Integration Test Specification

Common Timer Base

Department:	Aalborg Wireless Center
Creation Date:	11 March, 2003
Last Modified:	11 March, 2003 by Carsten Schmidt
ID and Version:	8434.525.03.001
Status:	Being Processed

0 Document Control

0.1 Document History

ID	Author	Date	Status
8434.525.03.001	CSH	11 March, 2003	Being Processed

0.2 References, Abbreviations, Terms

[TI 8010.801] 8010.801, References and Vocabulary, Texas Instruments

Table of Contents

1 Introduction

This document contains an overview over the functionality to be tested. Each point in this document should result in a test case. The aim of this document is testing with TAP.

2 Testing with CTB disabled

Most of the described test applications already exist today. These will be enhanced a little, so they can serve for CTB testing purposes. In this section the test cases are split in two parts. Namely test cases with special IDLE states (TIMEOUT, START_TIMEOUT, WAIT_TIMEOUT and MUTE) and test case without these states.

2.1 Without the use of special IDLE states in TAP.

2.1.1 Normal test case

Title	Running normal – without any timers. Default timeout should be 10000ms
Purpose	Insure that the protocol stack and TAP still works after the CTB implementation.
Test Description	Write a test application that tests sending / receiving of a primitive without using of IDLE states, except AWAIT
TDC test case nr:	Any of the existing TDC test cases today
Success Criteria	Test case should pass.

2.2 Using special IDLE states in TAP

2.2.1 With the use of TIMEOUT

Title	Running normal – with the use of TIMEOUT.
Purpose	Insure that the protocol stack and TAP still works after the CTB implementation. The feature TIMEOUT is used in the tap.
Test Description	Write a test application that tests sending / receiving of a primitive with the use of TIMEOUT. The TIMEOUT should be placed before an AWAIT event, so that the waiting time for a primitive is increased. The test case should contain two scenarios, one were a timeout will occur before something is received and one were something is received before the timeout.
TDC test case nr:	XX_TDC400A
Success Criteria	Test case should pass.

2.2.2 With the use of START_TIMEOUT and WAIT_TIMEOUT

Title	Running normal –WAIT_TIMEOUT should fail.
Purpose	Insure that the protocol stack and TAP still works after the CTB implementation. The test uses the timer functionality in the TAP. When calling WAIT_TIMEOUT the timer should be expired.
Test Description	Write a test application that tests sending / receiving of a primitive with the use of START_TIMEOUT and WAIT_TIMEOUT. Use primitive that starts timers on stack side. The scenario should be written, so that a WAIT_TIMEOUT will fail when called, because the timer already has expired in TAP.
TDC test case nr:	XX_TDC400B

Success Criteria	Test case should pass.

2.2.3 With the use of START_TIMEOUT and WAIT_TIMEOUT

Title	Running normal –WAIT_TIMEOUT should pass.
Purpose	Insure that the protocol stack and TAP still works after the CTB implementation. The test uses the timer functionality in the TAP. This time the timer should still be running.
Test Description	Write a test application that tests sending / receiving of a primitive with the use of START_TIMEOUT and WAIT_TIMEOUT. Use primitive that starts timers on stack side. The scenario should be written, so that a WAIT_TIMEOUT will NOT fail when called, since the timer already is running
TDC test case nr:	XX_TDC400C
Success Criteria	Test case should pass.

2.2.4 With the use of MUTE - fail

Title	Running normal – with the use of MUTE – failure
Purpose	Insure that the protocol stack and TAP still works after the CTB implementation. The test uses the MUTE functionality in the TAP. MUTE should fail.
Test Description	Write a test application that tests sending / receiving of a primitive with the use of MUTE. The test should receive something in the MUTE time, so that the test case fails (will be trapped and turned into pass, otherwise the case will fail).
TDC test case nr:	XX_TDC400D
Success Criteria	Test case should pass.

2.2.5 With the use of MUTE - pass

Title	Running normal – with the use of MUTE – pass
Purpose	Insure that the protocol stack and TAP still works after the CTB implementation. The test uses the MUTE functionality in the TAP. This time it passes.
Test Description	Write a test application that tests sending / receiving of a primitive with the use of MUTE. The MUTE time is ended without receiving something (pass).
TDC test case nr:	XX_TDC400E
Success Criteria	Test case should pass.

2.2.6 Disabling of CTB although already disabled

Title	Running normal – trying to disable CTB
Purpose	Insure that PS and TAP works when trying to disable CTB although it already is disabled.
Test Description	Write a test application that disables CTB although it already is disabled. The disabling should be done in between sending and receiving of primitives.
TDC test case nr:	XX_TDC400F

Success Criteria	Test case should pass.

2.2.7 Illustrating the need for CTB

Title	Running normal – failing because missing CTB
Purpose	Illustrate the need for CTB.
Test Description	Write a test application that forces the stack to run in a dummy loop x times. The number of loop executions should be controllable from the test case. Because of the executions in the stack nothing is sent back until the TAP has performed a timeout. In principle the test case should fail, but the error should be trapped, so that the test case passes.
TDC test case nr:	XX_TDC400G
Success Criteria	Test case should pass.

3 Testing with CTB enabled

3.1 Straight forward cases

3.1.1 Running with CTB enabled

Title	All previous test cases should be executed with CTB enabled, except XX_TDC400F.
Purpose	Insure that “old” test cases also works with CTB enabled.
Test Description	See previous descriptions
TDC test case nr:	XX_TDC401A->E
Success Criteria	Test cases should pass.

3.1.2 Illustrating that CTB works

Title	Illustration of CTB functionality
Purpose	Illustrate that previous test case, which “failed” without CTB now passes – e.g. CTB works.
Test Description	Write a test application that forces the stack to run in a dummy loop x times. The number of loop executions should be controllable from the test case. Because of the executions in the stack nothing is sent back until the TAP has performed a timeout. However the TAP will wait until, the STACK has performed the ticks and the TAP receives the TIMER_TICK_CNF. The test case will pass, because the time passed is matching the requesting time.
TDC test case nr:	XX_TDC401F
Success Criteria	Test case should pass.

3.1.3 Testing that the timing “limits” is correct in CTB

Title	Test that the timing limits is correct in CTB.
Purpose	Verify that the timing values (limits work correctly) – failure case.
Test Description	Write a test application that starts a timer on the stack side with 9999 seconds. After expiring of this, a primitive should be sent back to the stack. Do the same again with a timer value of 10001. This should cause the TAP to fail with the default timeout of 10000 The error should be trapped, so that the test case verdict is passed.
TDC test case nr:	XX_TDC401H
Success Criteria	Test case should pass.

3.1.4 Testing that the timing “limits” is correct in CTB

Title	Test that the timing limits is correct in CTB
Purpose	Verify that the timing values (limits work correctly) – pass case.

Test Description	Same as 3.1.3, but after the timer of 10001 second is started a TIMEOUT of one millisecond is started. This should cause the test case to pass.
TDC test case nr:	XX_TDC401I
Success Criteria	Test case should pass.

3.1.5 Testing that the CTB store is working

Title	Testing of the CTB store.
Purpose	Verify that storing of messages in the ctb_await routine works.
Test Description	Write a test application that sends 10 primitives (two different types should be used) to the stack, before awaiting the 10 primitives. This should invoke the use of the internal CTB store.
TDC test case nr:	XX_TDC401J
Success Criteria	Test case should pass.

3.1.6 Testing that the CTB store is working

Title	Testing of the CTB store, with disabling of CTB
Purpose	Verify that storing of messages in the ctb_await routine works. E.g. the maximum of stored messages is reached. The TAP should fail.
Test Description	Write a test application that sends 20 primitives to the stack, before awaiting the primitives. This should cause a CTB overflow in the TAP and an exit with error.
TDC test case nr:	XX_TDC401K
Success Criteria	Test case should fail.

3.2 Enabling/disabling of CTB

3.2.1 Testing enabling/disabling of CTB with stored primitives

Title	Testing of enabling/disabling of CTB, when primitives are stored.
Purpose	Verify that storing of messages in the ctb_await routine works, when CTB is disabled when primitives are stored.
Test Description	Write a test application that disables CTB in the beginning. Let it send 10 primitives (two different primitives should be used) to the stack and then enable CTB. Await the 10 primitives afterwards.
TDC test case nr:	XX_TDC402A
Success Criteria	Test case should pass.

3.2.2 Testing enabling/disabling of CTB – timer in tap is used.

Title	Testing of enabling/disabling of CTB – timer in tap is used.
Purpose	Verify that enabling/disabling of CTB with a started timer in the TAP works. Starting with CTB enabled.

Test Description	Write a test application that starts a timer (START_TIMEOUT) in the TAP. Disable CTB and call WAIT_TIMEOUT. This should work fine. Enabled CTB again and call START_TIMEOUT. Disable CTB and do some processing to ensure that the TIMER will expire. Call START_TIMEOUT, which then should fail. The error should be trapped, so that the case passes.
TDC test case nr:	XX_TDC402B
Success Criteria	Test case should pass.

3.2.3 Testing enabling/disabling of CTB – timer in tap is used.

Title	Testing of enabling/disabling of CTB – timer in tap is used.
Purpose	Verify that enabling/disabling of CTB with a started timer in the TAP works. Starting with CTB disabled.
Test Description	Write a test application that disables CTB at the beginning. Call START_TIMEOUT for starting of a timer in the TAP. Enable CTB and call WAIT_TIMEOUT. This should work fine. Disabled CTB again and call START_TIMEOUT. Do some processing to ensure that the TIMER is expired. Enable CTB and call START_TIMEOUT, which then should fail. The error should be trapped, so that the case passes.
TDC test case nr:	XX_TDC402C
Success Criteria	Test case should pass.

3.2.4 Testing enabling/disabling of CTB – between reception of primitives

Title	Testing of enabling/disabling of CTB – between reception of primitives.
Purpose	Verify that enabling/disabling of CTB between sending / awaiting primitives works.
Test Description	Write a test application that sends to primitives to the stack. Await one of them, disable CTB, and await the second primitive. Send to primitives to stack, await one of them, enable CTB and await the last one.
TDC test case nr:	XX_TDC402D
Success Criteria	Test case should pass.

3.2.5 Testing enabling of CTB, when already enabled

Title	Testing of enabling CTB, when already enabled.
Purpose	Verify that enabling of CTB, when already enabled, doesn't influence the testing.
Test Description	Write a test application that enables CTB, although it already is enabled.
TDC test case nr:	XX_TDC402E
Success Criteria	Test case should pass.

3.3 Special cases.

3.3.1 Testing that TAP fails when stack "crashes"

Title	Stack crashing should cause TAP failure
Purpose	Test the TAP2 registers a crash of the PS.
Test Description	Write a test application that forces the stack to crash or exit. When the TAP discovers that the process has been "killed" the TAP should fail. The error should be trapped, so that the test case verdict is passed.
TDC test case nr:	XX_TDC403A
Success Criteria	Test case should pass.

3.3.2 Testing CTB when debugging of stack

Title	Stack debugging should not cause TAP failure
Purpose	Verify that the test case doesn't fail, when CTB is enabled and the stack is being debugged.
Test Description	Write a test application that sends and receives primitives in a normal way. Insert a breakpoint some where in stack, to force the TAP to do "a timeout". However until the TIMER_TICK_CNF is received the TAP will not fail. When the stack is running again the TAP should just continue testing and the testcase should pass.
TDC test case nr:	XX_TDC403B
Success Criteria	Test case should pass.