# PDK/PDK DIAGNOSTICS User Guide

### Contents

**Return to PDK Software Developer Guide** 

- Introduction McSPI Diagnostic Library Diagnostics Test Description Features tested ECC Diagnostic Library
  - Diagnostics Test Description Features tested CRC Diagnostic Library
  - Diagnostics Test Description Features tested
  - ADC Diagnostic Library Diagnostics Test Description Features tested
  - CAN Diagnostic Library Diagnostics Test Description Features tested
  - Running Diagnostic Library Examples
    - Diag Test App diag ipu ecc test
- Introduction

TI ADAS Diagnostic Library is a collection of software functions for diagnosing the health of various modules and interfaces of the device. The functions of this library can be called by user at Start of system, Runtime or System shutdown.

The diagnostic library supports the following functionalities

- can be used to diagnose the functionality of the ADC module.
- can be used to diagnose the ECC functionality of EMIF, OCMC & IPU modules.
- can be used to configure CRC to calculate CRC signature of given data and verify its functionality.
- can be used to test SPI internal loopback for various frequencies and modes for all 4 channels and verify its functionalities.
- can be used to test the CAN internal/external loopback and ECC parity for DCAN1 and DCAN2 channels and verify its functionalities.

# **McSPI Diagnostic Library**

The MCSPI is a master/slave synchronous serial bus. The MCSPI modules include:

- Serial clock with programmable frequency, polarity, and phase for each channel
- Selection of SPI word length (4 to 32 bits)
- Full duplex/half duplex transfers in master mode.
- Built-in FIFO for a single channel.

### **Diagnostics Test Description**

- Initialize all the input parameters by using default init function which initializes the input parameters to a default value.
- Run the internal pad loopback by configuring the same signal as both input and output. Configure spim\_d[0] as output and spim\_d[1] as input and connect them.
- Compare the values of the Tx and Rx buffers.
- MCSPI can be tested in two modes:
  - DMA mode.
  - Polled mode.
- In the interrupt mode, application has to register an ISR handler for the MCSPI interrupt in which DiagLib\_mcspiProcISR has to be called.

### **Features tested**

- Test different instances of MCSPI module
- Test different channels supported by each MCSPI instance.
- Test for different frequencies: 1MHz, 48 MHz (Serial clock with programmable frequency)
- Test for different modes: Mode 0, Mode 1, Mode 2, Mode 3 (polarity, and phase for each channel)
- FIFO Buffer
- Test for different word length (8,16 & 32 only supported for test)
- Test MCSPI interrupts.

### **ECC Diagnostic Library**

Error Correcting Code (ECC) bits is redundant data added along with the original data, which is a technique that helps in detecting as well as correcting errors. The ECC scheme used on the TDA2xx/TDA2Px/TDA2Ex/TDA3xx devices is able to detect up to 2-bit failures and correct single bit failures.

### **Diagnostics Test Description**

- DiagLib for ECC provides test on 3 modules:
  - EMIF Quanta alignment address/write, SEC, DED.
  - OCMC SEC, DED, ECC code Error.
- IPU L2 RAM/L1DATA/L1TAG SEC, DED, ECC Code Error.
- Initialize all the input parameters by using default init function which initializes the input parameters to a default value.
- It is mandatory to set the type of test and the address where error to be tested.
- Write data to specified memory with ECC enabled. ECC will be generated.
- Disable ECC and induce error in data.
- Read data and check if ECC error interrupt is generated
- ECC can be tested in two modes:
  - Interrupt mode.
  - Polled mode Not applicable for IPU and errors causing hard faults.
- In the interrupt mode, application has to register an ISR handler for the respective module ECC interrupts in which DiagLib\_eccProcISR has to be called.

#### Features tested

- Support different IPs; EMIF, IPU, OCMC.
- ECC Logic
- Interrupts
- IP Memory Read/Write Operations

## **CRC Diagnostic Library**

The CRC module produces a signature for a given sequence of data values. The signature is generated through a feedback path from data bits. Identical input data sequences result in identical signatures when the CRC is initialized with a fixed seed value, whereas different sequences of input data, in general, result in different signatures.

### **Diagnostics Test Description**

- Initialize all the input parameters by using default init function which initializes the input parameters to a default value.
- Calculate CRC of specified message through software implementation of CRC (C-Model).
- Send the same message through CRC module to get HW CRC Value.
- Compare CRC (Module) Value with predetermined CRC (C-Model) Value.
- CRC can be tested in two modes:
  - Interrupt mode.
  - Polled mode.
- In the interrupt mode, application has to register an ISR handler for the CRC interrupt in which DiagLib\_crcProcISR has to be called.

### **Features tested**

- CRC registers of each channel.
- All test support both DMA and CPU based mode
- CRC Logic
- Interrupts

## ADC Diagnostic Library

The analog-to-digital converter (ADC) module is a successive-approximation register (SAR) general purpose analog-to-digital converter. The main features of the ADC include:

- 10-bit data
- 8 channels
- 750 KSPS at 13.5-MHz ADC\_CLK
- Two modes of conversion Single conversion (one-shot) & Continuous conversions.
- Programmable FSM sequencer that supports 16 steps for averaging, open delay, sampling delay etc.
- Two 64-deep FIFO for data storage.

### **Diagnostics Test Description**

- DiagLib for ADC provides 2 types of tests:
  - Test ADC conversions.
  - Test ADC overrun.
- Initialize all the input parameters by using default init function which initializes the input parameters to a default value.
- It is mandatory to configure the number of channels, input channel numbers and their corresponding expected voltages.

#### 9/29/2020

- Any other parameter can be optionally modified as required by the USER application.
- Test type has to be chosen and start test API has to be called from the application with appropriately populated input parameters.
- ADC can be tested in two modes:
  - Interrupt mode.
  - Polled mode.
- In the interrupt mode, application has to register an ISR handler for the ADC interrupts in which DiagLib\_adcProcISR has to be called.

### Features tested

- Single shot mode & continuous mode for x samples
- Range check
- FIFO buffer threshold, overrun.
- All channel support
- Interrupt and Polled mode.

# **CAN Diagnostic Library**

The CAN module is a serial communications protocol which efficiently supports distributed real-time applications. CAN has high immunity to electrical interference and the ability to self-diagnose and repair data errors. In a CAN network, many short messages are broadcast to the entire network, which provides for data consistency in every node of the system.

### **Diagnostics Test Description**

- Initialize all the input parameters by using default init function which initializes the input parameters to a default value.
- Run the internal and external pad loopback test.
- Run the ecc parity test.
- Any parameter can be optionally modified as required by the USER application
- Compare the values of the Tx and Rx buffers.
- Application has to register an ISR handler for the CAN interrupt in which DiagLib\_CanISR has to be called.

### Features tested

- Test all instances of CAN module for loopback and parity DCAN1 and DCAN2
- Parity error interrupts
- Single bit parity error detection and correction
- Double bit parity error detection
- TX and RX interrupts

## **Running Diagnostic Library Examples**

### **Diag Test App**

 $The example is present in path packages \ti\drv\stw\_lld\examples\diag\_test\_app.$ 

This app runs on tda2xx, tda2px, tda2ex and tda3xx. This app tests the functionality of the selected safety IPs. This app provides an example for the diagnosis of ADC, CRC, ECC(EMIF & OCMC), DCAN & McSPI modules.

The example provides the instance of

- ADC singleshot & continuous conversions of multiple channels.
- Inbuilt test CRC compression by comparing the CRC of default data with Golden CRC.
- McSPI1 internal loopback functionality.
- DCAN internal loopback functionality and single bit error correction.
- ECC EMIF Non Quanta Multiple writes error.
- ECC OCMC SEC error.

Note:

- For EMIF ECC Test on Tda2xx/Tda2px/Tda3xx/Tda2ex, Before connecting the EVM to CCS through JTag, modify the gel files and then reload these gel files as stated below.
- Enable the macro ENABLE\_ECC in ddr config gel files. i.e., TDA2xx\_ddr\_config.gel (Tda2xx), TDA2Px\_ddr\_config.gel (Tda2px), TDA2Ex\_ddr\_config.gel(Tda2ex), TDA3xx\_ddr\_config.gel(Tda3xx).
- For ADC Test connect the ADC setup as stated below.
  - Connect 665V to pin3 , 592V to pin5, 520V to pin 7 and 412V to pin 9

### diag ipu ecc test

The example is present in path packages packages\ti\diag\examples\diag\_ipu\_ecc\_test This app runs on tda3xx. This app tests the error detection and correction functionalities of IPU in M4. The example provides the instance of

- IPU L2 RAM SEC, DED & Code error.
- IPU L1 DATA SEC, DED & Code error.
- IPU L1 TAG SEC, DED & Code error.

Note:

- Only interrupt mode is supported for this test. The application registers IRQ handler inside which it calls a DiagLib\_ecclSR for respective modules.
- Before loading binary on Cortex M4 core run the gel file script AMMU\_config\_IPU\_ECC\_Test from the gel file provided at location docs\gel/ECC\_Diag\_Cfg.gel

Кеу	/stone=		
<ul> <li>{{         <ol> <li>switchcategory:MultiCore=</li> <li>For technical support on MultiCore devices, please post your questions in the <u>C6000 MultiCore Forum</u></li> <li>For questions related to the BIOS MultiCore SDK (MCSDK), please use the <u>BIOS Forum</u></li> </ol> </li> <li>Please post only comments related to the article PDK/PDK DIAGNOSTICS User Guide here.</li> </ul>	For technical support on MultiCore devices, please post your questions in the <u>Forum</u> For questions For questions For questions For questions BIOS MultiCore BIOS MultiCore BIOS Soft post your questions on For <u>questions</u> BIOS MultiCore BIOS Forum BIOS Forum Ase post only SDK (MCSDK), please use the BIOS Forum Ase post only BIOS Forum Ase post only BIOS FOR BIOS	DaVinci=For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article PDK/PDK DIAGNOSTICS User Guide here. MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article post Guide here.	OMAP35x=For technicalOMAPL1=For technicalMAVRK=For technicalsupport on OMAP pleasesupport on support onsupport on support onsupport on support on post your questions on questions on questions on questions on The OMAPFor The OMAP The OMAPFor The OMAP The OMAP The OMAP The OMAP The MAVRKFor multical post only Please post only post only post only comments about the articleThe OMAP Forum. Please about the about the about the about the about the articleThe MAVRK Please post only comments about the about the about the about the about the atricleThe MAVRK Please post ar only comments about the about the about the about the atricleThe MAVRK Please post ar only comments about the about the about the about the atricleThe MAVRK Please post ar about the about the about the about the about the atricleThe MAVRK Please post ar about the about the about the about the about the about the atricleThe MAVRK Please post ar atricleThe MAVRK Please about the about the about the about the about the atricleThe MAVRK Please about the about the about the about the about the articleThe MAVRK Please about the about the about the about the about the bulkThe MAVRK about the about the about the about the about the bulkThe MAVRK about the about the about the about the about the bulkThe MAVRK about the about the about the about the about the about the abo
Links			
Amplifiers & Linear Audio Broadband RF/IF & Dig Clocks & Timers Data Converters	DLP & MEMS High-Reliability gital Radio Interface Logic Power Management	Processors ARM Processors Digital Signal Processors (DS Microcontrollers (MCU) OMAP Applications Processor	Switches & Multiplexers Temperature Sensors & Control ICs Wireless Connectivity SP)

Retrieved from "https://processors.wiki.ti.com/index.php?title=PDK/PDK\_DIAGNOSTICS\_User\_Guide&oldid=231088"

This page was last edited on 13 October 2017, at 05:38.

Content is available under Creative Commons Attribution-ShareAlike unless otherwise noted.