PM73122

AAL1GATOR-32 DEVICE DRIVER

DEVICE DRIVER RELEASE NOTES

PROPRIETARY AND CONFIDENTIAL

RELEASE

ISSUE 2: AUGUST, 01

REVISION HISTORY

Issue No.	Issue Date	Details of Change
1		Document created
2	August 2001	Comments added to describe differences between beta-1.0 and rel-1.0. File versions updated.

1 ABOUT THIS RELEASE

This is the Production release (rel-1.0) of the PM73122 (AAL1GATOR-32) device driver. Rel-1.0 is intended for use with PM73122 Revision A and Revision C AAL1GATOR-32, AAL1GATOR-8 and AAL1GATOR-4 devices. Rel-1.0 is functionally complete and has been tested by PMC-Sierra as described in Section 3 below.

2 WHAT'S INCLUDED IN THIS RELEASE

This production release of the AAL1GATOR-32 driver includes the following files:

Directory	Filename	File Version
/src	al3_api.c	1.6
	al3_api2.c	1.9
	al3_chan.c	1.13
	al3_diag.c	1.6
	al3_dpr.c	1.10
	al3_hw.c	1.6
	al3_init.c	1.9
	al3_isr.c	1.8
	al3_rtos.c	1.7
	al3_stat.c	1.6
/inc	al3_api.h	1.7
	al3_chan.h	1.6
	al3_dev.h	1.6
	al3_diag.h	1.5
	al3_dpr.h	1.5

Directory	Filename	File Version
/inc	al3_fns.h	1.2
	al3_hw.h	1.6
	al3_init.h	1.4
	al3_isr.h	1.6
	al3_mdb.h	1.6
	al3_rtos.h	1.6
	al3_stat.h	1.4
	al3_typs.h	1.2
/example	al3_app.c[1]	1.6
	al3_app.h[1]	1.4
/	Makefile	1.5
/	software.lic	n/a
/	al3_relnotes_r2.pdf (this file)	n/a

Notes:

1. al3_app.c, al3_app.h contain example callback function implementation and example code.

3 TESTING STATUS OF THIS RELEASE

All AAL1GATOR-32 device driver Application Programming Interface (API) functions have been tested in-house using the PM73122 on a proprietary PMC-Sierra, Inc. application platform running VxWorks™ RTOS. As of the publication date of this document, there are no known bugs in this release. However, the customer is advised to consult our web site (http://www.pmc-sierra.com) for any relevant errata that may have been issued subsequent to the publication of this document.

4 WHAT'S NEW IN THIS RELEASE?

rel-1.0: Production release for the PM73122 device driver.

- Support for DBCES queues has been added.
- Support for AAL0 queues has been added.
- Interrupt Service Routine architecture has been modified to fix bugs and improve performance.

5 DEVICE ERRATA SOFTWARE WORKAROUNDS

This driver implements software workarounds for the Revision A device errata which, depending on the revision code of the AAL1gator device as read by the initialization routines, will be automatically enabled (Rev A) or disabled (Rev C). An option exists for those users who are only using Revision C devices to compile the driver without any workaround code by using the compile switch, "AL3_CSW_DISABLE_ALL_WORKAROUNDS". The only benefit of doing this is a reduction in the size of the compiled code. There are no workarounds required for Revision C devices. There are, however, some DBCES limitations in Revision C devices. The reader is referred to PM73122 Revision C Device Errata (PMC-2011072) for further details.

The following describes the workarounds implemented for Revision A device errata. The reader is referred to PM73122 Revision A Device Errata (PMC-2000421) for further details.

- SRTS Queue Underrun Impacts Other Lines in A1SP – When using the Synchronous Residual Time Stamp (SRTS) scheme in Unstructured Data Format (UDF) mode, the buffer that contains received SRTS nibbles underruns for a specific line. A signal is sent to the internal clock synthesizer (CGC) that indicates that all the lines are in SRTS underrun. This causes all lines within an A1SP block to revert to normal clocking. In order to prevent this problem, the driver defers enabling of the SRTS bit until the SRTS queue is active and not in the underrun state. The initialization of this workaround is in the `aalInitRegsLine()` (file: `al3_init.c`) and the active switching of the SRTS bit is in `aalCoreDPR()` (file `al3_dpr.c`). (This problem is only present on unstructured lines)

- LOW_CDV Queue Does Not Activate/Deactivate Correctly – For Unstructured Data Format (UDF) lines there is a LOW_CDV bit which can be set in the LIN_STR_MODE memory register which will cause cells to be scheduled every 47 bytes instead of every frame. This eliminates the CDV caused by the scheduler. This mode can only be used in UDF-ML mode when BYTES_PER_CELL is 47. In High Speed mode cells are always scheduled every 47 bytes which assumes that partial cells are never used in HS mode. If a queue is "added" to the ADDQ_FIFO and LOW_CDV is set, data errors may result because the FRAMES_PER_CELL field is defaulted to a 1 instead of 2. Also if a queue is deactivated by clearing the ACTIVE bit when LOW_CDV is set, the CSD can enter a lock condition. In order to prevent this problem, the driver always adds and deletes queues with the LOW_CDV bit cleared and the SUPPRESS_XMT bit set. The workaround is in `aalActivateQueueUnstr()` (file: `al3_chan.c`). (This problem is only present on unstructured UDF lines)

6 REPORTING PROBLEMS

Please refer to the revision numbers in the files when reporting problems. For technical support, please contact PMC-Sierra by e-mail at apps@pmc-sierra.com or by telephone at 604-415-4533.

PMC-Sierra, Inc.
8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: (604) 415-6000
Fax: (604) 415-6200

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Application Information: apps@pmc-sierra.com
Web Site: http://www.pmc-sierra.com

PMC-2000541 (R2) Issue date: August 2001