MMWAVE SDK Unit Test Procedure

Document Version: 1.0

DOCUMENT LICENSE

COPYRIGHT

CONTENTS

1. SPI Unit Tests

1. 1. Executable

```
mmwave_sdk_<ver>/packages/ti/drivers/spi/test/<device>/<device>_spi_mss.xer4f
```

1. 2. SPI Test Configuration Variables

Config Variable	Comments
gXWR1xxxLoopbackTest	Enable for all single EVM loopback tests
gXWR1xxxMasterWithXWR1xxx	Configure in Master mode for back-to-back tests
gXWR1xxxSlaveWithXWR1xxx	Configure in Slave mode for back-to-back tests
gXWR1xxxSlaveWithFTDITest	Enable for FTDI test to PC Application

When the unit test is executed using CCS, the test variables can be manipulated via the 'expression' window to the required value before running the test (i.e. after the loading of unit test binary (code reaches main) but before hitting run in CCS).

1. 3. SPI Test Status Variables

Status Variable	Comments
gXWR1xxxSlaveReady	Slave side is ready to receive transmissions for back-to-back test sequence

1. 4. R4 SPI Loopback Test

1. 4. 1. Test Summary

Single EVM test performs multiple on-board SPI peripheral tests.

1. 4. 2. Variable Assignment

Config Parameter	Loopback Value
gXWR1xxxLoopbackTest	1
gXWR1xxxMasterWithXWR1xxx	0
gXWR1xxxSlaveWithXWR1xxx	0
gXWR1xxxSlaveWithFTDITest	0

1. 4. 3. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\module_test\xwr16_R4_spi_loopback_<timestamp>.log" for the detailed test log.

1. 4. 4. Basic Test Sequence

- API check
- SPI loopback Test for SPI-A
    - Throughput results reported
- SPI loopback Test for SPIB on Slave 0
    - Throughput results reported
- SPI loopback Test for SPIB on Slave 1
    - Throughput results reported
- SPI loopback Test for SPIB on Slave 2
    - Throughput results reported

1. 5. R4 SPI back-to-back Test

1. 5. 1. Test Summary

Two EVMs are wired for SPI interface cross-connection. Tests on each EVM are run in both Master mode and in Slave mode. The Master mode sends a test sequence to the Slave mode EVM, and it echoes the sequence back to the Master mode EVM. That EVM does a data integrity test.

For the back-to-back tests, the Slave EVM must have completed initialization and be ready to receive prior to the Master EVM beginning transmission. A tester can either monitor the ‘gXWR1xxxSlaveReady’ variable or view the console output to know when the Slave is ready. At this point the Master device can start the test sequence (i.e. hit run in CCS), and the test sequence flows programmatically to test completion.

To connect 2 mmWave EVMs to the same machine and execute via CCS, refer to instructions mentioned here: http://software-dl.ti.com/ccs/esd/documents/sdto_ccs_multi-probe-debug.html

1. 5. 2. Variable Assignment

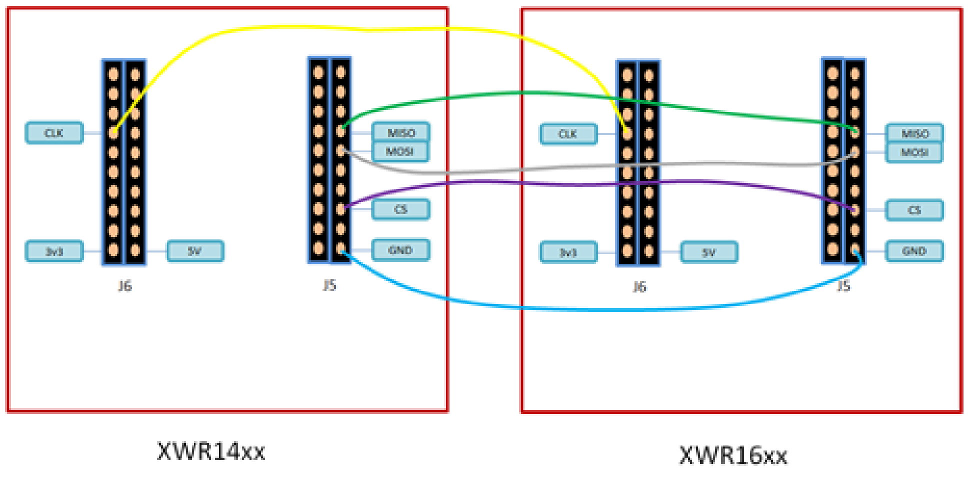
1. 5. 2. 1. Slave EVM

Parameter	Loopback Value
gXWR1xxxLoopbackTest	0
gXWR1xxxMasterWithXWR1xxx	0
gXWR1xxxSlaveWithXWR1xxx	1
gXWR1xxxSlaveWithFTDITest	0

1. 5. 2. 2. Master EVM

Parameter	Loopback Value
gXWR1xxxLoopbackTest	0
gXWR1xxxMasterWithXWR1xxx	1
gXWR1xxxSlaveWithXWR1xxx	0
gXWR1xxxSlaveWithFTDITest	0

1. 5. 3. Physical Setup: Hardware EVM to EVM cross-connections



1. 5. 4. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\module_test\xwr16_R4_SPI_bk2bk_Slave_(BackToBack)_<timestamp>.log" for the detailed test log.

Basic Test Sequence

- Back to Back test, one EVM in Master mode, one EVM in Slave mode
  - Sync and transmission tests, Master-to-Slave, echoed back Slave-to-Master

    - Test executes data integrity test at various supported supported bit rates and modes
    - Throughput results reported
- Switch Master/Sync modes between EVMs, Slave mode, Master mode
    - Sync and transmission tests, Master-to-Slave, echoed back Slave-to-Master
    - Test executes data integrity test at various supported supported bit rates and modes
    - Throughput results reported

1. 6. R4 SPI/FTDI Test

1. 6. 1. Test Summary

The EVM sends a test sequence to a Unix PC test application via FTDI. That application echoes the sequence back to the EVM. The EVM does a data integrity test.

1. 6. 2. Unix PC Utilities

The Unix PC application is available in the ‘mmwave_sdk_<ver>/packages/ti/drivers/spi/test/unix’ directory. Please see README.txt for build and execute instructions.

1. 6. 3. Windows PC Utilities

Similarly, a test application is available for Windows as well. Look for the source and batch file in the ‘mmwave_sdk_<ver>/packages/ti/drivers/spi/test/windows/’ directory. Note: the FTDI driver needs to be installed for windows (driver is in mmwave_sdk_<ver>/tools/ftdi folder).

1. 6. 4. Variable Assignment

Parameter	Value
gXWR1xxxLoopbackTest	0
gXWR1xxxMasterWithXWR1xxx	0
gXWR1xxxSlaveWithXWR1xxx	0
gXWR1xxxSlaveWithFTDITest	1

1. 6. 5. Physical setup

The EVM is connected to PC USB port.

1. 6. 6. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\AWR16\module_test\xwr16_R4_spiFTDI_<timestamp>.log" for the detailed test log.

1. 6. 7. Basic Test Sequence

- Configure driver for FTDI test
- Sync and transmission tests, PC application echoes back to EVM via SPI/FTDI
- Multiple test transmissions.
- The EVM does a data integrity test.

2. CAN Unit Tests

2. 1. Executable

mmwave_sdk_<ver>/packages/ti/drivers/can/test/<device>/<device>_can_mss.xer4f

2. 2. CAN Test Configuration Variables

Test Scenario	`testSelection` Value	Interface	Test Scenario
Internal loopback test	1	internal	Internal, digital loopback test
external loopback test	2	internal	external, analog loopback test

Parity test	3	PCAN	Parity test
Tx/Rx test	4	PCAN	Tx/Rx test

When the unit test is executed using CCS, the test variable 'testSelection' can be manipulated via the 'expression' window to the required value before running the test (i.e. after the loading of unit test binary (code reaches main) but before hitting run in CCS).

2. 3. R4 CAN Internal Loopback Test

2. 3. 1. Test Summary

The unit test configures for a chip-internal, digital loopback of the Tx and Rx signals. The test exercises multiple iterations. The unit test does a data integrity test.

2. 3. 2. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CAN_internal_loopback_<timestamp>.log" for the detailed test log.

2. 4. R4 CAN External Loopback Test

2. 4. 1. Test Summary

The unit test configures for a within-the-EVM, analog loopback of the Tx and Rx signals. The test exercises multiple iterations. The unit test does a data integrity test. No external setup or cabling is required for this test.

2. 4. 2. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CAN_external_loopback_<timestamp>.log" for the detailed test log.

2. 5. R4 CAN Tx/Rx Test to PCAN

2. 5. 1. Test Summary

In CAN mode the unit test implements a Transmit and Receive test with the PCAN test utility. The test exercises multiple iterations. The unit test does a data integrity test. Please refer to the "PCAN-USB Test Utility" section below for the physical test requirements using the PCAN system.

Transmit frame:

["a1 1a ff ff c1 1c b1 1b","a2 2a ff ff c2 2c b2 2b","a3 3a ff ff c3 3c b3 3b", "a4 4a ff ff c4 4c b4 4b","a5 5a ff ff c5 5c b5 5b","a6 6a ff ff c6 6c b6 6b","a7 7a ff ff c7 7c b7 7b","a8 8a ff ff c8 8c b8 8b"]

2. 5. 2. PCAN Unix command line

"~/peak-linux-driver-8.5.1/test/pcanfdtst rx -t 10 -n 9 -c 40M -b 1M -d 5M /dev/pcan32"

2. 5. 3. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CAN_Tx_Rx_<timestamp>.log" for the detailed test log.

3. CAN FD Unit Tests

3. 1. Executable

mmwave_sdk/packages/ti/drivers/canfd/test/<device>/<device>_canfd_mss.xer4f

3. 2. CAN FD Test Configuration Variables

Test Scenario	`testSelection` Value	`testFrameType` Value	Interface
Internal loopback test	1	1	internal
external loopback test	2	1	internal
Multiple Tx test	3	1	PCAN
External Tx/Rx Classic test	4	0	PCAN
External Tx/Rx FD test	4	1	PCAN
EVM-to-EVM test	5	1	PCAN
Tx Cancel test	6	1	PCAN
Power down test	7	1	PCAN

When the unit test is executed using CCS, the test variables 'testSelection' and testFrameType can be manipulated via the 'expression' window to the required value before running the test (i.e. after the loading of unit test binary (code reaches main) but before hitting run in CCS). The variable 'testFrameType' enable the CAN mode (lower rates) in the CAN FD unit test.

3. 3. R4 CANFD Multiple Tx

3. 3. 1. Test Summary

In CAN FD mode the unit test implements multiple Transmit sequences to the PCAN test utility. PCAN returns on the command line the Receive data. The test exercises multiple iterations. The user needs to check the received data. Please refer to the "PCAN-USB Test Utility" section below for the physical test requirements using the PCAN system.

Transmit frame:

["a1 1a ff ff c1 1c b1 1b","a2 2a ff ff c2 2c b2 2b","a3 3a ff ff c3 3c b3 3b", "a4 4a ff ff c4 4c b4 4b","a5 5a ff ff c5 5c b5 5b","a6 6a ff ff c6 6c b6 6b","a7 7a ff ff c7 7c b7 7b","a8 8a ff ff c8 8c b8 8b"]

3. 3. 2. PCAN Unix command line

"~/peak-linux-driver-8.5.1/test/pcanfdtst rx -t 10 -n 2 -c 40M -b 1M -d 5M /dev/pcan32"

3. 3. 3. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CANFD_Multiple_Tx_<timestamp>.log"

3. 4. R4 CANFD External Tx/Rx Classic

3. 4. 1. Test Summary

In CAN mode (note: not CAN FD) the unit test implements multiple Transmit sequences to the PCAN test utility. PCAN returns on the command line the Receive data. The test exercises multiple iterations. The user needs to check the received data. Please refer to the "PCAN-USB Test Utility" section below for the physical test requirements using the PCAN system.

Transmit frame:

["a1 1a ff ff c1 1c b1 1b","a2 2a ff ff c2 2c b2 2b","a3 3a ff ff c3 3c b3 3b", "a4 4a ff ff c4 4c b4 4b","a5 5a ff ff c5 5c b5 5b","a6 6a ff ff c6 6c b6 6b","a7 7a ff ff c7 7c b7 7b","a8 8a ff ff c8 8c b8 8b"]

3. 4. 2. PCAN Unix command line

"~/peak-linux-driver-8.5.1/test/pcanfdtst rx -t 10 -n 9 -c 40M -b 1M -d 5M /dev/pcan32"

3. 4. 3. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CANFD_External_Tx_Rx_Classic_<timestamp>.log"

3. 5. R4 CANFD External Tx/Rx FD

3. 5. 1. Test Summary

In CAN FD mode the unit test implements multiple Transmit sequences to the PCAN test utility. PCAN returns on the command line the Receive data. The test exercises multiple iterations. The user needs to check the received data. Please refer to the "PCAN-USB Test Utility" section below for the physical test requirements using the PCAN system.

Transmit frame:

["a1 1a ff ff c1 1c b1 1b","a2 2a ff ff c2 2c b2 2b","a3 3a ff ff c3 3c b3 3b", "a4 4a ff ff c4 4c b4 4b","a5 5a ff ff c5 5c b5 5b","a6 6a ff ff c6 6c b6 6b","a7 7a ff ff c7 7c b7 7b","a8 8a ff ff c8 8c b8 8b"]

3. 5. 2. PCAN Unix command line

"~/peak-linux-driver-8.5.1/test/pcanfdtst rx -t 10 -n 2 -c 40M -b 1M -d 5M /dev/pcan32"

3. 5. 3. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4CANFD_External_Tx_Rx_FD_<timestamp>.log"

3. 6. R4 CANFD internal loopback

3. 6. 1. Test Summary

The unit test configures for a chip-internal, digital loopback of the Tx and Rx signals. The test exercises multiple iterations. The unit test does a data integrity test.

3. 6. 2. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CANFD_internal_loopback_<timestamp>.log"

3. 7. R4 CANFD external loopback

3. 7. 1. Test Summary

The unit test configures for a within-the-EVM, analog loopback of the Tx and Rx signals. The test exercises multiple iterations. The unit test does a data integrity test.

3. 7. 2. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CANFD_external_loopback_<timestamp>.log"

3. 8. R4 CANFD Power Down

3. 8. 1. Test Summary

The EVM sends Power Down and Wake Up control signals to the PCAN application. Please refer to the "PCAN-USB Test Utility" section below for the physical test requirements using the PCAN system.

3. 8. 2. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CANFD_Power_Down_<timestamp>.log"

3. 9. R4 CANFD EVM-EVM

3. 9. 1. Test Summary

Two EVMs with connected CAN signals Transmit sequences to each other. The test exercises multiple iterations. The unit test does a data integrity test. The two EVMs must have their respective CAN signals connected per below:

EVM-1	EVM-2
CANFD-L	CANFD-L
GND	GND
CANFD-H	CANFD-H

Transmit frame:

["a1 1a ff ff c1 1c b1 1b","a2 2a ff ff c2 2c b2 2b","a3 3a ff ff c3 3c b3 3b", "a4 4a ff ff c4 4c b4 4b","a5 5a ff ff c5 5c b5 5b","a6 6a ff ff c6 6c b6 6b","a7 7a ff ff c7 7c b7 7b","a8 8a ff ff c8 8c b8 8b"]

3. 9. 2. Example Test Log

Please refer to release document file "mmwave_sdk_<ver>\docs\test\IWR16\can_test\xwr16_R4_CANFD_EVM-EVM_<timestamp>.log"

4. PCAN-USB Test Utility

CAN and CAN FD compatibility tests are performed between the mmWave SDK on the EVM to a PC using the PCAN Test Utility.

PCAN is a product of PEAK Systems. Refer to this link for more information, https://www.peak-system.com/PCAN-USB-FD.365.0.html?&L=1.

The product includes an interface cable (see below) and PC software/drivers.

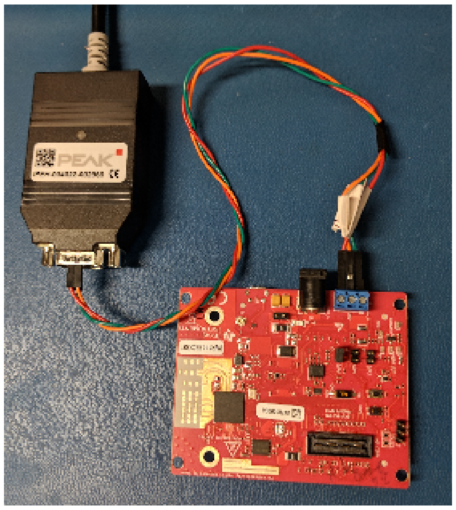
The User Guide is available at https://www.peak-system.com/produktcd/Pdf/English/PCAN-USB-FD_UserMan_eng.pdf

The Unix command line utility is available at https://www.peak-system.com/fileadmin/media/linux/index.htm

The Windows utilities are available at https://www.peak-system.com/quick/DrvSetup

4. 1. PCAN-USB board Setup

The PCAN device connects between a PC USB connector and the EVM CAN interface connector:



A High-speed CAN bus (ISO 11898-2) is connected to the 9-pin D-Sub connector. The pin assignment for CAN corresponds to the specification CiA® 303-1.

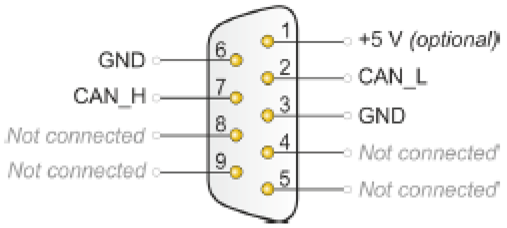


Figure 1: Pin assignment High-speed CAN
(view onto connector of the PCAN-USB FD adapter)

Signal Connections

Signal	EVM connection	PCAN 9-pin D-Sub
CAN-L	CANFD-L, CAN-L	pin 2
GND	GND	pin 3
CAN-H	CANFD-H, CAN-H	pin 7