

PM8315

TEMUX DEVICE DRIVER

HIGH DENSITY T1/E1 FRAMER WITH
INTEGRATED VT/TU MAPPER AND M13 MULTIPLEXER

DEVICE DRIVER ERRATA

ISSUE 1: OCTOBER 2000

PUBLIC REVISION HISTORY

Issue No.	Issue Date	Details of Change
1	October 2000	Document created.

CONTENTS

1	ISSUE 1 DEVICE DRIVER ERRATA.....	1
1.1	DRIVER IDENTIFICATION.....	1
2	TEMUX DEVICE DRIVER FUNCTIONAL DEFICIENCIES OVERVIEW	2
3	TEMUX DEVICE DRIVER FUNCTIONAL DEFICIENCY DETAILS	4
3.1	MEMORY.....	4
3.1.1	MEMORY CORRUPTION WHEN PASSING A NULL PTR TO TEMUXDPR API.....	4
3.2	DECLARATIONS.....	5
3.2.1	FUNCTION PROTOTYPE SYSTEMUXDPRTASK.....	5
3.2.2	FUNCTION PROTOTYPE TEMUXLOOPMX23	5
3.2.3	STMX_DPV STRUCT	6
3.3	ADDED FUNCTIONS.....	7
3.3.1	TEMUXSBITRIBRESET.....	7
3.4	TEMUX DEVICE DRIVER ISR.....	7
3.4.1	INCORRECT INDEXING OF ISV.....	7
3.4.2	IMPROPER NOTIFICATION OF SIGX COSS INTERRUPTS.....	8
3.5	TEMUXDPR	9
3.5.1	ISV BUFFER RECEIVED BUT NEVER FREED	9

1 ISSUE 1 DEVICE DRIVER ERRATA

This document is the errata notice for beta-1.0 release of the TEMUX (PM8315-PI) device driver. The beta-1.0 release of the TEMUX device driver and issue 1 errata supersede all prior editions and versions of the device driver.

1.1 DRIVER IDENTIFICATION

The information in this document applies to the beta-1.0 release of the PM8315 TEMUX device driver only. The driver revision number is indicated in the comment section of each file under Modification History. The complete beta-1.0 release of the TEMUX driver includes the following files:

Table 1 - TEMUX Device Driver Files

Directory	Filename	File Version
source	tmx_api.c	13
	tmx_app.c	4
	tmx_diag.c	6
	tmx_dpr.c	12
	tmx_hw.c	9
	tmx_isr.c	12
	tmx_rtos.c	8
	tmx_stat.c	5
	tmx_util.c	7
	headers	temux.h
tmx_api.h		13
tmx_app.h		3
tmx_dev.h		11
tmx_dpr.h		6
tmx_hw.h		9
tmx_isr.h		7

Directory	Filename	File Version
headers	tmx_mdb.h	9
	tmx_rtos.h	9
	tmx_util.h	4
examples	app.c ¹	3
	app.h ¹	3

Notes:

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

2 TEMUX DEVICE DRIVER FUNCTIONAL DEFICIENCIES OVERVIEW

This section outlines the known functional deficiencies of the beta-1.0 release of the PM8513 TEMUX device driver. The errata are explained in more detail in section 3 of this document.

Table 2 - FUNCTIONAL DEFICIENCIES SUMMARY LIST

#	Discrepancy	Workaround
3.1	Memory	
3.1.1	Memory Corruption when passing Null Ptr to temuxDPR	Use default implementation
3.2	Device Driver Compiler	
3.2.1	Function prototype sysTemuxDPRTask – declaration missing	Yes – code addition
3.2.2	Function prototype temuxLoopx23 – declaration discrepancy	Yes – code edit
3.2.3	STM_DPV struct – declaration missing	Yes – code edit
3.3	Added Functions	
3.3.1	TemuxSBITribReset function added to perform configuration reset on SBI bus on Overrun or Underrun status	Function added in next release of software driver
3.4	Device Driver ISR	
3.4.1	Incorrect Indexing of ISV	Yes – code edit
3.4.2	Improper notification of SIGX COSS interrupts	Yes – code edit
3.4	TemuxDPR	

#	Discrepancy	Workaround
3.4.1	ISV Buffer received but never freed	Yes – code edit

3 TEMUX DEVICE DRIVER FUNCTIONAL DEFICIENCY DETAILS

3.1 Memory

3.1.1 Memory Corruption When Passing a Null Ptr to temuxDPR API

Description

Passing a Null to temuxDPR may cause memory corruption.

Workaround

There are two methods for implementation of the ISR/DPR chain. The default method requires the Application to:

Call sysTemuxISVBufferGet() to get a new buffer
Pass that buffer to temuxISR() for processing
SEND the buffer to the DPR Task
RECEIVE the buffer and pass it as a parameter to temuxDPR()

The alternate implementation of the ISR/DPR chain requires the Application to:

call temuxISR() with a NULL buffer pointer
(temuxISR() automatically SENDs the buffer)
call temuxDPR() from within the DPR Task with a NULL buffer pointer
(temuxDPR() automatically RECEIVES the buffer)

The default implementation works. The workaround for this errata is to use the default method, not the alternate method.

Performance Without Workaround

If the Application uses the alternate implementation, the driver will use a random memory address as if it were an ISV buffer pointer, thereby corrupting memory or crashing the driver.

3.2 Declarations

3.2.1 Function Prototype sysTemuxDPRTask

Description

Function prototype extern sysTemuxDPRTask does not have a corresponding declaration.

Workaround

In the file tmx_rtos.c, add the following declaration:

```
extern void sysTemuxDPRTask (void);
```

Performance Without Workaround

This is a compiler issue only.

3.2.2 Function prototype temuxLoopMX23

Description

Function prototype extern INT4 temuxLoopMX23 (sTMX_HNDL, UINT2, UINT2, BOOLEAN) differs from declaration INT4 temuxLoopMX23(sTMX_HNDL devId,UINT2 mxNum, UINT2 up, UINT2 ais).

Workaround

In the files `tmx_dpr.c` and `tmx_api.h`, change the function declaration from:

```
INT4 temuxLoopMX23 (sTMX_HNDL devId, UINT2 mxNum, UINT2 up, UINT2 ais)
```

to:

```
INT4 temuxLoopMX23 (sTMX_HNDL devId, UINT2 mxNum, UINT2 up, BOOLEAN ais)
```

Performance Without Workaround

This is a compiler issue only.

3.2.3 STMX_DPV struct

Description

TEMUX driver is missing the declaration of the `sTMX_DPV` structure.

Workaround

In the file `tmx_dpr.h`, make a copy of the structure `sTMX_DPV_IO` and rename all of the elements in one of the copies to `sTMX_DPV` (as shown below)

```
struct tmx_dpv  
{  
    UINT4 data;  
    UINT4 channels;  
};  
typedef struct tmx_dpv sTMX_DPV;
```

Performance Without Workaround

This is a compiler issue only.

3.3 Added Functions

3.3.1 TemuxSBITribReset

Description

When a Tributary on the Insert SBI bus goes into Overrun or Underrun status, the configuration must be reset.

Workarounds

An API function `temuxSBITribReset()` has been added to the next release of the software driver which performs a configuration reset on the Insert SBI Bus Tributary. The USER of the driver must call this routine when an Overrun or Underrun occurs on an Insert SBI Bus Tributary.

Performance Without Workaround

Without the new function or if the Application fails to call the function when an overrun or underrun occurs, the SBI Insert link will lock up.

3.4 Temux Device Driver ISR

3.4.1 Incorrect Indexing of ISV

Description

In `coreISR()` in `tmx_isr.c`, the framer array member of the ISV is not indexed correctly. It is indexed from 0 to 27 instead of 1 through 28.

Workaround

In the **file** `tmx_isr.c`, in the `coreISR()`, change the following lines:

```
chanNum = 0x00;
```

```
while (channels || (polling && chanNum < NUM_T1))
```

to:

```
chanNum = 0x01;
```

```
while (channels || (polling && chanNum <= NUM_T1))
```

Performance Without Workaround

The incorrect indexing causes information to be interpreted as coming from the wrong Tributary.

3.4.2 Improper Notification of SIGX COSS Interrupts

Description

In `coreISR()` in `tmx_isr.c`, when populating the ISV for SIGX COSS interrupts a logical OR operator is used where a binary OR operator is required.

Workarounds

In the file `tmx_isr.c`, in the `coreISR()`, change the following lines:

```
pISV->framer[index].sigx.coss =
  ( ((sysTemuxReadReg(TMX_FR_SIGX_CFG(base,chanNum)) & 0x3F) << 24)
    || ((sysTemuxReadReg(TMX_FR_SIGX_STAT(base,chanNum)) & 0xFF) << 16)
    || ((sysTemuxReadReg(TMX_FR_SIGX_ADDR(base,chanNum)) & 0xFF) << 8)
    || ((sysTemuxReadReg(TMX_FR_SIGX_DATA(base,chanNum)) & 0xFF)    ));
```

to:

```
pISV->framer[index].sigx.coss =
  ( ((sysTemuxReadReg(TMX_FR_SIGX_CFG(base,chanNum)) & 0x3F) << 24)
    | ((sysTemuxReadReg(TMX_FR_SIGX_STAT(base,chanNum)) & 0xFF) << 16)
    | ((sysTemuxReadReg(TMX_FR_SIGX_ADDR(base,chanNum)) & 0xFF) << 8)
    | ((sysTemuxReadReg(TMX_FR_SIGX_DATA(base,chanNum)) & 0xFF)    ));
```

Performance Without Workaround

This causes the USER to not receive proper notification of SIGX COSS interrupts.

3.5 TemuxDPR

3.5.1 ISV Buffer Received But Never Freed

Description

In file `tmx_dpr.c` function `coreDPR()`: ISV buffer received, but never freed.

Workarounds

In the file `tmx_dpr.c`, in the `coreDPR()`, change the final line:

```
return TMX_OK;
```

to:

```
if (pBuf == NULL) sysTemuxISVBufferRtn(pISV);  
return TMX_OK;
```

Performance Without Workaround

Without the added line of code, the Driver would eventually exhaust the supply of buffers and the DPR callbacks would cease.

NOTES

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
105-8555 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

(604) 415-4533

Web Site: <http://www.pmc-sierra.com>

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

© 2000 PMC-Sierra, Inc.

PMC-2001481 (R1) Issue date: October 2000