



Technical Document – Confidential

GSM PROTOCOL STACK

TEST SPECIFICATION

PKTIO

Document Number:	8443.407.02.002
Version:	0.3
Status:	Draft
Approval Authority:	
Creation Date:	2002-Oct-15
Last changed:	2015-Mar-08 by XGUTTEFE
File Name:	pktio.doc

Important Notice

Texas Instruments Incorporated and/or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and/or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and/or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and/or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and/or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and/or the licensing of software does not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third-party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, including photocopying and recording, for any purpose without the express written permission of TI.

Change History

Date	Changed by	Approved by	Version	Status	Notes
2002-Oct-15	HM		0.1		1
2003-Apr-03	HM		0.2		2
2003-May-22	XGUTTEFE		0.3	Draft	

Notes:

1. Initial version
2. Fixed typo

Table of Contents

1.1	Terms.....	6
2	Overview.....	6
2.1	GRR (RLC/MAC) – Radio Link Control/Medium Access Control	7
2.2	LLC – Logical Link Control	7
2.3	GMM – GPRS Mobility Management	7
2.4	SM – Session Management.....	7
2.5	SNDCP - Subnetwork Dependant Convergence Protocol.....	7
2.6	GACI – GPRS Application Control Interface	7
2.7	USART - Universal Synchronous Asynchronous Receiver Transmitter Driver.....	7
2.8	TOM – Tunnelling of Messages.....	8
3	Parameters	8
4	TEST CASES.....	16
4.1	Internal Routing.....	16
4.1.1	PKTIO000: Setup the routing and PCO view for the PKTIO tests.....	16
4.2	Connection phase	16
4.2.1	PKTIO010: Connection of first device requested	16
4.2.2	PKTIO011: Response to connection of first device - positive.....	17
4.2.3	PKTIO012: Response to connection of first device negative.....	18
4.2.4	PKTIO013: Received DRV_SIGTYPE_DISCONNECT in conn. phase	19
4.2.5	PKTIO014: Clash of DISCONNECT signal and PKT_CONNECT_RES.....	20
4.2.6	PKTIO015: Clash of DISCONNECT signal and PKT_CONNECT_REJ	20
4.2.7	PKTIO016: DRV_SIGTYPE_DISCONNECT in state KER_NO_DTI.....	21
4.2.8	PKTIO020: Opening of DTI connection requested.....	21
4.2.9	PKTIO021: Opening of DTI connection confirmed.....	22
4.2.10	PKTIO022: Initial flow control primitive from peer received	23
4.2.11	PKTIO023: Opening of DTI connection rejected.....	23
4.2.12	PKTIO024: DTI disconnection in state KER_READY	24
4.2.13	PKTIO025: KER_DTI_READY, DRV_SIGTYPE_DISCONNECT received	25
4.2.14	PKTIO026: KER_DTI_OPENING, DRV_SIGTYPE_DISCONNECT received	26
4.2.15	PKTIO027: More than one read buffer supported by DIO driver.....	27
4.2.16	PKTIO028: PKT_DTI_CLOSE_REQ received.....	29
4.3	Modifications.....	29
4.3.1	PKTIO050: KER_READY, modification of driver settings immediately possible	29
4.3.2	PKTIO051: KER_READY, Modification of driver settings requested	30
4.3.3	PKTIO052: KER_MODIFY, end of flushing in signalled.....	31
4.3.4	PKTIO053: KER_MODIFY, DTI2_DISCONNECT_REQ.....	33
4.3.5	PKTIO054: KER_MODIFY_NO_DTI, DRV_SIGTYPE_FLUSH signalled.....	33
4.3.6	PKTIO055: KER_MODIFY, DRV_SIGTYPE_DISCONNECT signalled.....	34
4.3.7	PKTIO056: KER_MODIFY_NO_DTI and DRV_SIGTYPE_DISCONNECT	36
4.3.8	PKTIO057: KER_MODIFY / TX_FLUSH_BUFFER, end of flushing	36
4.3.9	PKTIO058: KER_MODIFY / TX_FLUSHING, end of flushing	38
4.3.10	PKTIO059: TX_FLUSHING or TX_FLUSH_BUFFER, packet written	39
4.3.11	PKTIO060: KER_DTI_OPENING, PKT_MODIFY_REQ immediate.....	40
4.3.12	PKTIO061: KER_DTI_OPENING, PKT_MODIFY_REQ after flush	41
4.3.13	PKTIO062: KER_MODIFY_DTI_OPENING, DRV_SIGTYPE_FLUSH	42
4.3.14	PKTIO063: KER_MODIFY_DTI_OPENING, DTI2_DISCONNECT_REQ	43
4.3.15	PKTIO064: KER_MODIFY_DTI_OPENING, PKT_DTI_OPEN_CNF	43
4.3.16	PKTIO065: KER_MODIFY_DTI_OPENING, DRV_SIGTYPE_DISCONNECT	44

4.3.17	PKTIO070: KER_NO_DTI, PKT_MODIFY_REQ immediate.....	45
4.3.18	PKTIO071: KER_DTI_OPENING, PKT_MODIFY_REQ after flush	46
4.4	Connected phase - focus on RX / DTX	47
4.4.1	PKTIO100: KER_READY / DTX_READY, UL packet received.....	47
4.4.2	PKTIO101: Flow control primitive from DTI after DL data tranfer	48
4.4.3	PKTIO102: No flow control primitive from DTI after DL data tranfer	49
4.4.4	PKTIO103: No flow control primitive from DTI after DL data tranfer	50
4.4.5	PKTIO104: DTI becomes ready while one descriptor present in PKTIO	51
4.4.6	PKTIO106: Reception of flow control primitive, no filled read buffer present	52
4.4.7	PKTIO107: Reception of flow control primitive, filled read buffer present	52
4.4.8	PKTIO108: RX_BUFFER, DRV_SIGTYPE_DISCONNECT from DIO	53
4.4.9	PKTIO109: KER_READY, RX_READY, DTX_READY, packet received	54
4.4.10	PKTIO110: KER_READY, RX_READY, DTX_NOT_READY, flow control.....	55
4.4.11	PKTIO111: KER_READY, RX_READY, DTX_NOT_READY, packets received.....	56
4.4.12	PKTIO112: KER_READY, RX_BUFFER, DTX_NOT_READY, flow control	57
4.4.13	PKTIO113: KER_READY, RX_BUFFER, DTX_NOT_READY, flow control	58
4.5	Connected phase - focus on RX / DTX, multi instance tests	60
4.6	Connected phase - focus on TX / DRX	60
4.6.1	PKTIO200: DRX_READY, DL packet received.....	60
4.6.2	PKTIO201: DRX_READY, DL packet received, DIO driver has one packet	61
4.6.3	PKTIO202: DRX_READY, TX_READY, packet written by driver	62
4.6.4	PKTIO203: DRX_NOT_READY, TX_BUFFER, packet written by driver.....	63
4.6.5	PKTIO204: DRX_NOT_READY, TX_BUFFER, disconnect by driver	64
4.6.6	PKTIO205: DRX_READY, TX_FLUSHING, DL packet received.....	65
4.6.7	PKTIO206: DRX_NOT_READY, TX_BUFFER, disconnect by peer.....	65
4.6.8	PKTIO207: DRX_NOT_READY, TX_BUFFER, disconnect req. by ACI	66
4.6.9	PKTIO208: DRX_DEAD_NOT_READY, TX_BUFFER.....	67
4.6.10	PKTIO209: DRX_DEAD_NOT_READY, TX_BUFFER.....	68
4.6.11	PKTIO210: DRX_DEAD_READY, TX_READY, driver disconnects.....	69
4.6.12	PKTIO211: DRX_READY, DL packet received, DIO driver has one packet	70
4.7	Config primitives	71
4.7.1	PKTIO301: Config Primitive SEND.....	71
4.7.2	PKTIO302: Driver returns correct looped back data after config primitive SEND.....	71
4.7.3	PKTIO303: Driver returns looped back data after config primitive SEND, wrong length	72
4.7.4	PKTIO304: Driver returns looped back data after config primitive SEND, length is ok but content is wrong	74
4.7.5	PKTIO305: Second Config Primitive SEND.....	75
4.7.6	PKTIO306: Flow control primive from peer received	75
4.7.7	PKTIO307: Driver returns second correct looped back data after config primitive SEND	76
5	Suites.....	77
	Appendices.....	82
	A. Acronyms.....	82
	B. Glossary.....	82

List of Figures and Tables

List of References



[ISO 9000:2000]

International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1.1 Terms

- Entity: Program which executes the functions of a layer
- Message: A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.
- Primitive: A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.
- Service Access Point: A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).

2 Overview

The Protocol Stacks are used to define the functionality of the GSM protocols for interfaces. The GSM specifications are normative when used to describe the functionality of interfaces, but the stacks and the subdivision of protocol layers does not imply or restrict any implementation.

The protocol stack for GPRS consists of several entities. Each entity has one or more service access points, over which the entity provides a service for the upper entity.

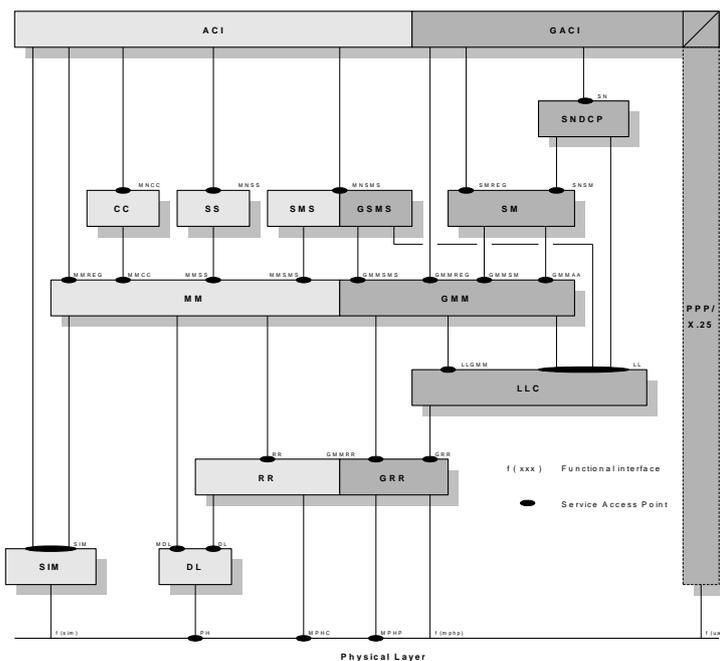


Figure 2-1: Architecture of the GSM/GPRS protocol stack

The information units passed via the SAPs are called primitives and consists of an operation code and several parameters. See the Users Guide for details.

The entities of the GPRS protocol stack are:

2.1 GRR (RLC/MAC) – Radio Link Control/Medium Access Control

This layer contains two functions: The Radio Link Control function provides a radio-solution-dependent reliable link. The Medium Access Control function controls the access signalling (request and grant) procedures for the radio channel, and the mapping of LLC frames onto the GSM physical channel.

2.2 LLC – Logical Link Control

The LLC entity provides multiple highly reliable logical links for a synchronous data transfer between the MS and the network. It supports variable-length information frames, acknowledged and unacknowledged data transfer, flow and sequence control, error detection and recovery, notification of unrecoverable errors, user identity confidentiality, and ciphering of user and signaling data.

2.3 GMM – GPRS Mobility Management

The GMM entity provides procedures for the mobility of the MS, such as informing the network of its present location, and user identity confidentiality. It manages the GMM context (attach, detach, routing area updating), supports security functions such as authentication of user and MS, controls ciphering of data, and initiates the response to paging messages.

2.4 SM – Session Management

The main function of the session management (SM) is to support PDP context handling of the user terminal. Session Management activates, modifies and deletes the contexts for packet data protocols (PDP). Session Management services are provided at the SMREG-SAP and the SNSM-SAP for anonymous and non-anonymous access. The non-anonymous and anonymous access procedures for PDP context activation and PDP context deactivation are available at the SMREG-SAP. In addition there exists a PDP context modification for non-anonymous PDP contexts.

2.5 SNDCP - Subnetwork Dependant Convergence Protocol

SNDCP carries out all functions related to transfer of Network Layer Protocol Data Units (N-PDUs) over GPRS in a transparent way. SNDCP helps to improve channel efficiency by means of compression techniques. The set of protocol entities above SNDCP consists of commonly used network protocols. They all use the same SNDCP entity, which then performs multiplexing of data coming from different sources to be sent using the service provided by the LLC layer.

2.6 GACI – GPRS Application Control Interface

The GACI is the GPRS extension of the ACI. It is specified in GSM 07.07 and 07.60. It is responsible for processing of the GPRS related AT Commands to setup, activate and deactivate the PDP context parameter. It also provides functionality for the interworking between GMM/SM/SNDCP and a packet oriented protocol like PPP.

2.7 USART - Universal Synchronous Asynchronous Receiver Transmitter Driver

The USART is a hardware component that facilitates a connection between the mobile station and terminal equipment (e.g. a PC). This interface uses some of the circuits described in V.24.

The data exchange provided by this unit is serial and asynchronous (synchronous communication is not in the scope of this document). A driver that uses interrupts to manage a circular buffer for the sending and receiving direction is necessary in order to use this component in the GPRS. The driver has to be able to perform flow control.

2.8 TOM – Tunnelling of Messages

The TOM entity is present if and only if HS136 is supported (the feature flag FF_HS136 is enabled).

The main function of TOM is to tunnel non-GSM signalling messages between the MS and the SGSN. The only non-GSM signalling which is currently supported by TOM is for the EGPRS-136 system (according to TIA/EIA-136-376). Data transfer in both uplink and downlink direction is possible. Two different priorities (high, low) of signalling data transfer are supported. TOM uses the unacknowledged mode of LLC and the acknowledged mode of GRR (RLC/MAC).

3 Parameters

FIELD (AN_EXAMPLE_FIELD)

0x00, 0x01

ENDFIELD (AN_EXAMPLE_FIELD, 2)

/*

* Some SDU test data

*/

FIELD (A_UL_SDU)

0x00, 0x03,

/* Length in bits*/

0x00, 0x00,

/* Offset in bits*/

0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E, 0x1F,
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27,
0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E, 0x2F,
0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E, 0x3F,
0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47,
0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F,
0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57,
0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F

/* 96 Data bytes*/

ENDFIELD (A_UL_SDU, 100)

FIELD (A_FIRST_UL_SDU)

0x40, 0x03,

/* Length in bits*/

0x00, 0x00,

/* Offset in bits*/

0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E, 0x1F,
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27,
0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E, 0x2F,
0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E, 0x3F,
0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47,
0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F,
0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57,
0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F

/* 104 Data bytes*/

ENDFIELD (A_FIRST_UL_SDU, 108)

FIELD (A_SECOND_UL_SDU)

0x40, 0x03,

/* Length in bits*/

0x00, 0x00,

/* Offset in bits*/

0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02,
0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E, 0x1F,
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27,

/* 104 Data bytes*/

```
0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E, 0x2F,
0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E, 0x3F,
0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47,
0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F,
0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57,
0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F
ENDFIELD (A_SECOND_UL_SDU, 108)
FIELD (A_THIRD_UL_SDU)
0x40, 0x03, /* Length in bits */
0x00, 0x00, /* Offset in bits */
0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03, /* 104 Data bytes*/
0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E, 0x1F,
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27,
0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E, 0x2F,
0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E, 0x3F,
0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47,
0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F,
0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57,
0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F
ENDFIELD (A_THIRD_UL_SDU, 108)
FIELD (A_DL_SDU)
0x00, 0x03, /* Length in bits */
0x00, 0x00, /* Offset in bits */
0x5F, 0x5E, 0x5D, 0x5C, 0x5B, 0x5A, 0x59, 0x58, /* 96 Data bytes*/
0x57, 0x56, 0x55, 0x54, 0x53, 0x52, 0x51, 0x50,
0x4F, 0x4E, 0x4D, 0x4C, 0x4B, 0x4A, 0x49, 0x48,
0x47, 0x46, 0x45, 0x44, 0x43, 0x42, 0x41, 0x40,
0x3F, 0x3E, 0x3D, 0x3C, 0x3B, 0x3A, 0x39, 0x38,
0x37, 0x36, 0x35, 0x34, 0x33, 0x32, 0x31, 0x30,
0x2F, 0x2E, 0x2D, 0x2C, 0x2B, 0x2A, 0x29, 0x28,
0x27, 0x26, 0x25, 0x24, 0x23, 0x22, 0x21, 0x20,
0x1F, 0x1E, 0x1D, 0x1C, 0x1B, 0x1A, 0x19, 0x18,
0x17, 0x16, 0x15, 0x14, 0x13, 0x12, 0x11, 0x10,
0x0F, 0x0E, 0x0D, 0x0C, 0x0B, 0x0A, 0x09, 0x08,
0x07, 0x06, 0x05, 0x04, 0x03, 0x02, 0x01, 0x00
ENDFIELD (A_DL_SDU, 100)
FIELD (A_FIRST_DL_SDU)
0x40, 0x03, /* Length in bits */
0x00, 0x00, /* Offset in bits */
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, /* 104 Data bytes*/
0x5F, 0x5E, 0x5D, 0x5C, 0x5B, 0x5A, 0x59, 0x58,
0x57, 0x56, 0x55, 0x54, 0x53, 0x52, 0x51, 0x50,
0x4F, 0x4E, 0x4D, 0x4C, 0x4B, 0x4A, 0x49, 0x48,
0x47, 0x46, 0x45, 0x44, 0x43, 0x42, 0x41, 0x40,
0x3F, 0x3E, 0x3D, 0x3C, 0x3B, 0x3A, 0x39, 0x38,
0x37, 0x36, 0x35, 0x34, 0x33, 0x32, 0x31, 0x30,
0x2F, 0x2E, 0x2D, 0x2C, 0x2B, 0x2A, 0x29, 0x28,
0x27, 0x26, 0x25, 0x24, 0x23, 0x22, 0x21, 0x20,
0x1F, 0x1E, 0x1D, 0x1C, 0x1B, 0x1A, 0x19, 0x18,
0x17, 0x16, 0x15, 0x14, 0x13, 0x12, 0x11, 0x10,
0x0F, 0x0E, 0x0D, 0x0C, 0x0B, 0x0A, 0x09, 0x08,
0x07, 0x06, 0x05, 0x04, 0x03, 0x02, 0x01, 0x00
ENDFIELD (A_FIRST_DL_SDU, 108)
```

FIELD (A_SECOND_DL_SDU)
0x40, 0x03,
0x00, 0x00,
0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02,
0x5F, 0x5E, 0x5D, 0x5C, 0x5B, 0x5A, 0x59, 0x58,
0x57, 0x56, 0x55, 0x54, 0x53, 0x52, 0x51, 0x50,
0x4F, 0x4E, 0x4D, 0x4C, 0x4B, 0x4A, 0x49, 0x48,
0x47, 0x46, 0x45, 0x44, 0x43, 0x42, 0x41, 0x40,
0x3F, 0x3E, 0x3D, 0x3C, 0x3B, 0x3A, 0x39, 0x38,
0x37, 0x36, 0x35, 0x34, 0x33, 0x32, 0x31, 0x30,
0x2F, 0x2E, 0x2D, 0x2C, 0x2B, 0x2A, 0x29, 0x28,
0x27, 0x26, 0x25, 0x24, 0x23, 0x22, 0x21, 0x20,
0x1F, 0x1E, 0x1D, 0x1C, 0x1B, 0x1A, 0x19, 0x18,
0x17, 0x16, 0x15, 0x14, 0x13, 0x12, 0x11, 0x10,
0x0F, 0x0E, 0x0D, 0x0C, 0x0B, 0x0A, 0x09, 0x08,
0x07, 0x06, 0x05, 0x04, 0x03, 0x02, 0x01, 0x00
/* Length in bits */
/* Offset in bits */
/* 104 Data bytes*/
ENDFIELD (A_SECOND_DL_SDU, 108)
FIELD (A_THIRD_DL_SDU)
0x40, 0x03,
0x00, 0x00,
0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03,
0x5F, 0x5E, 0x5D, 0x5C, 0x5B, 0x5A, 0x59, 0x58,
0x57, 0x56, 0x55, 0x54, 0x53, 0x52, 0x51, 0x50,
0x4F, 0x4E, 0x4D, 0x4C, 0x4B, 0x4A, 0x49, 0x48,
0x47, 0x46, 0x45, 0x44, 0x43, 0x42, 0x41, 0x40,
0x3F, 0x3E, 0x3D, 0x3C, 0x3B, 0x3A, 0x39, 0x38,
0x37, 0x36, 0x35, 0x34, 0x33, 0x32, 0x31, 0x30,
0x2F, 0x2E, 0x2D, 0x2C, 0x2B, 0x2A, 0x29, 0x28,
0x27, 0x26, 0x25, 0x24, 0x23, 0x22, 0x21, 0x20,
0x1F, 0x1E, 0x1D, 0x1C, 0x1B, 0x1A, 0x19, 0x18,
0x17, 0x16, 0x15, 0x14, 0x13, 0x12, 0x11, 0x10,
0x0F, 0x0E, 0x0D, 0x0C, 0x0B, 0x0A, 0x09, 0x08,
0x07, 0x06, 0x05, 0x04, 0x03, 0x02, 0x01, 0x00
/* Length in bits */
/* Offset in bits */
/* 104 Data bytes*/
ENDFIELD (A_THIRD_DL_SDU, 108)
FIELD (CONFIG_64_SDU)
0x00, 0x02,
0x00, 0x00,
0x00, 0x00, 0x00, 0x01, 0x00, 0x02, 0x00, 0x03,
0x00, 0x04, 0x00, 0x05, 0x00, 0x06, 0x00, 0x07,
0x00, 0x08, 0x00, 0x09, 0x00, 0x0a, 0x00, 0x0b,
0x00, 0x0c, 0x00, 0x0d, 0x00, 0x0e, 0x00, 0x0f,
0x00, 0x10, 0x00, 0x11, 0x00, 0x12, 0x00, 0x13,
0x00, 0x14, 0x00, 0x15, 0x00, 0x16, 0x00, 0x17,
0x00, 0x18, 0x00, 0x19, 0x00, 0x1a, 0x00, 0x1b,
0x00, 0x1c, 0x00, 0x1d, 0x00, 0x1e, 0x00, 0x1f
/* Length in bits */
/* Offset in bits */
ENDFIELD (CONFIG_64_SDU, 68)
FIELD (CONFIG_32_SDU)
0x00, 0x01,
0x00, 0x00,
0x00, 0x00, 0x00, 0x01, 0x00, 0x02, 0x00, 0x03,
0x00, 0x04, 0x00, 0x05, 0x00, 0x06, 0x00, 0x07,
0x00, 0x08, 0x00, 0x09, 0x00, 0x0a, 0x00, 0x0b,
0x00, 0x0c, 0x00, 0x0d, 0x00, 0x0e, 0x00, 0x0f
/* Length in bits */
/* Offset in bits */
ENDFIELD (CONFIG_32_SDU, 36)
FIELD (CONFIG_64_WRONG_CONTENT)
0x00, 0x02,
0x00, 0x00,
0x00, 0x00, 0x00, 0x01, 0x00, 0x02, 0x00, 0x03,
0x00, 0x04, 0x00, 0x05, 0x00, 0x06, 0x00, 0x07,

```
0x00, 0x08, 0x00, 0x09, 0x00, 0x0a, 0x00, 0x0b,
0x00, 0x0c, 0x00, 0x0d, 0x00, 0x0e, 0x00, 0x0f,
0x00, 0x10, 0x00, 0x11, 0x00, 0x12, 0x00, 0x13,
0x00, 0x14, 0x00, 0x15, 0x00, 0x16, 0x00, 0x17,
0x00, 0x18, 0x00, 0x19, 0x00, 0x1a, 0x00, 0x1b,
0x00, 0x1c, 0x00, 0x1d, 0x00, 0x1e, 0x00, 0xff
ENDFIELD (CONFIG_64_WRONG_CONTENT, 68)
FIELD (CONFIG_WRONG_LEN)
    0x40, 0x00, /* Length in bits */
    0x00, 0x00, /* Offset in bits */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (CONFIG_WRONG_LEN, 12)
FIELD (CONFIG_WRONG_CONTENT)
    0x40, 0x00, /* Length in bits */
    0x00, 0x00, /* Offset in bits */
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff
ENDFIELD (CONFIG_WRONG_CONTENT, 12)
DECLARATION (AN_EXAMPLE_ARRAY)
DECLARATION (DEFAULT_DCB_FROM_DIO)
DECLARATION (DEFAULT_DCB_FROM_ACI)
DECLARATION (MODIFIED_DCB_FROM_ACI)
DECLARATION (TRASH_DCB_FROM_ACI)
/* DECLARATION (AN_EXAMPLE_MSTRUCT) */
DECLARATION (AN_EXAMPLE_PSTRUCT_ARRAY)

/*
 * Simple ordinal numbers from zero to nine, type BYTE
 */
BYTE    BYTE_0        0
BYTE    BYTE_1        1
BYTE    BYTE_2        2
BYTE    BYTE_3        3
BYTE    BYTE_4        4
BYTE    BYTE_5        5
BYTE    BYTE_6        6
BYTE    BYTE_7        7
BYTE    BYTE_8        8
BYTE    BYTE_9        9
/*
 * Device numbers
 */
/*
BYTE    DEVICE_0       0
BYTE    DEVICE_1       1
BYTE    DEVICE_2       2
BYTE    DEVICE_3       3
BYTE    DEVICE_4       4
BYTE    DEVICE_5       5
BYTE    DEVICE_6       6
BYTE    DEVICE_7       7
BYTE    DEVICE_8       8
BYTE    DEVICE_9       9
*/
BYTE    DEVICE_47      0x47    /* No low device numbers guaranteed */
BYTE    DEVICE_66      0x66    /* Some other device */

BYTE    DTI_DIRECTION_NORMAL    0x00
BYTE    DTI_DIRECTION_INVERTED  0x01
```

```
/*  
 * Simple ordinal numbers from zero to nine, type SHORT  
 */  
SHORT   SHORT_0           0x0000  
SHORT   SHORT_1           0x0001  
SHORT   SHORT_2           0x0002  
SHORT   SHORT_3           0x0003  
SHORT   SHORT_4           0x0004  
SHORT   SHORT_5           0x0005  
SHORT   SHORT_6           0x0006  
SHORT   SHORT_7           0x0007  
SHORT   SHORT_8           0x0008  
SHORT   SHORT_9           0x0009
```

```
/*  
 * Maximum transmission unit  
 */  
SHORT   MTU_1500          1500  
SHORT   MTU_512           512  
SHORT   MTU_TRASH         10000
```

```
/*  
 * Return values as defined in gdi.h  
 */  
SHORT   DRV_OK             0x0000  
SHORT   DRV_BUFFER_FULL   0x0001  
SHORT   DRV_DISABLED       0x0002  
SHORT   DRV_ENABLED        0x0003  
SHORT   DRV_INITFAILURE    0x0004  
SHORT   DRV_INITIALIZED    0x0005  
SHORT   DRV_INTERNAL_ERROR 0x0006  
SHORT   DRV_INPROCESS      0x0007  
SHORT   DRV_INVALID_PARAMS 0x0008  
SHORT   DRV_NOTCONFIGURED  0x0009
```

```
/*  
 * Return values as defined in gdi.h  
 */  
SHORT   DRV_BUFTYPE_READ   0x0001  
SHORT   DRV_BUFTYPE_WRITE  0x0002
```

```
/*  
 * Signal events as defined in gdi.h  
 */  
SHORT   DRV_SIGTYPE_CLEAR  0x0002  
SHORT   DRV_SIGTYPE_FLUSH   0x0004  
SHORT   DRV_SIGTYPE_READ    0x0008  
SHORT   DRV_SIGTYPE_WRITE   0x0010  
SHORT   DRV_SIGTYPE_CONNECT 0x0020  
SHORT   DRV_SIGTYPE_DISCONNECT 0x0040
```

```
/*  
 * Simple ordinal numbers from zero to nine, type LONG  
 */  
LONG    LONG_0             0x00000000  
LONG    LONG_1             0x00000001  
LONG    LONG_2             0x00000002  
LONG    LONG_3             0x00000003  
LONG    LONG_4             0x00000004
```

```
LONG    LONG_5          0x00000005
LONG    LONG_6          0x00000006
LONG    LONG_7          0x00000007
LONG    LONG_8          0x00000008
LONG    LONG_9          0x00000009
```

```
/*
 * All baud rates allowed on shared memory except DIO_BAUD_AUTO
 */
```

```
LONG    DIO_BAUD_SHAREDMEM    0x000FFFFE
```

```
/*
 * The DTI peer. This is casted to a pointer to a C-String in real life, so it is impossible to set this
 * to a meaningful value here in the simulation environment. This restriction has to be handled in
 * the C-Code of the entity using a conditional compile switch.
 */
```

```
LONG    PEER_SNDP        0x47110000
```

```
/*
 * The DTI link identifier. Here an arbitrary value
 */
```

```
LONG    LINK_ID_47      0x12345647
LONG    LINK_ID_66      0x12345666
```

```
/*
 * MASK and STATE for DIO_Write(). These are not used in PKTIO and therefore shall be 0.
 */
```

```
LONG    DIO_STATE_NONE  0x00000000
LONG    DIO_STATE_IP    0x00000021
LONG    DIO_MASK_NONE   0x00000000
```

```
/*
 * ARRAYs
 */
```

```
BEGINARRAY (AN_EXAMPLE_ARRAY, 3)
           0x00, 0x01, 0x02
```

```
ENDARRAY
```

```
/*
 * PSTRUCTs
 */
```

```
BEGIN_PSTRUCT ("dio_dcb", DEFAULT_DCB_FROM_DIO)
    SET_COMP ("convergence", DIO_CONV_PACKET)
    SET_COMP ("data_mode", DIO_MODE_DATA)
    SET_COMP ("sleep_mode", DIO_SLEEP_DISABLE)
    SET_COMP ("mux_configuration", LONG_0)
    SET_COMP ("n1", BYTE_0)
    SET_COMP ("n2", BYTE_0)
    SET_COMP ("t1", BYTE_0)
    SET_COMP ("t2", BYTE_0)
    SET_COMP ("t3", BYTE_0)
    SET_COMP ("k", BYTE_0)
    SET_COMP ("mtu", MTU_1500)
    SET_COMP ("baud", DIO_BAUD_SHAREDMEM)
    SET_COMP ("data_bits", DIO_CHAR_8) /* Some defaults */
    SET_COMP ("stop_bits", DIO_STOP_1) /* Some defaults */
    SET_COMP ("parity", DIO_PARITY_NO) /* Some defaults */
    SET_COMP ("flow_control", DIO_FLOW_HW_HW) /* Shared memory is HW */
    SET_COMP ("xon", BYTE_0)
    SET_COMP ("xoff", BYTE_0)
    SET_COMP ("esc_char", BYTE_0)
```

```
        SET_COMP ("guard_period", DIO_ESC_OFF) /* No guard pattern detection */  
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("dio_dcb", DEFAULT_DCB_FROM_ACI)  
    SET_COMP ("convergence", DIO_CONV_PACKET)  
    SET_COMP ("data_mode", DIO_MODE_DATA)  
    SET_COMP ("sleep_mode", DIO_SLEEP_DISABLE)  
    SET_COMP ("mux_configuration", LONG_0)  
    SET_COMP ("n1", BYTE_0)  
    SET_COMP ("n2", BYTE_0)  
    SET_COMP ("t1", BYTE_0)  
    SET_COMP ("t2", BYTE_0)  
    SET_COMP ("t3", BYTE_0)  
    SET_COMP ("k", BYTE_0)  
    SET_COMP ("mtu", MTU_1500)  
    SET_COMP ("baud", DIO_BAUD_812500) /* Negotiated highest common */  
    SET_COMP ("data_bits", DIO_CHAR_8)  
    SET_COMP ("stop_bits", DIO_STOP_1)  
    SET_COMP ("parity", DIO_PARITY_NO)  
    SET_COMP ("flow_control", DIO_FLOW_HW_HW)  
    SET_COMP ("xon", BYTE_0)  
    SET_COMP ("xoff", BYTE_0)  
    SET_COMP ("esc_char", BYTE_0)  
    SET_COMP ("guard_period", DIO_ESC_OFF)  
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("dio_dcb", MODIFIED_DCB_FROM_ACI)  
    SET_COMP ("convergence", DIO_CONV_PACKET)  
    SET_COMP ("data_mode", DIO_MODE_DATA)  
    SET_COMP ("sleep_mode", DIO_SLEEP_DISABLE)  
    SET_COMP ("mux_configuration", LONG_0)  
    SET_COMP ("n1", BYTE_0)  
    SET_COMP ("n2", BYTE_0)  
    SET_COMP ("t1", BYTE_0)  
    SET_COMP ("t2", BYTE_0)  
    SET_COMP ("t3", BYTE_0)  
    SET_COMP ("k", BYTE_0)  
    SET_COMP ("mtu", MTU_512)  
    SET_COMP ("baud", DIO_BAUD_812500) /* Negotiated highest common */  
    SET_COMP ("data_bits", DIO_CHAR_8)  
    SET_COMP ("stop_bits", DIO_STOP_1)  
    SET_COMP ("parity", DIO_PARITY_NO)  
    SET_COMP ("flow_control", DIO_FLOW_HW_HW)  
    SET_COMP ("xon", BYTE_0)  
    SET_COMP ("xoff", BYTE_0)  
    SET_COMP ("esc_char", BYTE_0)  
    SET_COMP ("guard_period", DIO_ESC_OFF)  
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("dio_dcb", TRASH_DCB_FROM_ACI)  
    SET_COMP ("convergence", DIO_CONV_PACKET)  
    SET_COMP ("data_mode", DIO_MODE_DATA)  
    SET_COMP ("sleep_mode", DIO_SLEEP_DISABLE)  
    SET_COMP ("mux_configuration", LONG_0)  
    SET_COMP ("n1", BYTE_0)  
    SET_COMP ("n2", BYTE_0)  
    SET_COMP ("t1", BYTE_0)  
    SET_COMP ("t2", BYTE_0)  
    SET_COMP ("t3", BYTE_0)
```

```
    SET_COMP ("k", BYTE_0)
    SET_COMP ("mtu", MTU_TRASH)
    SET_COMP ("baud", DIO_BAUD_812500) /* Negotiated highest common */
    SET_COMP ("data_bits", DIO_CHAR_8)
    SET_COMP ("stop_bits", DIO_STOP_1)
    SET_COMP ("parity", DIO_PARITY_NO)
    SET_COMP ("flow_control", DIO_FLOW_HW_HW)
    SET_COMP ("xon", BYTE_0)
    SET_COMP ("xoff", BYTE_0)
    SET_COMP ("esc_char", BYTE_0)
    SET_COMP ("guard_period", DIO_ESC_OFF)
ENDSTRUCT

/*
 * MSTRUCTs
 */
/*
BEGIN_MSTRUCT("an_example_mstruct", AN_EXAMPLE_MSTRUCT)
ENDSTRUCT
*/

/*
 * PSTRUCT_ARRAYs
 */
/*
BEGIN_PSTRUCT_ARRAY (AN_EXAMPLE_PSTRUCT_ARRAY, 1)
    AN_EXAMPLE_PSTRUCT
ENDARRAY
*/
```

4 TEST CASES

4.1 Internal Routing

4.1.1 PKTIO000: Setup the routing and PCO view for the PKTIO tests

Description: Routings for the PKTIO tests are set.

Preamble: None

DIO	PKTIO	ACI
COMMAND (TAP RESET)		
COMMAND (MMI RESET)		
COMMAND (PKT RESET)		
COMMAND (SND RESET)		
COMMAND (TAP REDIRECT CLEAR)		
COMMAND (PKT REDIRECT CLEAR)		
COMMAND (MMI REDIRECT PKT NULL)		
COMMAND (PKT REDIRECT MMI TAP)		
COMMAND (PKT REDIRECT SND TAP)		
COMMAND (TAP REDIRECT TAP PKT)		

Parametrization

Primitive	Parameter	Value
-----------	-----------	-------

History: 15-Oct-02 HM Initial

4.2 Connection phase

4.2.1 PKTIO010: Connection of first device requested

Description: The DIO driver requests a connection for the first device by issuing the signalling callback with the parameter DRV_SIGTYPE_CONNECT. The PKTIO entity gets the information driver's capabilities using the DIO_GetCapabilities() function and informs ACI about the driver's wish to connect.

<A> Device_47

 Device_66

Variants: <A>.....

Preamble: PKTIO000

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_CONNECT)	
		<=====	
(2)		PKT_DIO_GETCAP_REQ	

```

(3) | | | *=====>*
    | | | | PKT_DIO_GETCAP_CNF |
(4) | | | *<=====*
```

Parametrization

Primitive	Parameter	Value
(1) PKT_DIO_SIGNAL_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
> signal_type	DRV_SIGTYPE_CONNECT	
(2) PKT_DIO_GETCAP_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(3) PKT_DIO_GETCAP_CNF		
retval	DRV_OK	
dio_dcb	DEFAULT_DCB_FROM_DIO	
(4) PKT_CONNECT_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
dio_dcb	DEFAULT_DCB_FROM_DIO	

History: 15-Oct-02 HM Initial

4.2.2 PKTIO011: Response to connection of first device - positive

Description: The PKTIO entity gets a PKT_CONNECT_RES response from ACI. The driver is configured successfully. Only one read buffer supported by the driver.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO010A
 PKTIO010B

```

ACI | | | PKTIO | | | DIO
(1) | | | | | | |
    | | | | | | |
    | | | | | | |
    | | | | | | |
(2) | | | | | | |
    | | | | | | |
(3) | | | | | | |
    | | | | | | |
(4) | | | | | | |
    | | | | | | |
(5) | | | | | | |
    | | | | | | |
(6) | | | | | | |
    | | | | | | |
(7) | | | | | | |
```


Primitive	Parameter	Value
(12) PKT_CONNECT_REJ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(13) PKT_DIO_CLOSEDEVICE_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(14) PKT_DIO_CLOSEDEVICE_CNF		
retval	DRV_OK	

History: 15-Oct-02 HM Initial

4.2.4 PKTIO013: Received DRV_SIGTYPE_DISCONNECT in conn. phase

Description: The PKTIO entity, while awaiting the PKT_CONNECT_RES primitive from ACI, gets a DRV_SIGTYPE_DISCONNECT from the driver. The disconnection is announced to ACI.

<A> Device_47
 Device_66

Variants: <A>....

Preamble: <A> PKTIO010A
 PKTIO010B

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_DISCONNECT)	
		*<=====	
(2)		PKT_DIO_CLOSEDEVICE_REQ	
		===== >	
(3)		PKT_DIO_CLOSEDEVICE_CNF	
		*<=====	
(4)	PKT_DISCONNECT_IND		
	*<=====		
MUTE (1000)			

Parametrization

Primitive	Parameter	Value
(15) PKT_DIO_SIGNAL_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_DISCONNECT	
(16) PKT_DIO_CLOSEDEVICE_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(17) PKT_DIO_CLOSEDEVICE_CNF		
retval	DRV_OK	
(18) PKT_DISCONNECT_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
cause	PKTCS_DISCONNECT	

History: 15-Oct-02 HM Initial

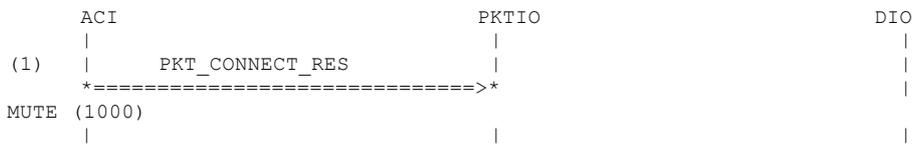
4.2.5 PKTIO014: Clash of DISCONNECT signal and PKT_CONNECT_RES

Description: After the driver indicated the disconnection, PKTIO receives the PKT_CONNECT_RES primitive from ACI. It is expected that nothing happens in this collision case, the primitive from ACI is simply ignored.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO013A
 PKTIO013B



Parametrization

<u>Primitive</u>	<u>Parameter</u>	<u>Value</u>
(19) PKT_CONNECT_RES		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
dio_dcb	DEFAULT_DCB_FROM_ACI	

History: 15-Oct-02 HM Initial

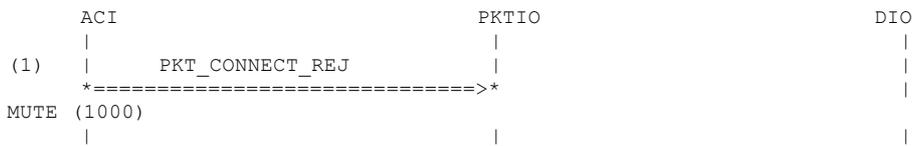
4.2.6 PKTIO015: Clash of DISCONNECT signal and PKT_CONNECT_REJ

Description: After the driver indicated the disconnection, PKTIO receives the PKT_CONNECT_REJ primitive from ACI. No action is expected by the PKTIO entity.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO013A
 PKTIO013B



Parametrization

<u>Primitive</u>	<u>Parameter</u>	<u>Value</u>
(20) PKT_CONNECT_REJ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66

History: 15-Oct-02 HM Initial

4.2.7 PKTIO016: DRV_SIGTYPE_DISCONNECT in state KER_NO_DTI

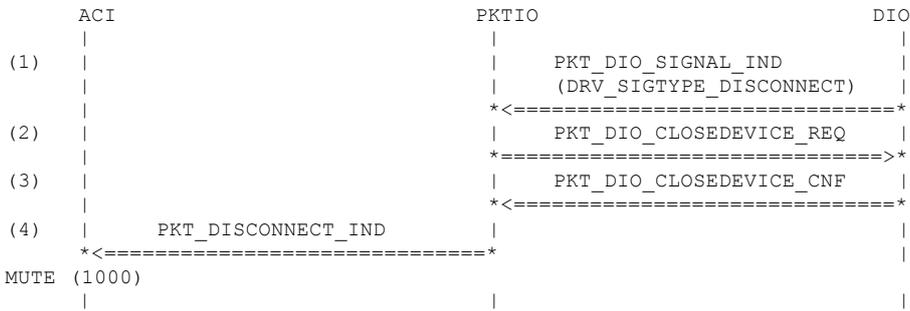
Description: In state KER_NO_DTI the driver sends the DRV_SIGTYPE_DISCONNECT signal. It has to be ensured that the buffer descriptors are got back from the DIO driver.

Note: This is the same as PKTIO013, but with different preamble.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO011A
 PKTIO011B



Parametrization

Primitive	Parameter	Value
(21) PKT_DIO_SIGNAL_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_DISCONNECT	
(22) PKT_DIO_CLOSEDEVICE_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(23) PKT_DIO_CLOSEDEVICE_CNF		
retval	DRV_OK	
(24) PKT_DISCONNECT_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
cause	PKTCS_DISCONNECT	

History: 15-Oct-02 HM Initial

4.2.8 PKTIO020: Opening of DTI connection requested

Description: In state KER_NO_DTI the ACI requests the opening of a DTI connection.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO011A
 PKTIO011B

```

    ACI                               PKTIO                               DIO
    |                                 |                                 |
(1) |   PKT_DTI_OPEN_REQ             |                                 |
    |   *=====>                    |                                 |
(2) |   DTI2_CONNECT_IND             |                                 |
    |   *<=====                     |                                 |
MUTE (1000)
    |                                 |                                 |
    
```

Parametrization

Primitive	Parameter	Value
(25) PKT_DTI_OPEN_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
peer	PEER_SNDPCP	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
dti_direction	DTI_DIRECTION_NORMAL	
(26) DTI2_CONNECT_IND		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
version	DTI_VERSION_10	

History: 15-Oct-02 HM Initial

4.2.9 PKTIO021: Opening of DTI connection confirmed

Description: In state KER_DTI_OPENING the ACI answered positive. The PKTIO entity confirms the opening of the DTI connection. The initial flow control primitive is sent by PKTIO.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO020A
 PKTIO020B

```

    ACI                               PKTIO                               DIO
    |                                 |                                 |
(1) |   DTI2_CONNECT_RES             |                                 |
    |   *=====>                    |                                 |
(2) |   DTI2_READY_IND               |                                 |
    |   *<=====                     |                                 |
(3) |   PKT_DTI_OPEN_CNF             |                                 |
    |   *<=====                     |                                 |
MUTE (1000)
    |                                 |                                 |
    
```

Parametrization

Primitive	Parameter	Value
(27) DTI2_CONNECT_RES		
<A>	link_id	LINK_ID_47

```

    <B>          link_id          LINK_ID_66
               version          DTI_VERSION_10
(28) DTI2_READY_IND
    <A>          link_id          LINK_ID_47
    <B>          link_id          LINK_ID_66
(29) PKT_DTI_OPEN_CNF
    <A>          device_no       DEVICE_47
    <B>          device_no       DEVICE_66
               cause           PKTCS_SUCCESS
    
```

History: 15-Oct-02 HM Initial

4.2.10 PKTIO022: Initial flow control primitive from peer received

Description: In state DTX_NOT_READY immediately after opening PKTIO receives the first flow control primitive. DTX state changes to DTX_READY.

<A> Device_47
 Device_66

Variants: <A>....

Preamble: <A> PKTIO021A
 PKTIO021B

```

          ACI                      PKTIO                      DIO
          |                        |                        |
(1)      |      DTI2_GETDATA_REQ  |                        |
          |                        |                        |
          *----->*                |                        |
MUTE (1000) |                        |                        |
          |                        |                        |
    
```

Parametrization

Primitive	Parameter	Value
(30) DTI2_GETDATA_REQ	link_id	LINK_ID_47
	link_id	LINK_ID_66

History: 15-Oct-02 HM Initial

4.2.11 PKTIO023: Opening of DTI connection rejected

Description: In state KER_DTI_OPENING the DTI2 answers negative, the connection is not to be opened. ACI is informed and the next state is KER_NO_DTI. The RX and TX states are not affected by this.

<A> Device_47
 Device_66

Variants: <A>....

Preamble: <A> PKTIO020A
 PKTIO020B

```

          ACI                      PKTIO                      DIO
          |                        |                        |
(1)      |      DTI2_DISCONNECT_REQ  |                        |
          |                        |                        |
          *----->*                |                        |
    
```

```
(2) |          PKT_DTI_CLOSE_IND          |          |
    | *<----->                         *          |
MUTE (1000) |          |          |
    |          |          |          |
```

Parametrization

Primitive	Parameter	Value
(31) DTI2_DISCONNECT_REQ		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
cause	DTI_CAUSE_NORMAL_CLOSE	
(32) PKT_DTI_CLOSE_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66

History: 15-Oct-02 HM Initial

4.2.12 PKTIO024: DTI disconnection in state KER_READY

Description: In state KER_READY the disconnection of the DTI connection is requested.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO022A
 PKTIO022B

```
          ACI          PKTIO          DIO
(1) |          DTI2_DISCONNECT_REQ          |          |
    | *=====>                         *          |
(2) |          PKT_DTI_CLOSE_IND          |          |
    | *<----->                         *          |
MUTE (1000) |          |          |
    |          |          |          |
```

Parametrization

Primitive	Parameter	Value
(33) DTI2_DISCONNECT_REQ		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
cause	DTI_CAUSE_NORMAL_CLOSE	
(34) PKT_DTI_CLOSE_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66

History: 15-Oct-02 HM Initial

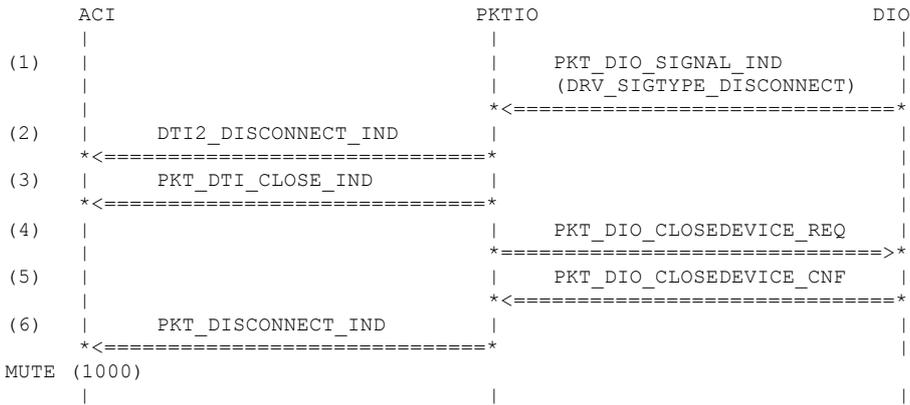
4.2.13 PKTIO025:KER_DTI_READY, DRV_SIGTYPE_DISCONNECT received

Description: In state KER_DTI_READY the driver sends a DRV_SIGTYPE_DISCONNECT. All PKTIO state machines are expected to be in the NULL state after the state transitions are finished.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO022A
 PKTIO022B



Parametrization

Primitive	Parameter	Value
(35) PKT_DIO_SIGNAL_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
	signal_type	DRV_SIGTYPE_DISCONNECT
(36) DTI2_DISCONNECT_IND		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
	cause	DTI_CAUSE_NORMAL_CLOSE
(37) PKT_DTI_CLOSE_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(38) PKT_DIO_CLOSEDEVICE_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(39) PKT_DIO_CLOSEDEVICE_CNF		
retval	DRV_OK	
(40) PKT_DISCONNECT_IND		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
	cause	PKTCS_DISCONNECT

History: 15-Oct-02 HM Initial
 06-Nov-02 HK DTI close before driver close.

4.2.14 PKTIO026: KER_DTI_OPENING, DRV_SIGTYPE_DISCONNECT received

Description: In state KER_DTI_OPENING the driver sends a DRV_SIGTYPE_DISCONNECT. All PKTIO state machines are expected to be in the NULL state after the state transitions are finished.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO020A
 PKTIO020B

```

          ACI                                PKTIO                                DIO
(1)      |                                  |          PKT_DIO_SIGNAL_IND          |
          |                                  |          (DRV_SIGTYPE_DISCONNECT) |
          |                                  |          *<=====*>          |
(2)      |          DTI2_DISCONNECT_IND      |                                  |
          |          *<=====*>              |                                  |
(3)      |          PKT_DTI_CLOSE_IND        |                                  |
          |          *<=====*>              |                                  |
(4)      |                                  |          PKT_DIO_CLOSEDEVICE_REQ    |
          |                                  |          *=====*>          |
(5)      |                                  |          PKT_DIO_CLOSEDEVICE_CNF    |
          |                                  |          *<=====*>          |
(6)      |          PKT_DISCONNECT_IND       |                                  |
          |          *<=====*>              |                                  |
MUTE (1000) |                                  |                                  |
          |                                  |                                  |
    
```

Parametrization

Primitive	Parameter	Value	
(41) PKT_DIO_SIGNAL_IND	<A>	device_no	DEVICE_47
		device_no	DEVICE_66
	signal_type		DRV_SIGTYPE_DISCONNECT
(42) DTI2_DISCONNECT_IND	<A>	link_id	LINK_ID_47
		link_id	LINK_ID_66
	cause		DTI_CAUSE_NORMAL_CLOSE
(43) PKT_DTI_CLOSE_IND	<A>	device_no	DEVICE_47
		device_no	DEVICE_66
(44) PKT_DIO_CLOSEDEVICE_REQ	<A>	device_no	DEVICE_47
		device_no	DEVICE_66

```
(45) PKT_DIO_CLOSEDEVICE_CNF
      retval          DRV_OK

(46) PKT_DISCONNECT_IND
      <A>              device_no      DEVICE_47
      <B>              device_no      DEVICE_66
      cause           PKTCS_DISCONNECT
```

History: 15-Oct-02 HM Initial
 06-Nov-02 HK DTI close before driver close.

4.2.15 PKTIO027: More than one read buffer supported by DIO driver

Description: Instead of the case which is tested in PKTIO011 - PKTIO022, here more than one read buffer is supported.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO010A
 PKTIO010B

```

      ACI                      PKTIO                      DIO
(1)  |                          |                          |
      |      PKT_CONNECT_RES    |                          |
      | *=====>*              |                          |
(2)  |                          |      PKT_DIO_SETCONFIG_REQ |
      |                          | *=====>*              |
(3)  |                          |      PKT_DIO_SETCONFIG_CNF |
      |                          | *<=====*              |
(4)  |                          |      PKT_DIO_READ_REQ      |
      |                          | *=====>*              |
(5)  |                          |      PKT_DIO_READ_CNF      |
      |                          | *<=====*              |
(6)  |                          |      PKT_DIO_READ_REQ      |
      |                          | *=====>*              |
(7)  |                          |      PKT_DIO_READ_CNF      |
      |                          | *<=====*              |
MUTE (1000)
(8)  |      PKT_DTI_OPEN_REQ    |                          |
      | *=====>*              |                          |
(9)  |      DTI2_CONNECT_IND    |                          |
      | *<=====*              |                          |
MUTE (1000)
(10) |      DTI2_CONNECT_RES    |                          |
      | *=====>*              |                          |
(11) |      DTI2_READY_IND     |                          |
      | *<=====*              |                          |
(12) |      PKT_DTI_OPEN_CNF   |                          |
      | *<=====*              |                          |
MUTE (1000)
(13) |      DTI2_GETDATA_REQ   |                          |
      | *=====>*              |                          |
MUTE (1000)
      |                          |                          |
```

Parametrization

Primitive	Parameter	Value
(47) PKT_CONNECT_RES		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
dio_dcb		DEFAULT_DCB_FROM_ACI
(48) PKT_DIO_SETCONFIG_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
dio_dcb		DEFAULT_DCB_FROM_ACI
(49) PKT_DIO_SETCONFIG_CNF		
retval		DRV_OK
(50) PKT_DIO_READ_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(51) PKT_DIO_READ_CNF		
retval		DRV_OK
(52) PKT_DIO_READ_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(53) PKT_DIO_READ_CNF		
retval		DRV_OK
(54) PKT_DTI_OPEN_REQ		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
peer		PEER_SNDP
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
dti_direction		DTI_DIRECTION_NORMAL
(55) DTI2_CONNECT_IND		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
version		DTI_VERSION_10
(56) DTI2_CONNECT_RES		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
version		DTI_VERSION_10
(57) DTI2_READY_IND		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
(58) PKT_DTI_OPEN_CNF		
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
cause		PKTCS_SUCCESS
(59) DTI2_GETDATA_REQ		
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66

History: 15-Oct-02 HM Initial
 29-Oct-02 HM Revised

4.2.16 PKTIO028: PKT_DTI_CLOSE_REQ received

Description: In state KER_DTI_OPENING the PKTIO entity receives a PKT_DTI_CLOSE_REQ primitive from ACI.

- <A> DTI2 disconnection requested in state KER_DTI_OPENING
- DTI2 disconnection requested in state KER_DTI_READY
- <C> DTI2 disconnection requested in state KER_DTI_READY, flow control received

Variants: <A>...<C>

Preamble: <A> PKTIO020A
 PKTIO021A
 <C> PKTIO022A

```

ACI                                     PKTIO                                     DIO
|                                     |                                     |
(1) |   PKT_DTI_CLOSE_REQ             |                                     |
    | *----->*                       |                                     |
(2) |   DTI2_DISCONNECT_IND          |                                     |
    | *-----<*                       |                                     |
(3) |   PKT_DTI_CLOSE_CNF            |                                     |
    | *-----<*                       |                                     |
MUTE (1000)                            |                                     |
|                                     |                                     |
    
```

Parametrization

Primitive	Parameter	Value
(60) PKT_DTI_CLOSE_REQ device_no	DEVICE_47	
(61) DTI2_DISCONNECT_IND link_id cause	LINK_ID_47 DTI_CAUSE_NORMAL_CLOSE	
(62) PKT_DTI_CLOSE_CNF device_no	DEVICE_47	

History: 15-Oct-02 HM Initial

4.3 Modifications

4.3.1 PKTIO050: KER_READY, modification of driver settings immediately possible

Description: In state KER_READY / TX_READY the driver settings are to be modified. The driver is flushed by calling DIO_Flush(), the function returns with DRV_OK as there is nothing to flush so that the modification immediately can take place. The state KER_MODIFY needs not to be entered, the next state is KER_READY again.

- <A> Successful case
- Parameters from ACI were trash, driver refused.

Variants: <A>...

Preamble: PKTIO022A

```

ACI                                     PKTIO                                     DIO
|                                     |                                     |
(1) |   PKT_MODIFY_REQ                |                                     |
    | *----->*                       |                                     |
(2) |                                     |   PKT_DIO_FLUSH_REQ             |
    
```

```

(3) | | | *=====>*
    | | | | PKT_DIO_FLUSH_CNF |
    | | | *<=====*
```

```

(4) | | | | PKT_DIO_GETBUFFER_REQ |
    | | | *=====>*
(5) | | | | PKT_DIO_GETBUFFER_CNF |
    | | | *<=====*
```

```

(6) | | | | PKT_DIO_SETCONFIG_REQ |
    | | | *=====>*
(7) | | | | PKT_DIO_SETCONFIG_CNF |
    | | | *<=====*
```

```

(8) | | | | PKT_MODIFY_CNF |
    | | | *<=====*
```

MUTE (1000)

Parametrization

Primitive	Parameter	Value
(63) PKT_MODIFY_REQ	DEVICE_47	
device_no	dio_dcb	MODIFIED_DCB_FROM_ACI
<A>	dio_dcb	TRASH_DCB_FROM_ACI
		
(64) PKT_DIO_FLUSH_REQ	DEVICE_47	
device_no		
(65) PKT_DIO_FLUSH_CNF	DRV_OK	
retval		
(66) PKT_DIO_GETBUFFER_REQ	DEVICE_47	
device_no		
(67) PKT_DIO_GETBUFFER_CNF	DRV_OK	
retval		
(68) PKT_DIO_SETCONFIG_REQ	DEVICE_47	
device_no	dio_dcb	MODIFIED_DCB_FROM_ACI
<A>	dio_dcb	TRASH_DCB_FROM_ACI
		
(69) PKT_DIO_SETCONFIG_CNF	DRV_OK	
<A>	retval	DRV_INVALID_PARAMS
	retval	
(70) PKT_MODIFY_CNF	DEVICE_47	
device_no	cause	PKTCS_SUCCESS
<A>	cause	PKTCS_INVALID_PARAMS
		

History: 15-Oct-02 HM Initial

4.3.2 PKTIO051: KER_READY, Modification of driver settings requested

Description: The driver settings are to be modified. The driver is flushed by calling DIO_Flush(), the function returns with DRV_INPROCESS as flushing needs time. The state KER_MODIFY / TX_FLUSHING is entered for cases <A> and , for cases <C> and <D> KER_MODIFY / TX_FLUSH_BUFFER is entered.
 <A> Parameters from ACI are good, initial TX_READY


```

MUTE (500)
(4) | | PKT_DIO_SIGNAL_IND | |
    | | (DRV_SIGTYPE_FLUSH) | |
    | | *<===== | |
(5) | | PKT_DIO_GETBUFFER_REQ | |
    | | *=====> | |
(6) | | PKT_DIO_GETBUFFER_CNF | |
    | | *<===== | |
(7) | | PKT_DIO_SETCONFIG_REQ | |
    | | *=====> | |
(8) | | PKT_DIO_SETCONFIG_CNF | |
    | | *<===== | |
(9) | | PKT_MODIFY_CNF | |
    | | *<===== | |
MUTE (1000)
    | | | |
    
```

Parametrization

Primitive	Parameter	Value
(74) PKT_DIO_SIGNAL_IND device_no signal_type	DEVICE_47 DRV_SIGTYPE_WRITE	
(75) PKT_DIO_GETBUFFER_REQ device_no	DEVICE_47	
(76) PKT_DIO_GETBUFFER_CNF retval	DRV_OK	
(77) PKT_DIO_SIGNAL_IND device_no signal_type	DEVICE_47 DRV_SIGTYPE_FLUSH	
(78) PKT_DIO_GETBUFFER_REQ device_no	DEVICE_47	
(79) PKT_DIO_GETBUFFER_CNF retval	DRV_OK	
(80) PKT_DIO_SETCONFIG_REQ device_no	DEVICE_47	
<A>	dio_dcb	MODIFIED_DCB_FROM_ACI
	dio_dcb	TRASH_DCB_FROM_ACI
(81) PKT_DIO_SETCONFIG_CNF <A> 	retval retval	DRV_OK DRV_INVALID_PARAMS
(82) PKT_MODIFY_CNF device_no	DEVICE_47	
<A>	cause	PKTCS_SUCCESS
	cause	PKTCS_INVALID_PARAMS

History: 15-Oct-02 HM Initial

4.3.4 PKTIO053: KER_MODIFY, DTI2_DISCONNECT_REQ

Description: In state KER_MODIFY / TX_FLUSHING the DTI connection is closed. This is indicated to ACI and the main state is changed to KER_MODIFY_NO_DTI, the DRX state changes from DRX_READY to DRX_DEAD_READY and the DTX state changes from DTX_READY to DTX_DEAD.

Note: The state transitions are not tested here. Only the emission of primitives is tested in this test case.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>...

Preamble: <A> PKTIO051A
 PKTIO051B

```

ACI                                     PKTIO                                     DIO
|                                       |                                       |
(1) | DTI2_DISCONNECT_REQ               |                                       |
    | *-----> *                         |                                       |
(2) | PKT_DTI_CLOSE_IND                 |                                       |
    | *-----> *                         |                                       |
MUTE (1000)                             |                                       |
|                                       |                                       |
    
```

Parametrization

Primitive	Parameter	Value
(83) DTI2_DISCONNECT_REQ	link_id	LINK_ID_47
	cause	DTI_CAUSE_NORMAL_CLOSE
(84) PKT_DTI_CLOSE_IND	device_no	DEVICE_47

History: 15-Oct-02 HM Initial

4.3.5 PKTIO054: KER_MODIFY_NO_DTI, DRV_SIGTYPE_FLUSH signalled

Description: In state KER_MODIFY_NO_DTI / TX_FLUSHING the driver signals the end of flushing. The driver's configuration is changed, ACI is informed about the result of the modification and the new main state KER_NO_DTI is entered.

Note: The state transitions are not tested here. Only the emission of primitives is tested in this test case.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>...

Preamble: <A> PKTIO053A
 PKTIO053B

```

ACI                                     PKTIO                                     DIO
|                                       |                                       |
(1) |                                       | PKT_DIO_SIGNAL_IND |
    |                                       | (DRV_SIGTYPE_WRITE)|
    | *-----> *                         | *-----> *         |
(2) |                                       | PKT_DIO_GETBUFFER_REQ |
    | *-----> *                         | *-----> *         |
(3) |                                       | PKT_DIO_GETBUFFER_CNF |
    | *-----> *                         | *-----> *         |
MUTE (500)                             |                                       |
    
```

```

(4) | | | PKT_DIO_SIGNAL_IND | |
    | | | (DRV_SIGTYPE_FLUSH) | |
    | | | *<===== | |
(5) | | | PKT_DIO_GETBUFFER_REQ | |
    | | | *=====> | |
(6) | | | PKT_DIO_GETBUFFER_CNF | |
    | | | *<===== | |
(7) | | | PKT_DIO_SETCONFIG_REQ | |
    | | | *=====> | |
(8) | | | PKT_DIO_SETCONFIG_CNF | |
    | | | *<===== | |
(9) | | | PKT_MODIFY_CNF | |
    | | | *<===== | |
MUTE (1000) | | | | |
    
```

Parametrization

Primitive	Parameter	Value
(85) PKT_DIO_SIGNAL_IND device_no signal_type	DEVICE_47 DRV_SIGTYPE_WRITE	
(86) PKT_DIO_GETBUFFER_REQ device_no	DEVICE_47	
(87) PKT_DIO_GETBUFFER_CNF retval	DRV_OK	
(88) PKT_DIO_SIGNAL_IND device_no signal_type	DEVICE_47 DRV_SIGTYPE_FLUSH	
(89) PKT_DIO_GETBUFFER_REQ device_no	DEVICE_47	
(90) PKT_DIO_GETBUFFER_CNF retval	DRV_OK	
(91) PKT_DIO_SETCONFIG_REQ device_no <A> 	DEVICE_47 dio_dcb dio_dcb	MODIFIED_DCB_FROM_ACI TRASH_DCB_FROM_ACI
(92) PKT_DIO_SETCONFIG_CNF <A> 	retval retval	DRV_OK DRV_INVALID_PARAMS
(93) PKT_MODIFY_CNF device_no <A> 	DEVICE_47 cause cause	PKTCS_SUCCESS PKTCS_INVALID_PARAMS

History: 15-Oct-02 HM Initial

4.3.6 PKTIO055: KER_MODIFY, DRV_SIGTYPE_DISCONNECT signalled

Description: In state KER_MODIFY the driver sends a DRV_SIGTYPE_DISCONNECT. All PKTIO state machines are expected to be in the NULL state after the state transitions are fin-

ished.
 <A> The driver disconnect is received in state TX_FLUSHING
 The driver disconnect is received in state TX_FLUSH_BUFFER

Variants:

<A>....

Preamble:

<A> PKTIO051A
 PKTIO051C

	ACI	PKTIO	DIO
(1)			
(2)			
(3)			
(4)			
(5)			
(6)			
(7)			
MUTE (1000)			

Parametrization

	Primitive	Parameter	Value
(94)	PKT_DIO_SIGNAL_IND		
	device_no	DEVICE_47	
	signal_type	DRV_SIGTYPE_DISCONNECT	
(95)	DTI2_DISCONNECT_IND		
	link_id	LINK_ID_47	
	cause	DTI_CAUSE_NORMAL_CLOSE	
(96)	PKT_DTI_CLOSE_IND		
	device_no	DEVICE_47	
(97)	PKT_MODIFY_CNF		
	device_no	DEVICE_47	
	cause	PKTCS_DISCONNECT	
(98)	PKT_DIO_CLOSEDEVICE_REQ		
	device_no	DEVICE_47	
(99)	PKT_DIO_CLOSEDEVICE_CNF		
	retval	DRV_OK	
(100)		PKT_DISCONNECT_IND	
	device_no	DEVICE_47	
	cause	PKTCS_DISCONNECT	

History: 15-Oct-02 HM Initial
 06-Nov-02 HK DTI close before driver close.

4.3.7 PKTIO056: KER_MODIFY_NO_DTI and DRV_SIGTYPE_DISCONNECT

Description: In state KER_MODIFY_NO_DTI the driver sends a DRV_SIGTYPE_DISCONNECT. All PKTIO state machines are expected to be in the NULL state after the state transitions are finished.

Preamble: PKTIO053A

	ACI	PKTIO	DIO
(1)			
		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_DISCONNECT)	
		<-----	
(2)	PKT_MODIFY_CNF		
	<-----		
(3)		PKT_DIO_CLOSEDEVICE_REQ	
		----->	
(4)		PKT_DIO_CLOSEDEVICE_CNF	
		<-----	
(5)	PKT_DISCONNECT_IND		
	<-----		
MUTE (1000)			

Parametrization

Primitive	Parameter	Value
(101)	device_no	PKT_DIO_SIGNAL_IND
	signal_type	DEVICE_47
		DRV_SIGTYPE_DISCONNECT
(102)	device_no	PKT_MODIFY_CNF
	cause	DEVICE_47
		PKTCS_DISCONNECT
(103)	device_no	PKT_DIO_CLOSEDEVICE_REQ
		DEVICE_47
(104)	retval	PKT_DIO_CLOSEDEVICE_CNF
		DRV_OK
(105)	device_no	PKT_DISCONNECT_IND
	cause	DEVICE_47
		PKTCS_DISCONNECT

History: 15-Oct-02 HM Initial

4.3.8 PKTIO057: KER_MODIFY / TX_FLUSH_BUFFER, end of flushing

Description: In state KER_MODIFY / TX_FLUSH_BUFFER the driver signals end of flushing activity by sending DRV_SIGTYPE_FLUSH. As there is still one descriptor stored in the PKTIO entity which has not been send, this is now submmitted to the DIO driver and flushing is performed again for this descriptor. Before end of flush in is assumed the driver indicates the successfully written packet by sending DRV_SIGTYPE_WRITE. This shall have no influence on the behaviour of PKTIO.

<A> Parameters from ACI were good, initial TX_FLUSH_BUFFER

 Parameters from ACI are trash, driver will refuse, initial TX_FLUSH_BUFFER

Variants: <A>...

Preamble: <A> PKTIO051C

 PKTIO051D

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_WRITE)	
		*<=====	
(2)		PKT_DIO_GETBUFFER_REQ	
		*=====>	
(3)		PKT_DIO_GETBUFFER_CNF	
		*<=====	
MUTE (500)			
(4)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_FLUSH)	
		*<=====	
(5)		PKT_DIO_GETBUFFER_REQ	
		*=====>	
(6)		PKT_DIO_GETBUFFER_CNF	
		*<=====	
(7)		PKT_DIO_WRITE_REQ	
		*=====>	
(8)		PKT_DIO_WRITE_CNF	
		*<=====	
(9)		PKT_DIO_FLUSH_REQ	
		*=====>	
(10)		PKT_DIO_FLUSH_CNF	
		*<=====	
MUTE (1000)			
(11)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_WRITE)	
		*<=====	
(12)		PKT_DIO_GETBUFFER_REQ	
		*=====>	
(13)		PKT_DIO_GETBUFFER_CNF	
		*<=====	
MUTE (1000)			

Parametrization

Primitive	Parameter	Value
(106)	device_no	PKT_DIO_SIGNAL_IND
	signal_type	DEVICE_47
		DRV_SIGTYPE_WRITE
(107)	device_no	PKT_DIO_GETBUFFER_REQ
		DEVICE_47
(108)	retval	PKT_DIO_GETBUFFER_CNF
		DRV_OK
(109)	device_no	PKT_DIO_SIGNAL_IND
	signal_type	DEVICE_47
		DRV_SIGTYPE_FLUSH
(110)	device_no	PKT_DIO_GETBUFFER_REQ
		DEVICE_47
(111)	retval	PKT_DIO_GETBUFFER_CNF
		DRV_OK
(7) (112)	device_no	PKT_DIO_WRITE_REQ
		DEVICE_47

Formatted: Bullets and Numbering

	state	DIO_STATE_IP
	mask	DIO_MASK_NONE
	sdu	A_DL_SDU
(113)	retval	PKT_DIO_WRITE_CNF DRV_OK
(114)	device_no	PKT_DIO_FLUSH_REQ DEVICE_47
(115)	retval	PKT_DIO_FLUSH_CNF DRV_INPROCESS
(116)	device_no signal_type	PKT_DIO_SIGNAL_IND DEVICE_47 DRV_SIGTYPE_WRITE
(117)	device_no	PKT_DIO_GETBUFFER_REQ DEVICE_47
(118)	retval	PKT_DIO_GETBUFFER_CNF DRV_OK

History: 15-Oct-02 HM Initial

4.3.9 PKTIO058: KER_MODIFY / TX_FLUSHING, end of flushing

Description: In state KER_MODIFY / TX_FLUSHING the driver signals end of flushing activity by sending DRV_SIGTYPE_FLUSH.

<A> Parameters from ACI were good, initial TX_FLUSHING

 Parameters from ACI are trash, driver will refuse, initial TX_FLUSHING

Variants: <A>....

Preamble: <A> PKTIO057A
 PKTIO057B

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_WRITE)	
		*<=====	
(2)		PKT_DIO_GETBUFFER_REQ	
		*=====>	
(3)		PKT_DIO_GETBUFFER_CNF	
		*<=====	
(4)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_FLUSH)	
		*<=====	
(5)		PKT_DIO_GETBUFFER_REQ	
		*=====>	
(6)		PKT_DIO_GETBUFFER_CNF	
		*<=====	
(7)		PKT_DIO_SETCONFIG_REQ	
		*=====>	
(8)		PKT_DIO_SETCONFIG_CNF	
		*<=====	
(9)	PKT_MODIFY_CNF		
	*<=====		
MUTE (1000)			

Parametrization

Primitive	Parameter	Value
(119)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_WRITE	
(120)	PKT_DIO_GETBUFFER_REQ	
device_no	DEVICE_47	
(121)	PKT_DIO_GETBUFFER_CNF	
retval	DRV_OK	
(122)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_FLUSH	
(123)	PKT_DIO_GETBUFFER_REQ	
device_no	DEVICE_47	
(124)	PKT_DIO_GETBUFFER_CNF	
retval	DRV_OK	
(125)	PKT_DIO_SETCONFIG_REQ	
device_no	DEVICE_47	
<A>	dio_dcb	MODIFIED_DCB_FROM_ACI
	dio_dcb	TRASH_DCB_FROM_ACI
(126)	PKT_DIO_SETCONFIG_CNF	
<A>	retval	DRV_OK
	retval	DRV_INVALID_PARAMS
(127)	PKT_MODIFY_CNF	
device_no	DEVICE_47	
<A>	cause	PKTCS_SUCCESS
	cause	PKTCS_INVALID_PARAMS

History: 15-Oct-02 HM Initial

4.3.10 PKTIO059: TX_FLUSHING or TX_FLUSH_BUFFER, packet written

Description: In state KER_MODIFY / TX_FLUSHING or TX_FLUSH_BUFFER the driver signals that a packet has been written. This shall not cause a state change of any states of the PKTIO entity, but the entity gets back the written descriptors at the soonest opportunity which is possible.

<A> TX_FLUSHING => TX_FLUSHING

 TX_FLUSH_BUFFER => TX_FLUSH_BUFFER

Variants: <A>...

Preamble: <A> PKTIO051A
 PKTIO051C

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_WRITE)	
		<=====	
(2)		PKT_DIO_GETBUFFER_REQ	
		<=====	

```
(3) | | | PKT_DIO_GETBUFFER_CNF |
MUTE (1000) | | | *<=====*
```

Primitive	Parameter	Value
(128)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_WRITE	
(129)	PKT_DIO_GETBUFFER_REQ	
device_no	DEVICE_47	
(130)	PKT_DIO_GETBUFFER_CNF	
retval	DRV_OK	

History: 15-Oct-02 HM Initial

4.3.11 PKTIO060: KER_DTI_OPENING, PKT_MODIFY_REQ immediate

Description: In state KER_DTI_OPENING the PKTIO entity receives a PKT_MODIFY_REQ. The modification can be immediately done as DIO_Flush returns with DRV_OK.

Note: This is almost the same as PKTIO050, but the state is different.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>....
Preamble: PKTIO020A

```
ACI | | | PKTIO | | | DIO
(1) | | | PKT_MODIFY_REQ | | | |
*=====*> | | | |
(2) | | | | | | | PKT_DIO_FLUSH_REQ |
| | | | | | | *=====*> |
(3) | | | | | | | PKT_DIO_FLUSH_CNF |
| | | | | | | *<=====* |
(4) | | | | | | | PKT_DIO_GETBUFFER_REQ |
| | | | | | | *=====*> |
(5) | | | | | | | PKT_DIO_GETBUFFER_CNF |
| | | | | | | *<=====* |
(6) | | | | | | | PKT_DIO_SETCONFIG_REQ |
| | | | | | | *=====*> |
(7) | | | | | | | PKT_DIO_SETCONFIG_CNF |
| | | | | | | *<=====* |
(8) | | | | | | | PKT_MODIFY_CNF |
*=====*> | | | |
MUTE (1000) | | | |
```

Primitive	Parameter	Value
(131)	PKT_MODIFY_REQ	
device_no	DEVICE_47	

```

    <A>          dio_dcb          MODIFIED_DCB_FROM_ACI
    <B>          dio_dcb          TRASH_DCB_FROM_ACI
(132)          PKT_DIO_FLUSH_REQ
    device_no    DEVICE_47
(133)          PKT_DIO_FLUSH_CNF
    retval      DRV_OK
(134)          PKT_DIO_GETBUFFER_REQ
    device_no    DEVICE_47
(135)          PKT_DIO_GETBUFFER_CNF
    retval      DRV_OK
(136)          PKT_DIO_SETCONFIG_REQ
    device_no    DEVICE_47
    <A>          dio_dcb          MODIFIED_DCB_FROM_ACI
    <B>          dio_dcb          TRASH_DCB_FROM_ACI
(137)          PKT_DIO_SETCONFIG_CNF
    <A>          retval          DRV_OK
    <B>          retval          DRV_INVALID_PARAMS
(138)          PKT_MODIFY_CNF
    device_no    DEVICE_47
    <A>          cause          PKTCS_SUCCESS
    <B>          cause          PKTCS_INVALID_PARAMS
    
```

History: 15-Oct-02 HM Initial

4.3.12 PKTIO061: KER_DTI_OPENING, PKT_MODIFY_REQ after flush

Description: In state KER_DTI_OPENING the driver settings are to be modified. The driver is flushed by calling DIO_Flush(), the function returns with DRV_INPROCESS as flushing needs time. The state KER_MODIFY_DTI_OPENING is entered.

Note: As no data packets were sent by DTI, in real life the driver never will return with DRV_INPROCESS, but always with DRV_OK. On the other hand, this should not do any harm for the test purpose.

Note: This is almost the same as PKTIO051, but the state is different.

<A> Parameters from ACI are good

 Parameters from ACI are trash, driver will refuse

Variants: <A>....
Preamble: PKTIO020A

```

    ACI          PKTIO          DIO
    |            |              |
(1) |          PKT_MODIFY_REQ  |
    *----->|
(2) |            |          PKT_DIO_FLUSH_REQ |
    |            |          *----->|
(3) |            |          PKT_DIO_FLUSH_CNF |
    |            |          *<-----*
MUTE (1000)
    |            |              |
    
```

Parametrization

Primitive	Parameter	Value
-----------	-----------	-------

```

(139)          PKT_MODIFY_REQ
        device_no  DEVICE_47
        <A>        dio_dcb      MODIFIED_DCB_FROM_ACI
        <B>        dio_dcb      TRASH_DCB_FROM_ACI

(140)          PKT_DIO_FLUSH_REQ
        device_no  DEVICE_47

(141)          PKT_DIO_FLUSH_CNF
        retval     DRV_INPROCESS
    
```

History: 15-Oct-02 HM Initial

4.3.13 PKTIO062: KER_MODIFY_DTI_OPENING, DRV_SIGTYPE_FLUSH

Description: In state KER_MODIFY_DTI_OPENING / TX_FLUSHING the driver signals end of flushing activity by sending DRV_SIGTYPE_FLUSH. The requested modification is performed. The next state is regardless of the result of the modification KER_DTI_OPENING / TX_READY.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>...

Preamble: <A> PKTIO061A
 PKTIO061B

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_FLUSH)	
		<=====	
(2)		PKT_DIO_GETBUFFER_REQ	
		=====	
(3)		PKT_DIO_GETBUFFER_CNF	
		=====	
(4)		PKT_DIO_SETCONFIG_REQ	
		=====	
(5)		PKT_DIO_SETCONFIG_CNF	
		=====	
(6)	PKT_MODIFY_CNF		
	=====		
MUTE (1000)			

Parametrization

	Primitive	Parameter	Value
(142)	device_no	PKT_DIO_SIGNAL_IND	DEVICE_47
	signal_type	DRV_SIGTYPE_FLUSH	
(143)	device_no	PKT_DIO_GETBUFFER_REQ	DEVICE_47
(144)	retval	PKT_DIO_GETBUFFER_CNF	DRV_OK
(145)	device_no	PKT_DIO_SETCONFIG_REQ	DEVICE_47

```

    <A>          dio_dcb          MODIFIED_DCB_FROM_ACI
    <B>          dio_dcb          TRASH_DCB_FROM_ACI
(146)
    <A>          PKT_DIO_SETCONFIG_CNF
    <A>          retval          DRV_OK
    <B>          retval          DRV_INVALID_PARAMS
(147)
    device_no    PKT_MODIFY_CNF
    <A>          DEVICE_47
    <A>          cause          PKTCS_SUCCESS
    <B>          cause          PKTCS_INVALID_PARAMS
    
```

History: 15-Oct-02 HM Initial

4.3.14 PKTIO063: KER_MODIFY_DTI_OPENING, DTI2_DISCONNECT_REQ

Description: In state KER_MODIFY_DTI_OPENING / TX_FLUSHING the opening of the DTI connection is closed. This is indicated to ACI and the main state is changed to KER_MODIFY_NO_DTI.

Note: This is almost the same as PKTIO053, but the start state is different.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>....

Preamble: <A> PKTIO061A
 PKTIO061B

```

    ACI          PKTIO          DIO
    |            |              |
(1) |            DTI2_DISCONNECT_REQ |              |
    *----->*
(2) |            PKT_DTI_CLOSE_IND  |              |
    *-----*
MUTE (1000)
    |            |              |
    
```

Parametrization

Primitive	Parameter	Value
(148)	DTI2_DISCONNECT_REQ	
link_id	LINK_ID_47	
cause	DTI_CAUSE_NORMAL_CLOSE	
(149)	PKT_DTI_CLOSE_IND	
device_no	DEVICE_47	

History: 15-Oct-02 HM Initial

4.3.15 PKTIO064: KER_MODIFY_DTI_OPENING, PKT_DTI_OPEN_CNF

Description: In state KER_MODIFY_DTI_OPENING / TX_FLUSHING the opening of the DTI connection is confirmed. Also the first flow control primitive from the DTI partner is received. The next main state is KER_MODIFY.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>....

Preamble: <A> PKTIO061A
 PKTIO061B

	ACI	PKTIO	DIO
(1)	DTI2_CONNECT_RES		
	=====>		
(2)	DTI2_READY_IND		
	<=====		
(3)	PKT_DTI_OPEN_CNF		
	<=====		
MUTE (1000)			
(4)	DTI2_GETDATA_REQ		
	=====>		
MUTE (1000)			

Parametrization

	Primitive	Parameter	Value
(150)	link_id	DTI2_CONNECT_RES	LINK_ID_47
	version	DTI_VERSION_10	
(151)	link_id	DTI2_READY_IND	LINK_ID_47
(152)	device_no	PKT_DTI_OPEN_CNF	DEVICE_47
	cause	PKTCS_SUCCESS	
(153)	link_id	DTI2_GETDATA_REQ	LINK_ID_47

History: 15-Oct-02 HM Initial

4.3.16 PKTIO065: KER_MODIFY_DTI_OPENING, DRV_SIGTYPE_DISCONNECT

Description: In state KER_MODIFY_DTI_OPENING the driver signals a DRV_SIGTYPE_DISCONNECT. All PKTIO state machines are expected to be in the NULL state after the state transitions are finished.

Preamble: PKTIO061A

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_DISCONNECT)	
		<=====	
(2)	DTI2_DISCONNECT_IND		
	<=====		
(3)	PKT_DTI_CLOSE_IND		
	<=====		
(4)	PKT_MODIFY_CNF		
	<=====		
(5)		PKT_DIO_CLOSEDEVICE_REQ	
		=====>	
(6)		PKT_DIO_CLOSEDEVICE_CNF	

```

(7) | | | | *<=====
    | | PKT_DISCONNECT_IND | | |
    | *<=====
MUTE (1000) | | | |
    | | | | |
    
```

Parametrization

Primitive	Parameter	Value
(154)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_DISCONNECT	
(155)	DTI2_DISCONNECT_IND	
link_id	LINK_ID_47	
cause	DTI_CAUSE_NORMAL_CLOSE	
(156)	PKT_DTI_CLOSE_IND	
device_no	DEVICE_47	
(157)	PKT_MODIFY_CNF	
device_no	DEVICE_47	
cause	PKTCS_DISCONNECT	
(158)	PKT_DIO_CLOSEDEVICE_REQ	
device_no	DEVICE_47	
(159)	PKT_DIO_CLOSEDEVICE_CNF	
retval	DRV_OK	
(160)	PKT_DISCONNECT_IND	
device_no	DEVICE_47	
cause	PKTCS_DISCONNECT	

History: 15-Oct-02 HM Initial

4.3.17 PKTIO070: KER_NO_DTI, PKT_MODIFY_REQ immediate

Description: In state KER_NO_DTI the PKTIO entity receives a PKT_MODIFY_REQ. The modification can be immediately done as DIO_Flush returns with DRV_OK.

Note: This is almost the same as PKTIO050 / PKTIO060, but the state is different.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>....
Preamble: PKTIO011A

```

          ACI                               PKTIO                               DIO
(1) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | *=====> *                           | | | | | | | | | | | | | | | | | | | | | |
(2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | *=====> *                           | | | | | | | | | | | | | | | | | | | | | |
(3) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | *=====> *                           | | | | | | | | | | | | | | | | | | | | | |
(4) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | *=====> *                           | | | | | | | | | | | | | | | | | | | | | |
(5) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | *=====> *                           | | | | | | | | | | | | | | | | | | | | | |
(6) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
    | *=====> *                           | | | | | | | | | | | | | | | | | | | | | |
    
```

```

(7) | | | *=====>*
    | | | | PKT_DIO_SETCONFIG_CNF |
(8) | | | *<=====*
```

MUTE (1000)

Parametrization

Primitive	Parameter	Value
(161)	PKT_MODIFY_REQ	
device_no	DEVICE_47	
<A>	dio_dcb	MODIFIED_DCB_FROM_ACI
	dio_dcb	TRASH_DCB_FROM_ACI
(162)	PKT_DIO_FLUSH_REQ	
device_no	DEVICE_47	
(163)	PKT_DIO_FLUSH_CNF	
retval	DRV_OK	
(164)	PKT_DIO_GETBUFFER_REQ	
device_no	DEVICE_47	
(165)	PKT_DIO_GETBUFFER_CNF	
retval	DRV_OK	
(166)	PKT_DIO_SETCONFIG_REQ	
device_no	DEVICE_47	
<A>	dio_dcb	MODIFIED_DCB_FROM_ACI
	dio_dcb	TRASH_DCB_FROM_ACI
(167)	PKT_DIO_SETCONFIG_CNF	
<A>	retval	DRV_OK
	retval	DRV_INVALID_PARAMS
(168)	PKT_MODIFY_CNF	
device_no	DEVICE_47	
<A>	cause	PKTCS_SUCCESS
	cause	PKTCS_INVALID_PARAMS

History: 15-Oct-02 HM Initial

4.3.18 PKTIO071: KER_DTI_OPENING, PKT_MODIFY_REQ after flush

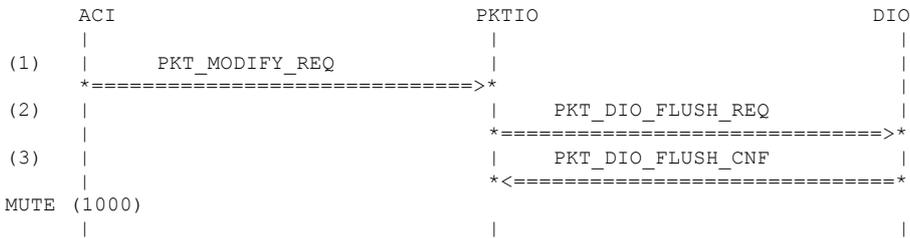
Description: In state KER_DTI_OPENING the driver settings are to be modified. The driver is flushed by calling DIO_Flush(), the function returns with DRV_INPROCESS as flushing needs time. The state KER_MODIFY_NO_DTI is entered.

Note: As no data packets were sent by DTI, in real life the driver never will return with DRV_INPROCESS, but always with DRV_OK. On the other hand, this should not do any harm for the test purpose.

Note: This is almost the same as PKTIO051 / PKTIO061, but the state is different.

<A> Parameters from ACI are good
 Parameters from ACI are trash, driver will refuse

Variants: <A>...
Preamble: PKTIO011A



Parametrization

Primitive	Parameter	Value
(169)	PKT_MODIFY_REQ	
device_no	DEVICE_47	
<A>	dio_dcb	MODIFIED_DCB_FROM_ACI
	dio_dcb	TRASH_DCB_FROM_ACI
(170)	PKT_DIO_FLUSH_REQ	
device_no	DEVICE_47	
(171)	PKT_DIO_FLUSH_CNF	
retval	DRV_INPROCESS	

History: 15-Oct-02 HM Initial

4.4 Connected phase - focus on RX / DTX

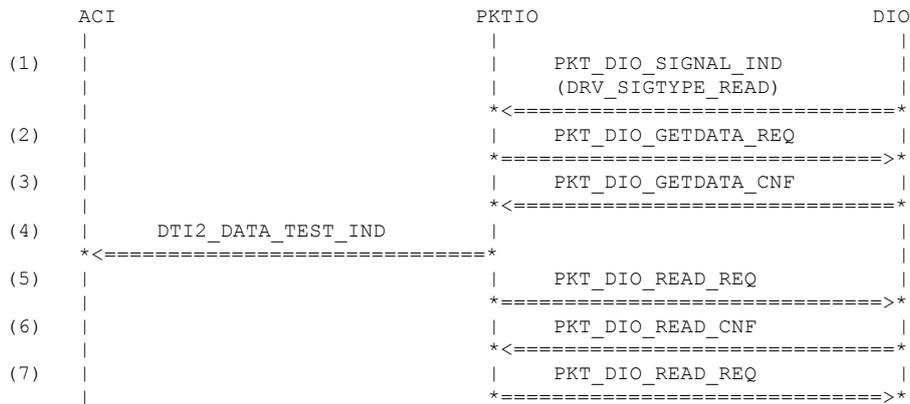
4.4.1 PKTIO100: KER_READY / DTX_READY, UL packet received

Description: In state KER_READY / DTX_READY the driver indicates the reception of a packet. As DTI is ready, the packet is fetched from the driver and transferred to the DTI peer. A new read buffer is provided to the driver.

<A> Device_47
 Device_66

Variants: <A>....

Preamble: <A> PKTIO022A
 PKTIO022B



```
(8) | | | PKT_DIO_READ_CNF |
    | | | *<===== |
MUTE (1000) | | | | |
    | | | | |
```

Parametrization

Primitive	Parameter	Value
(172)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_READ	
(173)	PKT_DIO_GETDATA_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(174)	PKT_DIO_GETDATA_CNF	
retval	DRV_OK	
state	DIO_SA	
sdu	A_UL_SDU	
(175)	DTI2_DATA_TEST_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_UL_SDU	
(176)	PKT_DIO_READ_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(177)	PKT_DIO_READ_CNF	
retval	DRV_OK	
(178)	PKT_DIO_READ_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(179)	PKT_DIO_READ_CNF	
retval	DRV_BUFFER_FULL	

History: 15-Oct-02 HM Initial

4.4.2 PKTIO101: Flow control primitive from DTI after DL data tranfer

Description: After having transferred a DTI packet, a flow control primitive is received. State becomes READY again.

<A> Device_47
 Device_66

Variants: <A>...

Preamble: <A> PKTIO100A
 PKTIO100B

```
ACI | | | PKTIO | | | DIO
    | | | | | | |
(1) | | DTI2_GETDATA_REQ | | |
    | | *===== | | |
    | | * | | |
```

MUTE (1000)

Parametrization

Primitive	Parameter	Value
(180)	DTI2_GETDATA_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66

History: 15-Oct-02 HM Initial

4.4.3 PKTIO102: No flow control primitive from DTI after DL data transfer

Description: After having transferred a DTI packet, no flow control primitive is received by the PKTIO entity. The entity receives a second DL data packet. Afterwards there is one filled descriptor stored by the PKTIO entity and one empty descriptor owned by the DIO driver.

<A> Device_47
 Device_66

Variants:

<A>....

Preamble:

<A> PKTIO100A
 PKTIO100B

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND (DRV_SIGTYPE_READ)	
		<=====	
(2)		PKT_DIO_GETDATA_REQ	
		=====	
(3)		PKT_DIO_GETDATA_CNF	
		<=====	
(4)		PKT_DIO_READ_REQ	
		=====	
(5)		PKT_DIO_READ_CNF	
		<=====	
MUTE (1000)			

Parametrization

Primitive	Parameter	Value
(181)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_READ	
(182)	PKT_DIO_GETDATA_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(183)	PKT_DIO_GETDATA_CNF	
retval	DRV_OK	
state	DIO_SA	
sdu	A_FIRST_UL_SDU	

```
(184)          PKT_DIO_READ_REQ
    <A>         device_no      DEVICE_47
    <B>         device_no      DEVICE_66

(185)          PKT_DIO_READ_CNF
    retval     DRV_OK
```

History: 15-Oct-02 HM Initial
 28-Oct-02 HM Revised

4.4.4 PKTIO103: No flow control primitive from DTI after DL data tranfer

Description: The initial state here is that DTX is not READY, the DIO driver has one read buffer and the PKTIO entity has one read buffer filled with data. While DTI has not become READY in the meantime a packet is received. It is checked that the PKTIO entity doesn't offer a read buffer to the DIO driver to ensure the number of parallel occupied descriptors in the system PKTIO / DIO does not exceed two. The final state is that the DTX process is not READY, there is no read buffer present in the driver and there are two read buffers filled with data present in the PKTIO entity.

<A> Device_47
 Device_66

Variants:

<A>....

Preamble:

<A> PKTIO102A
 PKTIO102B

```
          ACI          PKTIO          DIO
(1)      |             |             |
          |             |   PKT_DIO_SIGNAL_IND   |
          |             |   (DRV_SIGTYPE_READ)   |
          |             | *<=====*>          |
(2)      |             |   PKT_DIO_GETDATA_REQ   |
          |             | *=====*>          |
(3)      |             |   PKT_DIO_GETDATA_CNF   |
          |             | *<=====*>          |
MUTE (1000) |             |             |
          |             |             |
```

Parametrization

Primitive	Parameter	Value
(186)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_READ	
(187)	PKT_DIO_GETDATA_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(188)	PKT_DIO_GETDATA_CNF	
retval	DRV_OK	
state	DIO_SA	
sdu	A_SECOND_UL_SDU	

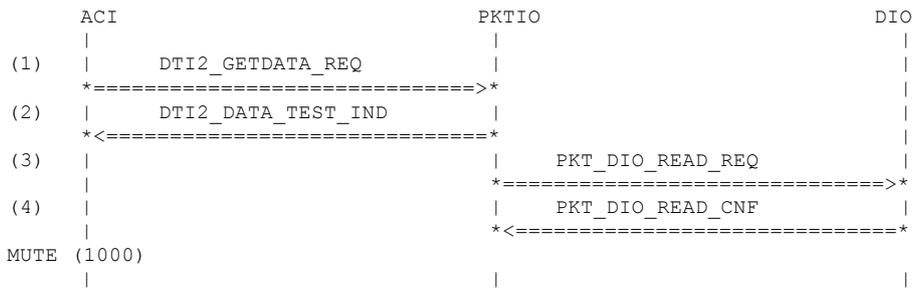
History: 15-Oct-02 HM Initial
 28-Oct-02 HM Revised

4.4.5 PKTIO104: DTI becomes ready while one descriptor present in PKTIO

Description: The initial state here is that DTX is not READY. The PKTIO entity has no free read buffer and the DIO driver has no filled read buffer. In case <A>, the PKTIO entity has one filled read buffer and the DIO driver has one free read buffer, in case the PKTIO entity has two filled read buffers and the DIO driver has no read buffer at all. The expected behaviour is that the PKTIO entity sends the filled read data to the DTI peer and offers exactly a new empty read buffer to the DIO driver.
 <A> Initial: PKTIO 1 filled read buffer, DIO 1 free read buffer
 <A> Final: PKTIO 1 free read buffer, DIO 1 free read buffer
 Initial: PKTIO 2 filled read buffers
 Final: PKTIO 1 filled read buffer, DIO 1 free read buffer
 <C>,...<D> same as before, but with DEVICE_66

Variants: <A>....<D>

Preamble:
 <A> PKTIO102A
 PKTIO103A
 <C> PKTIO102B
 <D> PKTIO103B



Parametrization

Primitive	Parameter	Value
(189)	DTI2_GETDATA_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_47
<C>	link_id	LINK_ID_66
<D>	link_id	LINK_ID_66
(190)	DTI2_DATA_TEST_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_47
<C>	link_id	LINK_ID_66
<D>	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_FIRST_UL_SDU	
(191)	PKT_DIO_READ_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_47
<C>	device_no	DEVICE_66
<D>	device_no	DEVICE_66


```

(4) | | | *=====>*
    | | | | PKT_DIO_READ_CNF |
MUTE (1000) | | | *<=====*
```

Parametrization

Primitive	Parameter	Value
(194)	DTI2_GETDATA_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
(195)	DTI2_DATA_TEST_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_SECOND_UL_SDU	
(196)	PKT_DIO_READ_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(197)	PKT_DIO_READ_CNF	
retval	DRV_BUFFER_FULL	

History: 15-Oct-02 HM Initial
 28-Oct-02 HM Revised

4.4.8 PKTIO108: RX_BUFFER, DRV_SIGTYPE_DISCONNECT from DIO

Description: The PKTIO entity has one or more filled read buffers from the application and therefore its RX process is in state RX_BUFFER. The driver signals that the device is to be closed.

- <A> The entity has one filled read buffer while the driver has one free read buffer
- The entity has two filled read buffers while the driver has no free read buffer
- <C> The entity has no filled read buffer, DTX_NOT_READY after data transfer

Variants:

Preamble: PKTIO102A
 PKTIO103A
 PKTIO100A

```

(1) | | | PKT_DIO_SIGNAL_IND |
    | | | | (DRV_SIGTYPE_DISCONNECT) |
    | | | *<=====*
```

```

(2) | DTI2_DISCONNECT_IND |
    | *<=====*
```

```

(3) | PKT_DTI_CLOSE_IND |
    | *<=====*
```

```

(4) | | | PKT_DIO_CLOSEDEVICE_REQ |
    | | | *=====>*
```

```

(5) | | | PKT_DIO_CLOSEDEVICE_CNF |
    | | | *<=====*
```

```

(4) | PKT_DISCONNECT_IND |
    | *<=====*
```

MUTE (1000) | | |

Parametrization

Primitive	Parameter	Value
(198)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_DISCONNECT	
(199)	DTI2_DISCONNECT_IND	
link_id	LINK_ID_47	
cause	DTI_CAUSE_NORMAL_CLOSE	
(200)	PKT_DTI_CLOSE_IND	
device_no	DEVICE_47	
(201)	PKT_DIO_CLOSEDEVICE_REQ	
device_no	DEVICE_47	
(202)	PKT_DIO_CLOSEDEVICE_CNF	
retval	DRV_OK	
(203)	PKT_DISCONNECT_IND	
device_no	DEVICE_47	
cause	PKTCS_DISCONNECT	

History: 15-Oct-02 HM Initial
 29-Oct-02 HM Revised

4.4.9 PKTIO109: KER_READY, RX_READY, DTX_READY, packet received

Description: We have a DIO driver which supports 2 (more than 2?) read buffers. The driver signals that a packet has been received. It is tested that exactly one new empty read buffer is handed over to the driver. The expected final state is KER_READY, RX_READY, DTX_NOT_READY.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO027A
 PKTIO027B

```

(1) | | | PKT_DIO_SIGNAL_IND | |
    | | | (DRV_SIGTYPE_READ) | |
    | | | *<===== |
(2) | | | PKT_DIO_GETDATA_REQ | |
    | | | *=====> |
(3) | | | PKT_DIO_GETDATA_CNF | |
    | | | *<===== |
(4) | | DTI2_DATA_TEST_IND | |
    | | *<===== |
(5) | | | PKT_DIO_READ_REQ | |
    | | | *=====> |
(6) | | | PKT_DIO_READ_CNF | |
    | | | *<===== |
MUTE (1000) | | | | |
    | | | | |
    
```

Parametrization

Primitive	Parameter	Value
-----------	-----------	-------

```

(204)          PKT_DIO_SIGNAL_IND
<A>          device_no      DEVICE_47
<B>          device_no      DEVICE_66
             signal_type    DRV_SIGTYPE_READ

(205)          PKT_DIO_GETDATA_REQ
<A>          device_no      DEVICE_47
<B>          device_no      DEVICE_66

(206)          PKT_DIO_GETDATA_CNF
             retval
             state
             sdu            DRV_OK
                        DIO_OK
                        A_UL_SDU

(207)          DTI2_DATA_TEST_IND
<A>          link_id        LINK_ID_47
<B>          link_id        LINK_ID_66
             parameters
             sdu            NOT_USED
                        A_UL_SDU

(208)          PKT_DIO_READ_REQ
<A>          device_no      DEVICE_47
<B>          device_no      DEVICE_66

(209)          PKT_DIO_READ_CNF
             retval          DRV_OK
    
```

History: 15-Oct-02 HM Initial

4.4.10 PKTIO110: KER_READY, RX_READY, DTX_NOT_READY, flow control

Description: We have a DIO driver which supports 2 read buffers. The PKTIO entity receives a flow control primitive from the peer. It is expected that no visible reaction to the outside is seen. The expected end state is KER_READY, RX_READY, DTX_READY.

Note: See comments for PKTIO101, PKTIO106. Could be unified.

<A> DEVICE_47
 DEVICE_66

Variants: <A>....

Preamble: <A> PKTIO109A
 PKTIO109B

```

(1) |          DTI2_GETDATA_REQ          |
    | *-----> *                      |
MUTE (1000)
    |          |                      |
    
```

Parametrization

Primitive	Parameter	Value
(210)	DTI2_GETDATA_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66

History: 15-Oct-02 HM Initial

4.4.11 PKTIO111: KER_READY, RX_READY, DTX_NOT_READY, packets received

Description: We have a DIO driver which supports 2 read buffers. The driver signals that a packet has been received. The DTI peer is not ready. The PKTIO entity fetches the read buffer and stores it, awaiting a DTI primitive from the peer. The foreseen final state is KER_READY, RX_BUFFER, DTX_NOT_READY, the PKTIO entity owning two filled read buffers and the DIO driver owning no empty read buffer.

<A> DEVICE_47
 DEVICE_66

Variants:

<A>...

Preamble:

<A> PKTIO109A
 PKTIO109B

```
(1) | | | PKT_DIO_SIGNAL_IND | |
    | | | (DRV_SIGTYPE_READ) | |
    | | | *<===== | |
(2) | | | PKT_DIO_GETDATA_REQ | |
    | | | *=====> | |
(3) | | | PKT_DIO_GETDATA_CNF | |
    | | | *<===== | |
MUTE (1000)
(4) | | | PKT_DIO_SIGNAL_IND | |
    | | | (DRV_SIGTYPE_READ) | |
    | | | *<===== | |
(5) | | | PKT_DIO_GETDATA_REQ | |
    | | | *=====> | |
(6) | | | PKT_DIO_GETDATA_CNF | |
    | | | *<===== | |
MUTE (1000)
    | | | | |
```

Parametrization

Primitive	Parameter	Value
(211)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_READ	
(212)	PKT_DIO_GETDATA_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(213)	PKT_DIO_GETDATA_CNF	
retval	DRV_OK	
state	DIO_SA	
sdu	A_FIRST_UL_SDU	
(214)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_READ	
(215)	PKT_DIO_GETDATA_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(216)	PKT_DIO_GETDATA_CNF	
retval	DRV_OK	

state DIO_SA
 sdu A_SECOND_UL_SDU

History: 15-Oct-02 HM Initial

4.4.12 PKTIO112: KER_READY, RX_BUFFER, DTX_NOT_READY, flow control

Description: We have a DIO driver which supports 2 read buffers. The PKTIO entity possesses two filled read buffers, the DIO driver possesses no read buffer. The DTI peer signals flow control, a stored packed is sent and the DIO driver provided with an empty read buffer. The intermediate state is KER_READY, RX_BUFFER, DTX_NOT_READY, still one filled read buffer stored in the PKTIO entity and one empty read buffer stored in the DIO driver. Subsequently we get the second flow control primitive, PKTIO sends the second stored read data and provides a second empty read buffer to the DIO driver. The foreseen second intermediate state is KER_READY, RX_READY, DTX_NOT_READY, the PKTIO entity possessing no read buffer and the DIO driver possessing two empty read buffers. After receiving a flow control primitive from the DTI peer, the final DTX state becomes DTX_READY without any observable reaction from the test machine.

<A> DEVICE_47
 DEVICE_66

Variants:

<A>...

Preamble:

<A> PKTIO111A
 PKTIO111B

```

(1) | DTI2_GETDATA_REQ | |
    | *=====| |
(2) | DTI2_DATA_TEST_IND | |
    | *<=====| |
(3) | | | PKT_DIO_READ_REQ |
    | | | *=====| |
(4) | | | PKT_DIO_READ_CNF |
    | | | *<=====| |
MUTE (1000)
(5) | DTI2_GETDATA_REQ | |
    | *=====| |
(6) | DTI2_DATA_TEST_IND | |
    | *<=====| |
(7) | | | PKT_DIO_READ_REQ |
    | | | *=====| |
(8) | | | PKT_DIO_READ_CNF |
    | | | *<=====| |
MUTE (1000)
(9) | DTI2_GETDATA_REQ | |
    | *=====| |
MUTE (1000)
    | | | |
    
```

Parametrization

Primitive	Parameter	Value
(217)	DTI2_GETDATA_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
(218)	DTI2_DATA_TEST_IND	
<A>	link_id	LINK_ID_47

```

<B>          link_id          LINK_ID_66
      parameters
      sdu      NOT_USED
              A_FIRST_UL_SDU

(219)        PKT_DIO_READ_REQ
<A>          device_no       DEVICE_47
<B>          device_no       DEVICE_66

(220)        PKT_DIO_READ_CNF
      retval  DRV_OK

(221)        DTI2_GETDATA_REQ
<A>          link_id         LINK_ID_47
<B>          link_id         LINK_ID_66

(222)        DTI2_DATA_TEST_IND
<A>          link_id         LINK_ID_47
<B>          link_id         LINK_ID_66
      parameters
      sdu      NOT_USED
              A_SECOND_UL_SDU

(223)        PKT_DIO_READ_REQ
<A>          device_no       DEVICE_47
<B>          device_no       DEVICE_66

(224)        PKT_DIO_READ_CNF
      retval  DRV_OK

(225)        DTI2_GETDATA_REQ
<A>          link_id         LINK_ID_47
<B>          link_id         LINK_ID_66
    
```

History: 15-Oct-02 HM Initial

4.4.13 PKTIO113: KER_READY, RX_BUFFER, DTX_NOT_READY, flow control

Description: This is almost the same as PKTIO112, with the difference that here after the DIO_Read() calls packets are received by the driver. We want to check if order of received packets is maintained.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO111A
 PKTIO111B

```

(1) | DTI2_GETDATA_REQ | |
    | *=====|
(2) | DTI2_DATA_TEST_IND | |
    | *<=====|
(3) | | | PKT_DIO_READ_REQ | |
    | | | *=====|
(4) | | | PKT_DIO_READ_CNF | |
    | | | *<=====|
MUTE (1000)
(5) | | | PKT_DIO_SIGNAL_IND | |
    | | | (DRV_SIGTYPE_READ) | |
    | | | *<=====|
(6) | | | PKT_DIO_GETDATA_REQ | |
    | | | *=====|
(7) | | | PKT_DIO_GETDATA_CNF | |
    | | | *<=====|
    
```

```

MUTE (1000)
(8) | DTI2_GETDATA_REQ | |
    | *=====>* | |
(9) | DTI2_DATA_TEST_IND | |
    | *<=====* | |
(10) | | | PKT_DIO_READ_REQ | |
    | | | *=====>* | |
(11) | | | PKT_DIO_READ_CNF | |
    | | | *<=====* | |
MUTE (1000)
(12) | DTI2_GETDATA_REQ | |
    | *=====>* | |
(13) | DTI2_DATA_TEST_IND | |
    | *<=====* | |
(14) | | | PKT_DIO_READ_REQ | |
    | | | *=====>* | |
(15) | | | PKT_DIO_READ_CNF | |
    | | | *<=====* | |
MUTE (1000)
(16) | DTI2_GETDATA_REQ | |
    | *=====>* | |
MUTE (1000)
    | | | |
    
```

Parametrization

Primitive	Parameter	Value
(226)	DTI2_GETDATA_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
(227)	DTI2_DATA_TEST_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_FIRST_UL_SDU	
(228)	PKT_DIO_READ_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(229)	PKT_DIO_READ_CNF	
retval	DRV_OK	
(230)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_READ	
(231)	PKT_DIO_GETDATA_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(232)	PKT_DIO_GETDATA_CNF	
retval	DRV_OK	
state	DIO_SA	
sdu	A_THIRD_UL_SDU	
(233)	DTI2_GETDATA_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66

```

(234)          DTI2_DATA_TEST_IND
<A>          link_id          LINK_ID_47
<B>          link_id          LINK_ID_66
parameters   NOT_USED
sdu          A_SECOND_UL_SDU

(235)          PKT_DIO_READ_REQ
<A>          device_no       DEVICE_47
<B>          device_no       DEVICE_66

(236)          PKT_DIO_READ_CNF
retval       DRV_OK

(237)          DTI2_GETDATA_REQ
<A>          link_id          LINK_ID_47
<B>          link_id          LINK_ID_66

(238)          DTI2_DATA_TEST_IND
<A>          link_id          LINK_ID_47
<B>          link_id          LINK_ID_66
parameters   NOT_USED
sdu          A_THIRD_UL_SDU

(239)          PKT_DIO_READ_REQ
<A>          device_no       DEVICE_47
<B>          device_no       DEVICE_66

(240)          PKT_DIO_READ_CNF
retval       DRV_OK

(241)          DTI2_GETDATA_REQ
<A>          link_id          LINK_ID_47
<B>          link_id          LINK_ID_66
    
```

History: 15-Oct-02 HM Initial

4.5 Connected phase - focus on RX / DTX, multi instance tests

4.6 Connected phase - focus on TX / DRX

4.6.1 PKTIO200: DRX_READY, DL packet received

Description: In state DRX_READY a packet from the DTI2 peer is received. It can be successfully written to the driver. As there is exactly one place in the PKTIO entity where a further DTI2_DATA_REQ could be buffered, the flow control primitive is sent immediately.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO022A
 PKTIO022B

```

          ACI                      PKTIO                      DIO
          |                        |                        |
(1)      | DTI2_DATA_TEST_REQ      |                        |
          | *=====>*            |                        |
(2)      |                        | PKT_DIO_WRITE_REQ      |
          |                        | *=====>*            |
(3)      |                        | PKT_DIO_WRITE_CNF      |
          |                        | *<=====*            |
    
```

```
(4) | DTI2_READY_IND |
    | *-----* |
MUTE (1000) |
    | |
```

Parametrization

Primitive	Parameter	Value
(242)	DTI2_DATA_TEST_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_DL_SDU	
(2) (243)	PKT_DIO_WRITE_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
state	DIO_STATE_IP	
mask	DIO_MASK_NONE	
sdu	A_DL_SDU	
(244)	PKT_DIO_WRITE_CNF	
retval	DRV_OK	
(245)	DTI2_READY_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66

Formatted: Bullets and Numbering

History: 15-Oct-02 HM Initial

4.6.2 PKTIO201: DRX_READY, DL packet received, DIO driver has one packet

Description: In state DRX_READY a packet from the DTI2 peer is received. It can not be successfully written to the driver and therefore has to be buffered in the PKTIO entity. The TX state is expected to become TX_BUFFER, the the DRX state is expected to become DRX_NOT_READY.

<A> The driver supports exactly one TX buffer, DEVICE_47
 The driver supports exactly one TX buffer, DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO200A
 PKTIO200B

```
ACI | PKTIO | DIO
(1) | DTI2_DATA_TEST_REQ | |
    | *-----* | |
(2) | | PKT_DIO_WRITE_REQ | |
    | | *-----* | |
(3) | | PKT_DIO_WRITE_CNF | |
    | | *-----* | |
MUTE (1000) | | |
```

Parametrization

Primitive	Parameter	Value
-----------	-----------	-------

```

(246)          DTI2_DATA_TEST_REQ
<A>          link_id          LINK_ID_47
<B>          link_id          LINK_ID_66
parameters   NOT_USED
sdu          A_DL_SDU

(247)          PKT_DIO_WRITE_REQ
<A>          device_no       DEVICE_47
<B>          device_no       DEVICE_66
state        DIO_STATE_IP
mask         DIO_MASK_NONE
sdu          A_DL_SDU

(248)          PKT_DIO_WRITE_CNF
retval       DRV_BUFFER_FULL
    
```

Formatted: Bullets and Numbering

History: 15-Oct-02 HM Initial

4.6.3 PKTIO202: DRX_READY, TX_READY, packet written by driver

Description: In state DRX_READY, TX_READY the PKTIO entity receives a signal from the DIO driver indicating that a packet could successfully be written by the driver. The PKTIO entity fetches the written descriptor out of control of the DIO driver. Now flow control for the DTI peer is necessary here as this already has been done, no DL packet is waiting in the PKTIO entity.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO200A
 PKTIO200B

	ACI		PKTIO		DIO
(1)					
				PKT_DIO_SIGNAL_IND	
				(DRV_SIGTYPE_WRITE)	
				<=====	
(2)				PKT_DIO_GETBUFFER_REQ	
				=====	
(3)				PKT_DIO_GETBUFFER_CNF	
				<=====	
MUTE (1000)					

Parametrization

Primitive	Parameter	Value
(249)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_WRITE	
(250)	PKT_DIO_GETBUFFER_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(251)	PKT_DIO_GETBUFFER_CNF	
retval	DRV_OK	

History: 15-Oct-02 HM Initial

4.6.4 PKTIO203: DRX_NOT_READY, TX_BUFFER, packet written by driver

Description: In state DRX_NOT_READY, TX_BUFFER the PKTIO entity receives a signal from the DIO driver indicating that a packet could successfully be written by the driver. As the PKTIO entity has stored a descriptor which is waiting to be sent, this is transferred to the driver. The PKTIO entity indicates by using a flow control primitive that it is ready to receive further data.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO201A
 PKTIO201B

	ACI		PKTIO		DIO
(1)				PKT_DIO_SIGNAL_IND	
				(DRV_SIGTYPE_WRITE)	
				<=====	
(2)				PKT_DIO_GETBUFFER_REQ	
				=====	
(3)				PKT_DIO_GETBUFFER_CNF	
				<=====	
(4)				PKT_DIO_WRITE_REQ	
				=====	
(5)				PKT_DIO_WRITE_CNF	
				<=====	
(6)		DTI2_READY_IND			
		<=====			
MUTE	(1000)				

Parametrization

Primitive	Parameter	Value
(252)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_WRITE	
(253)	PKT_DIO_GETBUFFER_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(254)	PKT_DIO_GETBUFFER_CNF	
retval	DRV_OK	
(4) (255)	PKT_DIO_WRITE_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
state	DIO_STATE_IP	
mask	DIO_MASK_NONE	
sdu	A_DL_SDU	
(256)	PKT_DIO_WRITE_CNF	
retval	DRV_OK	

Formatted: Bullets and Numbering

(257) DTI2_READY_IND
 <A> link_id LINK_ID_47
 link_id LINK_ID_66

History: 15-Oct-02 HM Initial

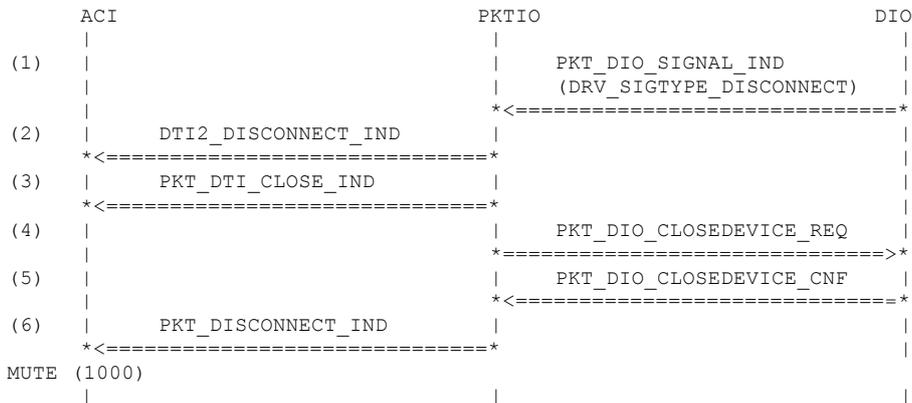
4.6.5 PKTIO204: DRX_NOT_READY, TX_BUFFER, disconnect by driver

Description: In state DRX_NOT_READY, TX_BUFFER the PKTIO entity receives a signal from the DIO driver indicating that the device is to be closed. The device is closed, the closing is indicated to the DTI peer, to the DTI manager within ACI and to ACI for the device itself.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO201A
 PKTIO201B



Parametrization

Primitive	Parameter	Value
(258)	PKT_DIO_SIGNAL_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
signal_type	DRV_SIGTYPE_DISCONNECT	
(259)	DTI2_DISCONNECT_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
cause	DTI_CAUSE_NORMAL_CLOSE	
(260)	PKT_DTI_CLOSE_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(261)	PKT_DIO_CLOSEDEVICE_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66

```
(262)          PKT_DIO_CLOSEDEVICE_CNF
              retval          DRV_OK

(263)          PKT_DISCONNECT_IND
              <A>          device_no          DEVICE_47
              <B>          device_no          DEVICE_66
              cause          PKTCS_DISCONNECT
```

History: 15-Oct-02 HM Initial

4.6.6 PKTIO205: DRX_READY, TX_FLUSHING, DL packet received

Description: In state DRX_READY, TX_FLUSHING the PKTIO entity receives a DL packet from the DTI peer. As the driver is in the process of flushing the PKTIO entity won't send the packet to the DIO driver but store it internally. The next states are DRX_NOT_READY, TX_FLUSH_BUFFER.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO201A
 PKTIO201B

```
          ACI          PKTIO          DIO
          |          |          |
(1)      |          DTI2_DATA_TEST_REQ          |          |
          |          *=====>*          |          |
MUTE (1000) |          |          |          |
```

Parametrization

Primitive	Parameter	Value
(264)	DTI2_DATA_TEST_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_DL_SDU	

History: 15-Oct-02 HM Initial

4.6.7 PKTIO206: DRX_NOT_READY, TX_BUFFER, disconnect by peer

Description: In state DRX_NOT_READY, TX_BUFFER the PKTIO entity receives a DTI2_DISCONNECT_REQ primitive from the DTI peer. The next state is DRX_DEAD_NOT_READY. The driver remains fully functional.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO201A
 PKTIO201B

```
          ACI          PKTIO          DIO
          |          |          |
(1)      |          DTI2_DISCONNECT_REQ          |          |
          |          *=====>*          |          |
```

```
(2) |          PKT_DTI_CLOSE_IND          |          |
    | *<=====*>                        |          |
MUTE (1000) |          |          |
    |          |          |          |
```

Parametrization

Primitive	Parameter	Value
(265)	DTI2_DISCONNECT_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
cause	DTI_CAUSE_NORMAL_CLOSE	
(266)	PKT_DTI_CLOSE_IND	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66

History: 15-Oct-02 HM Initial

4.6.8 PKTIO207: DRX_NOT_READY, TX_BUFFER, disconnect req. by ACI

Description: In state DRX_NOT_READY, TX_BUFFER the PKTIO entity receives a PKT_DTI_DISCONNECT_REQ primitive from ACI. The next state is DRX_DEAD_NOT_READY. The driver remains fully functional.

<A> DEVICE_47
 DEVICE_66

Variants: <A>...

Preamble: <A> PKTIO201A
 PKTIO201B

```
          ACI                                PKTIO                                DIO
(1) |          PKT_DTI_CLOSE_REQ          |          |
    | *=====*>                          |          |
(2) |          DTI2_DISCONNECT_IND        |          |
    | *<=====*>                          |          |
(2) |          PKT_DTI_CLOSE_CNF          |          |
    | *<=====*>                          |          |
MUTE (1000) |          |          |
    |          |          |          |
```

Parametrization

Primitive	Parameter	Value
(267)	PKT_DTI_CLOSE_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
(268)	DTI2_DISCONNECT_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
cause	DTI_CAUSE_NORMAL_CLOSE	
(269)	PKT_DTI_CLOSE_CNF	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66

History: 15-Oct-02 HM Initial

4.6.9 PKTIO208: DRX_DEAD_NOT_READY, TX_BUFFER

Description: In state DRX_DEAD_NOT_READY, TX_BUFFER the PKTIO entity receives a signal that the data could be written. As only the DTI connection is dead, it is expected that the buffered data is handed over to the DIO driver and no flow control primitive to the (non-existent) DTI2 peer is sent. The next state is TX_READY, DRX_DEAD_READY (expected, but not tested).

<A> DTI2 DISCONNECT was by peer

 DTI2 DISCONNECT was by ACI

Variants:

<A>...

Preamble:

<A> PKTIO206A

 PKTIO207A

	ACI		PKTIO		DIO
(1)					
				PKT_DIO_SIGNAL_IND	
				(DRV_SIGTYPE_WRITE)	
				<=====	
(2)				PKT_DIO_GETBUFFER_REQ	
				=====	
(3)				PKT_DIO_GETBUFFER_CNF	
				<=====	
(4)				PKT_DIO_WRITE_REQ	
				=====	
(5)				PKT_DIO_WRITE_CNF	
				<=====	
MUTE	(1000)				

Parametrization

Primitive	Parameter	Value
(270)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_WRITE	
(271)	PKT_DIO_GETBUFFER_REQ	
device_no	DEVICE_47	
(272)	PKT_DIO_GETBUFFER_CNF	
retval	DRV_OK	
(4) (273)	PKT_DIO_WRITE_REQ	
device_no	DEVICE_47	
state	DIO_STATE_IP	
mask	DIO_MASK_NONE	
sdu	A_DL_SDU	
(274)	PKT_DIO_WRITE_CNF	
retval	DRV_OK	

Formatted: Bullets and Numbering

History: 15-Oct-02 HM Initial

4.6.10 PKTIO209: DRX_DEAD_NOT_READY, TX_BUFFER

Description: In state DRX_DEAD_NOT_READY, TX_BUFFER the DTI2 connection is reopened. As there is still buffered data in the PKTIO entity, it is expected that no flow control primitive is sent to the DTI2 peer. The state is DRX_NOT_READY until the driver indicates that the data could be written, then in the state transition to DRX_READY the flow control primitive is sent.

Preamble: PKTIO206A

```

    ACI                               PKTIO                               DIO
    |                                 |                                 |
(1) |   PKT_DTI_OPEN_REQ             |                                 |
    | *=====>*                     |                                 |
(2) |   DTI2_CONNECT_IND            |                                 |
    | *<=====*                       |                                 |
MUTE (1000)
(3) |   DTI2_CONNECT_RES             |                                 |
    | *=====>*                     |                                 |
(4) |   PKT_DTI_OPEN_CNF            |                                 |
    | *<=====*                       |                                 |
MUTE (1000)
(5) |                                 |   PKT_DIO_SIGNAL_IND         |
    |                                 |   (DRV_SIGTYPE_WRITE)      |
    |                                 | *<=====*                 |
(6) |                                 |   PKT_DIO_GETBUFFER_REQ     |
    |                                 | *=====>*                 |
(7) |                                 |   PKT_DIO_GETBUFFER_CNF    |
    |                                 | *<=====*                 |
(8) |                                 |   PKT_DIO_WRITE_REQ       |
    |                                 | *=====>*                 |
(9) |                                 |   PKT_DIO_WRITE_CNF      |
    |                                 | *<=====*                 |
(10)|   DTI2_READY_IND              |                                 |
    | *<=====*                       |                                 |
MUTE (1000)
    |                                 |                                 |

```

Parametrization

Primitive	Parameter	Value
(275)	PKT_DTI_OPEN_REQ	
device_no	DEVICE_47	
peer	PEER_SNDP	
link_id	LINK_ID_47	
dti_direction	DTI_DIRECTION_NORMAL	
(276)	DTI2_CONNECT_IND	
link_id	LINK_ID_47	
version	DTI_VERSION_10	
(277)	DTI2_CONNECT_RES	
link_id	LINK_ID_47	
version	DTI_VERSION_10	
(278)	PKT_DTI_OPEN_CNF	
device_no	DEVICE_47	
cause	PKTCS_SUCCESS	
(279)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_WRITE	

```

(280)          PKT_DIO_GETBUFFER_REQ
      device_no DEVICE_47

(281)          PKT_DIO_GETBUFFER_CNF
      retval    DRV_OK

+@- (282)      PKT_DIO_WRITE_REQ
      device_no DEVICE_47
      state     DIO_STATE_IP
      mask      DIO_MASK_NONE
      sdu       A_DL_SDU

(283)          PKT_DIO_WRITE_CNF
      retval    DRV_OK

(284)          DT12_READY_IND
      link_id   LINK_ID_47
    
```

Formatted: Bullets and Numbering

History: 15-Oct-02 HM Initial

4.6.11 PKTIO210: DRX_DEAD_READY, TX_READY, driver disconnects

Description: In state DRX_DEAD_READY, TX_READY the drivers signals disconnection.
 Note: Looks more or less like PKTIO108.

Preamble: PKTIO208A

```

          ACI          PKTIO          DIO
(1)      |            |            |
          |            | PKT_DIO_SIGNAL_IND |
          |            | (DRV_SIGTYPE_DISCONNECT) |
          |            | *<=====*> |
(2)      |            |            |
          |            | PKT_DIO_CLOSEDEVICE_REQ |
          |            | *=====*> |
(3)      |            |            |
          |            | PKT_DIO_CLOSEDEVICE_CNF |
          |            | *<=====*> |
(4)      |            |            |
          |            | PKT_DISCONNECT_IND |
          |            | *<=====*> |
MUTE (1000) |            |            |
          |            |            |
    
```

Parametrization

Primitive	Parameter	Value
(285)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_DISCONNECT	
(286)	PKT_DIO_CLOSEDEVICE_REQ	
device_no	DEVICE_47	
(287)	PKT_DIO_CLOSEDEVICE_CNF	
retval	DRV_OK	
(288)	PKT_DISCONNECT_IND	
device_no	DEVICE_47	
cause	PKTCS_DISCONNECT	

History: 15-Oct-02 HM Initial

4.6.12 PKTIO211: DRX_READY, DL packet received, DIO driver has one packet

Description: In state DRX_READY a packet from the DTI2 peer is received. It can be successfully written to the DIO driver as this is supporting more than only one packet. Subsequently a third DTI primitive comes in, as the PKTIO entity limits the number of TX packets in the system to avoid a memory overrun, this is not written to the driver anymore. The TX state is expected to become TX_BUFFER, the the DRX state is expected to become DRX_NOT_READY.

<A> DEVICE_47

 DEVICE_66

Preamble:

<A> PKTIO200A

 PKTIO200B

```

          ACI                               PKTIO                               DIO
(1)      |      DTI2_DATA_TEST_REQ          |      |      |
          |      *=====>                |      |      |
(2)      |      |      PKT_DIO_WRITE_REQ   |      |      |
          |      |      *=====>                |      |      |
(3)      |      |      PKT_DIO_WRITE_CNF   |      |      |
          |      |      *=====>                |      |      |
(4)      |      DTI2_READY_IND              |      |      |
          |      *<=====                |      |      |
MUTE (1000)
(4)      |      DTI2_DATA_TEST_REQ          |      |      |
          |      *=====>                |      |      |
MUTE (1000)
          |      |      |      |
    
```

Parametrization

Primitive	Parameter	Value
(289)	DTI2_DATA_TEST_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_SECOND_DL_SDU	
(290)	PKT_DIO_WRITE_REQ	
<A>	device_no	DEVICE_47
	device_no	DEVICE_66
state	DIO_STATE_IP	
mask	DIO_MASK_NONE	
sdu	A_SECOND_DL_SDU	
(291)	PKT_DIO_WRITE_CNF	
retval	DRV_OK	
(292)	DTI2_READY_IND	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
(293)	DTI2_DATA_TEST_REQ	
<A>	link_id	LINK_ID_47
	link_id	LINK_ID_66
parameters	NOT_USED	
sdu	A_THIRD_DL_SDU	

History: 15-Oct-02 HM Initial

4.7 Config primitives

4.7.1 PKTIO301: Config Primitive SEND

Description: PKTIO has opened driver and DTI connection. For test purposes the config primitive SEND is sent to PKTIO. This parameter indicates the number of octets to be transmitted.

Preamble: PKTIO022a

	ACI	PKTIO	DIO
	COMMAND (PKT CONFIG SEND 71 64)		
(1)			
(2)			
MUTE (1000)			

Parametrization

Primitive	Parameter	Value
(294)	PKT_DIO_WRITE_REQ	
device_no	DEVICE_47	
state	DIO_STATE_IP	
mask	DIO_MASK_NONE	
sdu	CONFIG_64_SDU	
(295)	PKT_DIO_WRITE_CNF	
retval	DRV_OK	

History: 05-Nov-02 HK Initial

4.7.2 PKTIO302: Driver returns correct looped back data after config primitive SEND

Description: After receiving a config primitive SEND the PKTIO entity has written data to the driver. Now the driver returns the same data. Length and content of the returned packet match the sent packet. The returned packet is passed to the DTI interface.

Preamble: PKTIO301

	ACI	PKTIO	DIO
(1)			
(2)			
(3)			

```

(4) | DTI2_DATA_TEST_IND |
    | *<-----* |
(5) | | PKT_DIO_READ_REQ |
    | | *----->* |
(6) | | PKT_DIO_READ_CNF |
    | | *-----* |
(7) | | PKT_DIO_READ_REQ |
    | | *----->* |
(8) | | PKT_DIO_READ_CNF |
    | | *-----* |
MUTE (1000) | |
    | |
    
```

Parametrization

Primitive	Parameter	Value
(296)	PKT_DIO_SIGNAL_IND	
device_no	DEVICE_47	
signal_type	DRV_SIGTYPE_READ	
(297)	PKT_DIO_GETDATA_REQ	
device_no	DEVICE_47	
(298)	PKT_DIO_GETDATA_CNF	
retval	DRV_OK	
state	DIO_SA	
sdu	CONFIG_64_SDU	
(299)	DTI2_DATA_TEST_IND	
link_id	LINK_ID_47	
parameters	NOT_USED	
sdu	CONFIG_64_SDU	
(300)	PKT_DIO_READ_REQ	
device_no	DEVICE_47	
(301)	PKT_DIO_READ_CNF	
retval	DRV_OK	
(302)	PKT_DIO_READ_REQ	
device_no	DEVICE_47	
(303)	PKT_DIO_READ_CNF	
retval	DRV_BUFFER_FULL	

History: 05-Nov-02 HK Initial

4.7.3 PKTIO303: Driver returns looped back data after config primitive SEND, wrong length

Description: After receiving a config primitive SEND the PKTIO entity has written data to the driver. The length of the returned packet does not match the length of the sent packet. A packet of length 8 and with all octets set to 0 is sent to the DTI interface to signal that the length was wrong.

Preamble: PKTIO301

ACI	PKTIO	DIO
(1)	 PKT_DIO_SIGNAL_IND (DRV_SIGTYPE_READ) 	
(2)	 PKT_DIO_GETDATA_REQ 	
(3)	 PKT_DIO_GETDATA_CNF 	
(4)	DTI2_DATA_TEST_IND 	
(5)	 PKT_DIO_READ_REQ 	
(6)	 PKT_DIO_READ_CNF 	
(7)	 PKT_DIO_READ_REQ 	
(8)	 PKT_DIO_READ_CNF 	
MUTE (1000)	 	

Parametrization

<u>Primitive</u>	<u>Parameter</u>	<u>Value</u>
(304) device_no signal_type	PKT_DIO_SIGNAL_IND DEVICE_47 DRV_SIGTYPE_READ	
(305) device_no	PKT_DIO_GETDATA_REQ DEVICE_47	
(306) retval state sdu	PKT_DIO_GETDATA_CNF DRV_OK DIO_SA A_THIRD_DL_SDU	
(307) link_id parameters sdu	DTI2_DATA_TEST_IND LINK_ID_47 NOT_USED CONFIG_WRONG_LEN	
(308) device_no	PKT_DIO_READ_REQ DEVICE_47	
(309) retval	PKT_DIO_READ_CNF DRV_OK	
(310) device_no	PKT_DIO_READ_REQ DEVICE_47	
(311) retval	PKT_DIO_READ_CNF DRV_BUFFER_FULL	

History: 06-Nov-02 HK Initial

4.7.4 PKTIO304: Driver returns looped back data after config primitive SEND, length is ok but content is wrong

Description: After receiving a config primitive SEND the PKTIO entity has written data to the driver. The length of the returned packet does match the length of the sent packet. But the content is not the same. A packet of length 8 and with all octets set to 0xFF is sent to the DTI interface to signal that the length was wrong.

Preamble: PKTIO301

	ACI	PKTIO	DIO
(1)		PKT_DIO_SIGNAL_IND	
		(DRV_SIGTYPE_READ)	
		<=====	
(2)		PKT_DIO_GETDATA_REQ	
		=====>	
(3)		PKT_DIO_GETDATA_CNF	
		<=====	
(4)	DTI2_DATA_TEST_IND		
	<=====		
(5)		PKT_DIO_READ_REQ	
		=====>	
(6)		PKT_DIO_READ_CNF	
		<=====	
(7)		PKT_DIO_READ_REQ	
		=====>	
(8)		PKT_DIO_READ_CNF	
		<=====	
MUTE (1000)			

Parametrization

Primitive	Parameter	Value
(312)	device_no	PKT_DIO_SIGNAL_IND
	signal_type	DEVICE_47
		DRV_SIGTYPE_READ
(313)	device_no	PKT_DIO_GETDATA_REQ
		DEVICE_47
(314)	retval	PKT_DIO_GETDATA_CNF
	state	DRV_OK
	sdu	DIO_SA
		CONFIG_64_WRONG_CONTENT
(315)	link_id	DTI2_DATA_TEST_IND
	parameters	LINK_ID_47
	sdu	NOT_USED
		CONFIG_WRONG_CONTENT
(316)	device_no	PKT_DIO_READ_REQ
		DEVICE_47
(317)	retval	PKT_DIO_READ_CNF
		DRV_OK
(318)	device_no	PKT_DIO_READ_REQ
		DEVICE_47

(322) link_id DTI2_GETDATA_REQ
 LINK_ID_47

History: 15-Oct-02 HM Initial

4.7.7 PKTIO307: Driver returns second correct looped back data after config primitive SEND

Description: After receiving a second config primitive SEND the PKTIO entity has written data to the driver. Now the driver returns the same data. Length and content of the returned packet match the sent packet. The returned packet is passed to the DTI interface.

Preamble: PKTIO306

	ACI		PKTIO		DIO
(1)				PKT_DIO_SIGNAL_IND	
				(DRV_SIGTYPE_READ)	
				<=====	
(2)				PKT_DIO_GETDATA_REQ	
				=====>	
(3)				PKT_DIO_GETDATA_CNF	
				<=====	
(4)		DTI2_DATA_TEST_IND			
		<=====			
(5)				PKT_DIO_READ_REQ	
				=====>	
(6)				PKT_DIO_READ_CNF	
				<=====	
(7)				PKT_DIO_READ_REQ	
				=====>	
(8)				PKT_DIO_READ_CNF	
				<=====	
MUTE	(1000)				

Parametrization

Primitive	Parameter	Value
(323)	device_no	PKT_DIO_SIGNAL_IND
	signal_type	DEVICE_47
		DRV_SIGTYPE_READ
(324)	device_no	PKT_DIO_GETDATA_REQ
		DEVICE_47
(325)	retval	PKT_DIO_GETDATA_CNF
	state	DRV_OK
	sdu	DIO_SA
		CONFIG_32_SDU
(326)	link_id	DTI2_DATA_TEST_IND
	parameters	LINK_ID_47
	sdu	NOT_USED
		CONFIG_32_SDU

```
(327)          PKT_DIO_READ_REQ
      device_no  DEVICE_47

(328)          PKT_DIO_READ_CNF
      retval    DRV_OK

(329)          PKT_DIO_READ_REQ
      device_no  DEVICE_47

(330)          PKT_DIO_READ_CNF
      retval    DRV_BUFFER_FULL
```

History: 05-Nov-02 HK Initial

5 Suites

```
/* Set internal routings */
INT_ROUTING:          PKTIO000;

/* Bring device 47 to initial ready state, flow control */
SUI_PKTIO22A:         INT_ROUTING,
                     PKTIO010A,
                     PKTIO011A,
                     PKTIO020A,
                     PKTIO021A,
                     PKTIO022A;

/* Bring device 66 to initial ready state, flow control */
SUI_PKTIO22B:         INT_ROUTING,
                     PKTIO010B,
                     PKTIO011B,
                     PKTIO020B,
                     PKTIO021B,
                     PKTIO022B;

/* Mix it wildly */
SUI_PKTIO22A_22B:     INT_ROUTING,
                     PKTIO010A,
                     PKTIO011A,
                     PKTIO010B,
                     PKTIO020A,
                     PKTIO011B,
                     PKTIO021A,
                     PKTIO020B,
                     PKTIO021B,
                     PKTIO022B,
                     PKTIO022A;

/* Reject opening of DTI connection, device 47 */
SUI_PKTIO23A:         INT_ROUTING,
                     PKTIO010A,
                     PKTIO011A,
                     PKTIO020A,
                     PKTIO023A;

/* Reject opening of DTI connection, device 47 */
SUI_PKTIO23B:         INT_ROUTING,
                     PKTIO010B,
                     PKTIO011B,
                     PKTIO020B,
                     PKTIO023B;
```

```
/* Reject opening of DTI connection, mixed */
SUI_PKTIO23A_23B:    INT_ROUTING,
                    PKTIO010B,
                    PKTIO010A,
                    PKTIO011B,
                    PKTIO011A,
                    PKTIO020B,
                    PKTIO020A,
                    PKTIO023B,
                    PKTIO023A;
```

```
/* Mix 22A and 23B */
SUI_PKTIO22A_23B:    INT_ROUTING,
                    PKTIO010A,
                    PKTIO010B,
                    PKTIO011A,
                    PKTIO011B,
                    PKTIO020A,
                    PKTIO020B,
                    PKTIO021A,
                    PKTIO022A,
                    PKTIO023B;
```

```
/* Some UL data transfer, device 47 */
SUI_PKTIO_112A:      INT_ROUTING,
                    PKTIO010A,
                    PKTIO027A,
                    PKTIO109A,
                    PKTIO111A,
                    PKTIO112A;
```

```
/* Some UL data transfer, device 66 */
SUI_PKTIO_112B:      INT_ROUTING,
                    PKTIO010B,
                    PKTIO027B,
                    PKTIO109B,
                    PKTIO111B,
                    PKTIO112B;
```

```
/* Some other UL data transfer, device 47 */
SUI_PKTIO_113A:      INT_ROUTING,
                    PKTIO010A,
                    PKTIO027A,
                    PKTIO109A,
                    PKTIO111A,
                    PKTIO113A;
```

```
/* Some other UL data transfer, device 66 */
SUI_PKTIO_113B:      INT_ROUTING,
                    PKTIO010B,
                    PKTIO027B,
                    PKTIO109B,
                    PKTIO111B,
                    PKTIO113B;
```

```
/* Mix device 47 and device 66 */
SUI_PKTIO_112A_112B: INT_ROUTING,
                    PKTIO010A,
                    PKTIO010B,
                    PKTIO027A,
```

```

                PKTIO027B,
                PKTIO109A,
                PKTIO109B,
                PKTIO111A,
                PKTIO111B,
                PKTIO112A,
                PKTIO112B;

/* And again */
SUI_PKTIO_113A_113B:  INT_ROUTING,
                PKTIO010B,
                PKTIO010A,
                PKTIO027B,
                PKTIO027A,
                PKTIO109B,
                PKTIO109A,
                PKTIO111B,
                PKTIO111A,
                PKTIO113B,
                PKTIO113A;

/* Device 47, downlink data */
SUI_PKTIO_207A:      INT_ROUTING,
                PKTIO010A,
                PKTIO011A,
                PKTIO020A,
                PKTIO021A,
                PKTIO022A,
                PKTIO200A,
                PKTIO201A,
                PKTIO207A;

/* Device 66, downlink data */
SUI_PKTIO_207B:      INT_ROUTING,
                PKTIO010B,
                PKTIO011B,
                PKTIO020B,
                PKTIO021B,
                PKTIO022B,
                PKTIO200B,
                PKTIO201B,
                PKTIO207B;

/* Mix 47 and 66 */
SUI_PKTIO_207A_207B:  INT_ROUTING,
                PKTIO010A,
                PKTIO010B,
                PKTIO011A,
                PKTIO011B,
                PKTIO020A,
                PKTIO020B,
                PKTIO021A,
                PKTIO021B,
                PKTIO022A,
                PKTIO022B,
                PKTIO200A,
                PKTIO200B,
                PKTIO201A,
                PKTIO201B,
                PKTIO207A,
                PKTIO207B;
```

/* Mix downlink and uplink */

SUI_PKTIO_113B_207A: INT_ROUTING,
PKTIO010A,
PKTIO011A,
PKTIO010B,
PKTIO020A,
PKTIO021A,
PKTIO027B,
PKTIO022A,
PKTIO200A,
PKTIO109B,
PKTIO201A,
PKTIO111B,
PKTIO113B,
PKTIO207A;

/* Some very long modification for device 47 to play around with */

SUI_PKTIO_058A: INT_ROUTING,
PKTIO010A,
PKTIO011A,
PKTIO020A,
PKTIO021A,
PKTIO022A,
PKTIO200A,
PKTIO201A,
PKTIO051C,
PKTIO057A,
PKTIO058A;

/* Mix 058A (device 47) and 207B (device 66) */

SUI_PKTIO_058A_207B: INT_ROUTING,
PKTIO010A,
PKTIO011A,
PKTIO010B,
PKTIO011B,
PKTIO020A,
PKTIO021A,
PKTIO020B,
PKTIO021B,
PKTIO022A,
PKTIO022B,
PKTIO200A,
PKTIO200B,
PKTIO201A,
PKTIO201B,
PKTIO051C,
PKTIO057A,
PKTIO058A,
PKTIO207B;

SUI_PKTIO_058A_113B: INT_ROUTING,
PKTIO010A,
PKTIO011A,
PKTIO010B,
PKTIO020A,
PKTIO021A,
PKTIO022A,
PKTIO027B,
PKTIO200A,
PKTIO109B,

PKTIO201A,
PKTIO051C,
PKTIO111B,
PKTIO057A,
PKTIO058A,
PKTIO113B;

/*
History: 30-Oct-02 HM Initial
*/

Appendices

A. Acronyms

DS-WCDMA Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000)

Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: <http://www.imt-2000.org/>>