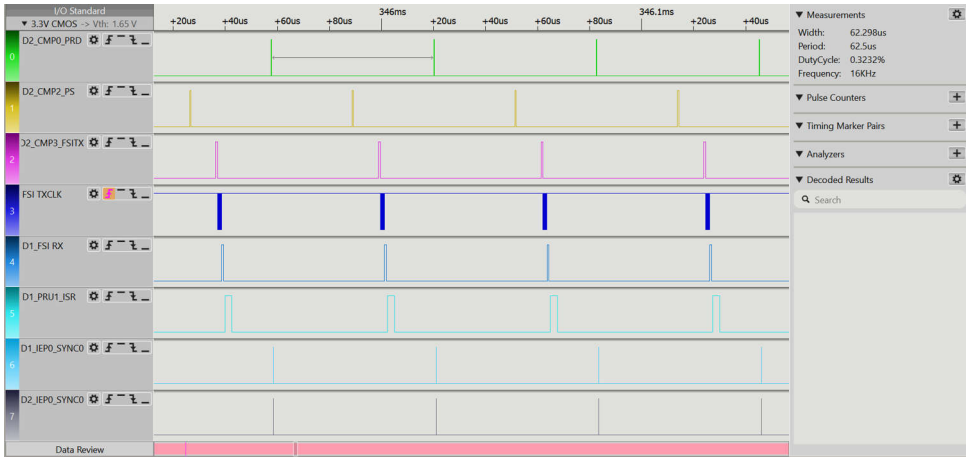

Figure 3-2. Overall Timing of all Signaling

The cycle time is set to 62.5us at the PWM cycle level (16kHz) which shows on channel 0. The position sense starts at the central point of the cycle which shows on channel 1. The FSI latency including the processing time for FSI TX and RX also the data transmitting is shown on channel 2, channel 4 and channel 3 respectively. The total communication latency via FSI is around 3.046us for transmitting 32 bytes data words. Channel 6 and 7 are the IEP SYNCOUT0 signals generated by both device 1 and device 2 which are aligned well by compensate the delay. Figure 3-3 shows the verification results mentioned above. Also Figure 3-4 shows the drift of IEP SYNCOUT0 pulse between device 1 and device 2 is less than 5ns with overnight testing. The drift is compensated in the device 2. Thus, the process synchronize the IEP counters of two AM243x devices.

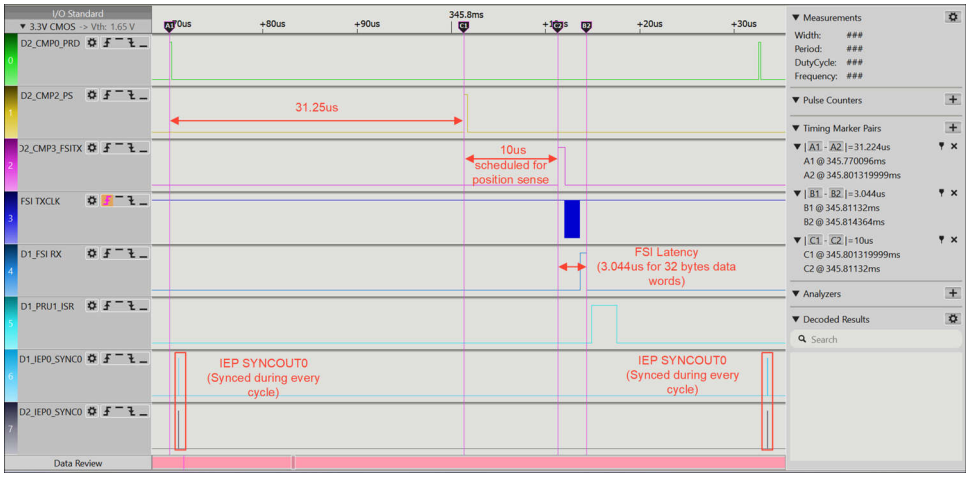

Figure 3-3. Verification Results of the Timing

Copyright © 2025 Texas Instruments Incorporated

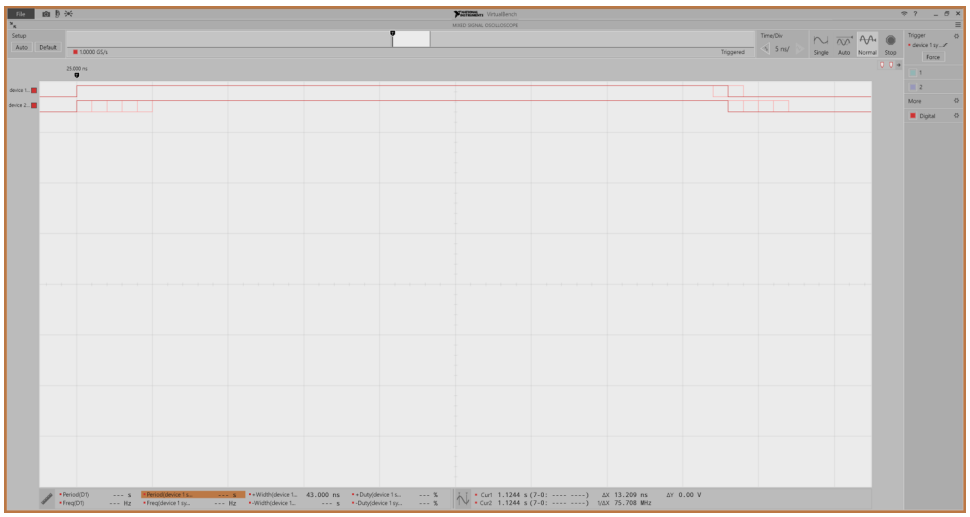

Figure 3-4. Overnight Testing Result for Drift of IEP SYNCOUT0 Between Devices

The data transmitting validation result is shown on Figure 3-5. 32 bytes data are written into the ICSSG Dynamic Random-Access Memory (DRAM) of device 2 in advance as the position data. Tightly Coupled Memory (TCM) of device 1 stores the data received from FSI RX buffer. All the data is copied to the debug buffer *gRxBufData* inside the PRU core interrupt to showcase the data validation.

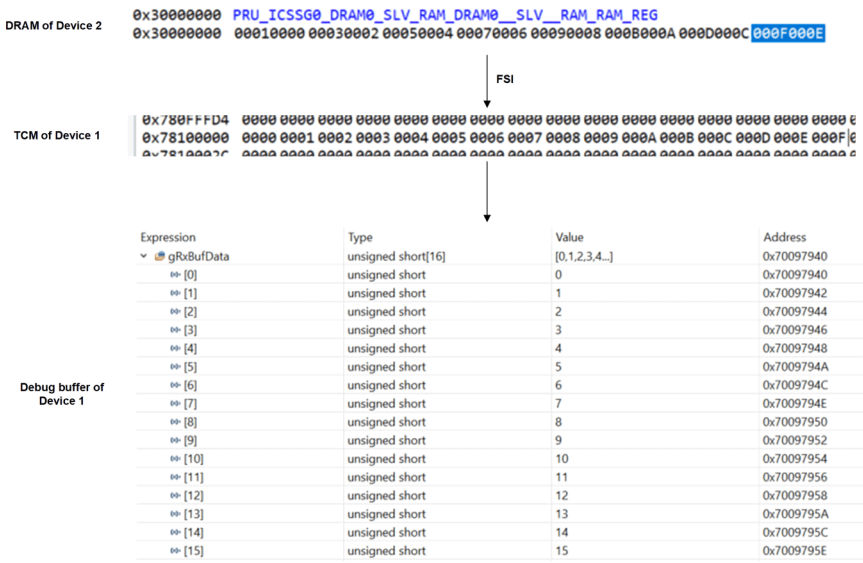

Figure 3-5. Data Transmitting Validation Result

Copyright © 2025 Texas Instruments Incorporated

4 Summary

This application note describes how to use IEP timer and PRU cores to perform time synchronization between chips and transmitting data with configurable timing under control cycle via FSI interface for closed-loop motor control.

5 References

1. Texas Instruments, *AM243x Sitara Microcontrollers*, data sheet.
2. Texas Instruments, *AM64x/AM243x Technical Reference Manual*, user guide.

Copyright © 2025 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2025, Texas Instruments Incorporated