

# ERRATA SHEET

**Date:** June 7, 2008  
**Document Release:** Version 1.4  
**Device Affected:** LPC2103

This errata sheet describes both the known functional problems and any deviations from the electrical specifications known at the release date of this document.

Each deviation is assigned a number and its history is tracked in a table at the end of the document.

2008 June 7

**Document revision history**

Rev	Date	Description
1.4	June 7, 2008	Added VBAT.1
1.3	February 12, 2008	1. Added Rev 'A' and updated the errata history table
1.2	July 13, 2007	1. Added MAM.2 2. Added I <sup>2</sup> C1.1 3. Updated ESD.1
1.1	February 27, 2006	This table was added after document revision 1.1

**Identification:**

The LPC2103 devices typically have the following top-side marking:

LPC2103xxx

xxxxxxx

xxYYWW R

The last letter in the third line (field 'R') will identify the device revision. This Errata Sheet covers the following revisions of the LPC2103:

Revision Identifier (R)	Comment
-	Initial device revision
A	Second device revision

Field 'YY' states the year the device was manufactured. Field 'WW' states the week the device was manufactured during that year.

**Errata Overview - Functional Problems**

Functional Problem	Short Description	Device Revision the problem occurs in
Core.1	Incorrect update of the Abort link register	-, A
MAM.1	Incorrect read of data from SRAM	-
MAM.2	Code execution failure can occur with MAM Mode 2	-, A
SPI.1	Incorrect shifting of data in slave mode at lower frequencies	-
SSP.1	Initial data bits/clocks corrupted in SSP transmission	-, A
Timer.1	Timer Counter reset occurs on incorrect edge in counter mode	-
Vdd.1	Device may not work properly under increased power consumption	-
I <sup>2</sup> C1.1	I <sup>2</sup> C1 pins are not bi-directional GPIO pins	-

**Errata Overview - AC/DC Deviations**

AC/DC Deviations	Short Description	Device Revision the deviation occurs in
ESD.1	ESD weakness on RTCX1 pin	-
VBAT.1	Increased power consumption on the VBAT pin when the VDD <sub>(1V8)</sub> core pin is left floating	-

**Errata Notes**

Note	Short Description
NA	NA

## Functional Problems of LPC2103

### Core.1 Incorrect update of the Abort Link register in Thumb state

**Introduction:** If the processor is in Thumb state and executing the code sequence STR, STMIA or PUSH followed by a PC relative load, and the STR, STMIA or PUSH is aborted, the PC is saved to the abort link register.

**Problem:** In this situation the PC is saved to the abort link register in word resolution, instead of half-word resolution.

**Conditions:**

The processor must be in Thumb state, and the following sequence must occur:

<any instruction>

<STR, STMIA, PUSH> <---- data abort on this instruction

LDR rn, [pc,#offset]

In this case the PC is saved to the link register R14\_abt in only word resolution, not half-word resolution. The effect is that the link register holds an address that could be #2 less than it should be, so any abort handler could return to one instruction earlier than intended.

**Work around:** In a system that does not use Thumb state, there will be no problem.

In a system that uses Thumb state but does not use data aborts, or does not try to use data aborts in a recoverable manner, there will be no problem.

Otherwise the workaround is to ensure that a STR, STMIA or PUSH cannot precede a PC-relative load. One method for this is to add a NOP before any PC-relative load instruction. However this is would have to be done manually.

### MAM.1 Incorrect read of data from SRAM after Reset and MAM is not enabled or partially enabled

**Introduction:** The Memory Accelerator Module (MAM) provides accelerated execution from the on-chip flash at higher frequencies.

**Problem:** If code is running from on-chip Flash, a write to an SRAM location followed by an immediate read from the same SRAM location corrupts the data been read. For instance, a stack push operation immediately followed by a stack pop operation

**Work-around:** User code should enable the MAM after Reset and before any RAM accesses; this means MAMTIM and MAMCR should be set as follows:

**MAMTIM:** For CPU clock frequencies slower than 20MHz, set MAMTIM to 0x01. For CPU clock frequencies between 20MHz and 40MHz, set MAMTIM to 0x02, and for values above 40MHz set MAMTIM to 0x03.

**MAMCR:** Set MAMCR to 0x02 (MAM functions fully enabled)

MAMTIM should be written before MAMCR.

### MAM.2 Under certain conditions in MAM Mode 2 code execution out of internal Flash can fail

**Introduction:** The MAM block maximizes the performance of the ARM processor when it is running code in Flash memory. It includes three 128-bit buffers called the Prefetch Buffer, the Branch Trail Buffer

and the data buffer. It can operate in 3 modes; Mode 0 (MAM off), Mode 1 (MAM partially enabled) and Mode 2 (MAM fully enabled).

- Problem:** Under certain conditions when the MAM is fully enabled (Mode 2) code execution from internal Flash can fail. The conditions under which the problem can occur is dependent on the code itself along with its positioning within the Flash memory.
- Workaround:** If the above problem is encountered then Mode 2 should not be used. Instead, partially enable the MAM using Mode 1.

### **SPI.1 Incorrect shifting of data in slave mode at lower frequencies**

**Introduction:** In slave mode, the SPI can set the clock phase (CPHA) to 0 or 1.

**Problem:** Consider the following conditions:

- a. SPI is configured as a slave (with CPHA=0).
- b. SPI is running at a low frequency.

In slave mode, the SPIF (SPI Transfer Complete Flag) bit is set on the last sampling edge of SCK. If CPHA is set to 0 then the last sampling edge of SCK would be the rising edge.

Under the above conditions, if the SPI Data Register (SPDR) is written to less than a half SCLK cycle after the SPIF bit is set (this would happen if the SPI frequency is low) then the SPDR will shift data one clock early for the upcoming transfers.

Lowering the SPI frequency would increase the likelihood of the SPDR write happening in the first half SCK cycle of the last sampling clock.

**Work-around:** There are two possible workarounds:

- 1) Use CPHA=1.
- 2) If the data is shifted incorrectly when CPHA is set to 0 then delaying the write to SPDR after the half SCK cycle of the last sampling clock would resolve this issue.

### **SSP.1 Initial data bits/clocks of the SSP transmission are shorter than subsequent pulses at higher frequencies**

**Introduction:** The SSP is a Synchronous Serial Port (SSP) controller capable of operation on a SPI, 4-wire SSI or a Microwire bus. The SSP can operate at a maximum speed of 30MHz and it referred to as SPI1 in the device documentation.

- Problem:** At high SSP frequencies, it is found that the first four pulses are shorter than the subsequent pulses. At 30MHz, the first pulse can be expected to be approximately 10ns shorter and the second pulse around 5ns shorter. The remaining two pulses are around 2ns shorter than subsequent pulses. At 25MHz, the length of the first pulse would be around 7ns shorter. The subsequent three pulses are around 2ns shorter. At 20MHz only the first pulse is affected and it is around 2ns shorter. All subsequent pulses are fine. The deviation of the initial data bits/clocks will decrease as the SSP frequency decreases.

**Work-around:** None.

### **Timer.1 In counter mode, the Timer Counter reset does not occur on the correct incoming edge**

**Introduction:** Timer0 and Timer1 can be used in a counter mode. In this mode, the Timer Counter register can be incremented on rising, falling or both edges which occur on a selected CAP input pin.

This counter mode can be combined with the match functionality to provide additional features. One of the features would be to reset the Timer Counter register on a match. The same would also apply for Timer1.

**Problem** The Timer Counter reset does not trigger on the same incoming edge when the match takes place between the corresponding Match register and the Timer Counter register. The Timer Counter register will be reset only on the next incoming edge.

**Work-around:** There are two possible workarounds:

1. Combine the Timer Counter reset feature with the “interrupt on match” feature. The interrupt on match occurs on the correct incoming edge. In the ISR, the Timer Counter register can also be reset. This solution can only work if no edges are expected during the duration of the ISR.
2. In this solution, the “interrupt on match” feature is not used. Instead, the following specific initialization can achieve the counting operation:

- a. Initialize the Timer Counter register to 0xFFFFFFFF.
- b. If “n” edges have to be counted then initialize the corresponding Match register with value n-1. For instance, if 2 edges need to be counted then load the Match register with value 1

More details on the above example:

- a. Edge 1- Timer overflows and Timer Counter (TC) is set to 0.
- b. Edge 2- TC=1. Match takes place.
- c. Edge 3- TC=0.
- d. Edge 4- TC=1. Match takes place.
- e. Edge 5- TC=0.

### **Vdd.1 Device may not work properly under increased power consumption conditions**

**Introduction:** From the Vdd<sub>1V8</sub> pin to the ARM7 core, there is a voltage drop. This voltage drop increases with higher currents (or higher power consumption). Higher system frequency and/or faster peripherals increase power consumption thereby increasing the voltage drop from the Vdd<sub>1V8</sub> pin to the core.

**Problem:** Under increased power consumption conditions, the device may not work properly. The likelihood of the problem showing up increases if the device is run at higher speeds (approaching 70MHz) and/or if the peripherals are running at close to system frequency speed (i.e If the APBDIV register is set to a value lower than 4 which would imply that the peripherals are run faster than 1/4 of the system frequency).

Also, having a lower voltage on Vdd<sub>1V8</sub> pin (e.g. below 1.8V) can cause this issue to surface.

**Work-around:** Increase the voltage on Vdd<sub>1V8</sub> to 1.95V. It is not harmful to the device if this voltage reaches 2V.

### **I<sup>2</sup>C1.1 I<sup>2</sup>C1 pins are not bi-directional GPIO pins**

**Introduction:** There are two I<sup>2</sup>C interfaces, I<sup>2</sup>C0 and I<sup>2</sup>C1. I<sup>2</sup>C0 functions are shared as alternate functions on port pins P0.2 and P0.3. I<sup>2</sup>C1 functions are shared on port pins P0.17 and P0.18.

**Problem:** I<sup>2</sup>C1 pins are currently open-drain output pins but they should be regular bi-directional GPIO pins. I<sup>2</sup>C0 pins are configured as open-drain output pins (for I<sup>2</sup>C bus compliance).

**Work-around:** None.

## AC/DC Deviations

### **ESD.1: The device does not meet the 2kV ESD requirements on the RTCX1 pin**

Introduction: The LPC2103 is rated for 2kV ESD. The RTCX1 pin is the input pin for the RTC oscillator circuit.

Problem: The LPC2103 does not meet the required 2kV ESD specified.

Workarounds: Observe proper ESD handling precautions for the RTCX1 pin.

### **VBAT.1: Increased power consumption on the VBAT pin when the VDD<sub>(1V8)</sub> core pin is left floating**

Introduction: The LPC2103 has a VDD<sub>(1V8)</sub> core pin which provides power to the internal circuitry and also, has a VBAT pin which provides power only to the RTC (Real Time Clock). VBAT pin can be to an external battery or to the 3.3 v I/O port supply (VDD<sub>(3V3)</sub> pin) used by the device.

Problem: When the VDD<sub>(1V8)</sub> core pin is floated, power consumption increases on the VBAT pin.

Workarounds: VDD<sub>(1V8)</sub> core pin must always be connected to its power supply or to ground.