ES_LPC2134

Errata sheet LPC2134

Rev. 2 — 1 March 2011 | Errata sheet

Document information

Info	Content
Keywords	LPC2134 errata
Abstract	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.

Revision history

Rev	Date	Description
2	20110301	• The format of this errata sheet has been redesigned to comply with the new identity guidelines of NXP Semiconductors. • Added ADC.1
1.9	20080607	• Added WDT.1. • Added table for errata notes. • Added Errata Note 4.
1.8	20070709	• Errata history table for MAM.2 was updated.
1.7	20070608	• Added MAM.2. • Added ESD.2.
1.6	20060616	• This table was added after document revision 1.6.

Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

1. Product identification

The LPC2134 devices typically have the following top-side marking:

LPC2134xxx

xxxxxxx

xxYYWW R

The last letter in the last line (field 'R') will identify the device revision. This Errata Sheet covers the following revisions of the LPC2134:

Table 1. Revision overview table

Revision identitfier	Revision description
'-'	Initial device revision
'A'	Second device revision
'B'	Third device revision
'C'	Fourth device revision
'D'	Fifth device revision

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

2. Errata overview

Table 2. Functional problems table

Functional problems	Short description	Revision identifier	Detailed description
Core.1	Incorrect update of the Abort link register	'-', 'A', 'B', 'C', 'D'	Section 3.1
Timer.1	Missed Interrupt Potential	'-', 'A', 'B'	Section 3.2
PWM.1	Missed Interrupt Potential for Match Functionality	'-', 'A', 'B'	Section 3.3
Timer.2	Timer Counter reset occurs on incorrect edge in counter mode	'-', 'A', 'B', 'C', 'D'	Section 3.4
Power Down.1	Restricted Vdd while device is in power down mode	'A'	Section 3.5
BOD.1	BOD reset not functional	'-'	Section 3.6
UART.1	Coinciding VPB read and hardware register update	'-', 'A', 'B'	Section 3.7
MAM.1	Incorrect read of data from SRAM	'-', 'A', 'B'	Section 3.8
MAM.2	Code execution failure can occur with MAM Mode 2	'-', 'A', 'B', 'C', 'D'	Section 3.9
SPI.1	Incorrect shifting of data in slave mode at lower frequencies	'-', 'A', 'B'	Section 3.10
SSP.1	Initial data bits/clocks corrupted in SSP transmission	'-', 'A', 'B', 'C', 'D'	Section 3.11
DC/DC.1	DC/DC Converter start-up issue	'-', 'A', 'B', 'C'	Section 3.12
WDT.1	Accessing non-Watchdog APB registers in the middle of the feed sequence causes a reset	'-', 'A', 'B', 'C', 'D'	Section 3.13
ADC.1	External sync inputs not operational	'-', 'A', 'B', 'C', 'D'	Section 3.14

Table 3. AC/DC deviations table

AC/DC deviations	Short description	Revision identifier	Detailed description
ESD.1	ESD-HBM Stress issue	'-', 'A', 'B'	Section 4.1
ESD.2	ESD weakness on RTCX1 pin	'-', 'A', 'B', 'C', 'D'	Section 4.2

Table 4. Errata notes table

Errata notes	Short description	Revision identifier	Detailed description
Note 1	Increased power consumption from battery while RTC is running from the main 3.3V supply	'-', 'A', 'B', 'C', 'D'	Section 5.1
Note 2	Port pin P0.31 must not be driven low during reset.	'-', 'A', 'B', 'C', 'D'	Section 5.2
Note 3	ADC1 is not functional on the LPC2134 which has datecodes up to 0548.	'-', 'A', 'B', 'C'	Section 5.3
Note 4	When the input voltage is Vi ≥ Vdd I/O + 0.5 v on port pin P0.25 (configured as general purpose input pin), current must be limited to less than 4 mA by using a series limiting resistor.	'-', 'A', 'B', 'C', 'D'	Section 5.4

3. Functional problems detail

3.1 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.

Workaround:

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 would have to be done manually.

3.2 Timer.1: Missed interrupt potential

Introduction:

The Timers may be configured so that events such as Match and Capture, cause interrupts. Bits in the Interrupt Register (IR) indicate the source of the interrupt, whether from Capture or Match.

Problem:

If more than one interrupt for multiple Match events using the same Timer are enabled, it is possible that one of the match interrupts may not be recognized. If this occurs no more interrupts from that specific match register will be recognized. This could happen in a scenario where the match events are very close to each other. This issue also affects the Capture functionality.

Specific details:

Suppose that two match events are very close to each other (Say Match0 and Match1). Also assume that the Match0 event occurs first. When the Match0 interrupt occurs the 0th bit of the Interrupt Register will be set. To exit the Interrupt Service Routine of Match0, this bit has to be cleared in the Interrupt Register. The clearing of this bit might be done by using the following statement:

T0_IR = 0x1;

It is possible that software will be writing a 1 to bit 0 of the Interrupt Register while a Match1 event occurs, meaning that hardware needs to set the bit 1 of the Interrupt Register. In this case, since hardware is accessing the register at the same time as software, bit 1 for Match1 never gets set, causing the interrupt to be missed.

In summary, while software is writing to the Interrupt Register, any Match or Capture event (which are configured to interrupt the core) occurring at the same time may result in the subsequent interrupt not being recognized.

Similarly for the Capture event, if a capture event occurs while a Match event is being is serviced then the Capture event might be missed if the software and hardware accesses coincide.

Affected features:

1. Interrupt on Match for Timer0/1.
2. Interrupt on Capture for Timer0/1.
3. These same features will be affected when using PWM.

Workaround:

There is no clear workaround for this problem but some of the below mentioned solutions could work with some applications.

Possible workarounds for Match functionality:

1. If the application only needs two Match registers then distribute them between Timer 0 and Timer 1 to avoid this problem.
2. Stop the timer before accessing the Interrupt register for clearing the interrupt and then start timer again after the access is completed.

3. Polling for interrupt: Supposing that there are two Match events (Match X and Match Y). At the end of the Interrupt Service Routine (ISR) for Match X, compare the Timer Counter value with the Match Register Y value. If the Timer Counter value is more than the Match Register Y value then it is possible that this event might have been missed. In this case jump to the ISR directly and service Match event Y.

Possible workarounds for Capture functionality:

1. Try to spread the capture events between both timers if there are two capture events. If the application also has a match event then one of the capture events may suffer.
2. Polling for Capture: At the end of a Match interrupt ISR or Capture event ISR compare the previous Capture value with the current Capture value. If the Capture value has changed then the Capture event might have been missed. In this case, jump to the ISR directly and service the Capture event.

3.3 PWM.1: Missed Interrupt Potential for the Match functionality. The description is same as above.

3.4 Timer.2: 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.

Workaround:

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:

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

3.5 Power Down.1: Restricted Vdd while device is in power down mode

Introduction:

This device has two reduced power modes: idle mode and power down mode. In power down mode, the oscillator is shut down and the chip receives no internal clocks.

Problem:

If Vdd is below the specified value, a reset could occur, waking up the device and the saved values in SRAM will be re initialized according to a power-up procedure.

Workaround:

The two possible workarounds would be:

1. Avoid operating in power down mode.
2. If the device is operated in power down mode then Vdd should be 3.3V +10%/-5%.

3.6 Brown-out Detection(BOD).1: BOD reset does not get triggered when the voltage of Vdd falls below 2.6 V

Introduction:

The BOD monitors the Vdd in two stages. If the voltage falls below 2.9 V, the BOD asserts an interrupt signal to the Vectored Interrupt Controller. The second stage of the low voltage detection asserts Reset to inactivate the device when Vdd falls below 2.6 V.

Problem:

BOD reset does not get triggered when the voltage of Vdd falls below 2.6 V.

Workaround:

None.

3.7 UART.1: Coinciding VPB read and hardware register update

Introduction:

Reading the contents of the IIR,LSR and MSR registers will clear certain bits in the register.

1. Reading the IIR should clear the THRE status if THRE is the highest priority pending interrupt (only affects UART1).
2. Reading LSR should clear the OE/PE/FE/BI bits (affects both UART0 and UART1).
3. Reading MSR should clear the Delta DCD/Trailing Edge RI/Delta DSR/Delta CTS bits (Only affects UART1).

Problem:

If hardware is setting one of these above bits while the software is reading the contents of the register the reading process clears all bits in the register including the bit that got set by hardware. The software reads the old value though and the bit that got set by hardware is lost.

Specific details:

Suppose IIR has a modem status interrupt while the other interrupts are inactive and software reads the IIR value (polling) while hardware sets the THRE interrupt then software will read the Modem Interrupt value while the THRE interrupt is cleared i.e the THRE interrupt is lost.

Suppose the LSR is all zeros and software is reading the register while hardware is generating a parity error then the parity error bit is cleared while the software reads the old value (all zeros) i.e. the parity error is lost.

Suppose MSR is all zeros and software is polling the value of the register while the value of CTS is changing then the change in CTS value should result in the Delta CTS bit getting set. Instead software will read all zeros and the Delta CTS bit in the MSR register will be cleared i.e. the Delta CTS status is lost.

Workaround:

IIR reading:

The IIR bug can be worked around by disabling the modem status interrupt effectively making THRE the lowest priority interrupt. The work-around does not work in software interrupt polling mode. Modem status has to be handled by software polling MSR.

Now there are two cases:

1. A THRE interrupt is pending, software responds to the interrupt by reading the IIR while another, higher priority interrupt is set (e.g. RDA). In this case software will read the THRE status although the status will not be cleared where it should have been. After handling the THRE and RDA interrupt another dummy THRE interrupt may occur, unless in the meantime software has filled THR. This is considered an error although not fatal.

2. A high priority interrupt is pending, software responds to the interrupt by reading the IIR register while a THRE interrupt is set. In this case, software will read the higher priority interrupt and the THRE interrupt will be handled later. This behaviour is as expected.

LSR reading:

A work-around for this problem is to service the OE/PE/FE/BI condition before another character is received which will trigger an LSR update. So basically, service the interrupt in one-character time.

MSR reading:

The MSR bug can be worked-around by not using the Delta DCD/Trailing Edge RI/Delta DSR/Delta CTS bits in the MSR but instead use the DCD/TRI/DSR/CTS bits in the same register. To prevent, a transition from being missed software should poll the register's value at a sufficiently high rate.

3.8 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

Workaround:

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

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

MAMTIM should be written before MAMCR.

3.9 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.

3.10 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:

1. SPI is configured as a slave (with CPHA=0).
2. 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.

Workaround:

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.

3.11 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 30 MHz and it is 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 30 MHz, the first pulse can be expected to be approximately 10 ns shorter and the second pulse around 5 ns shorter. The remaining two pulses are around 2 ns shorter than subsequent pulses.

At 25 MHz, the length of the first pulse would be around 7 ns shorter. The subsequent three pulses are around 2 ns shorter.

At 20 MHz only the first pulse is affected and it is around 2 ns shorter. All subsequent pulses are fine.

The deviation of the initial data bits/clocks will decrease as the SSP frequency decreases.

Workaround:

None.

3.12 DC/DC.1: DC/DC converter start-up issue

Introduction:

The device operating voltage range is 3.0 V to 3.6 V and it has an internal DC/DC converter that provides 1.8 V to the ARM7 Core.

Problem:

If during a power-on reset the voltage on Vdd takes longer than 200 ms to ramp from below 0.8 V to above 2.0 V, the chip-internal DC/DC converter might not start up correctly. If this happens, the crystal oscillator will not be running, resulting in no code execution. As an example, having a Vdd rise time of less than 10 V/s might trigger this problem.

The same problem might occur during a supply voltage drop during which Vdd remains between 300 mV and 80 mV for more than 200 ms before going back to the specified Vdd level. As an example, having a residue battery voltage of less than 0.3 V but more than 0.08 V in a rechargeable battery application might trigger this problem when the charger providing the 3 V supply is being connected.

Workaround:

Apply another power-on Reset during which Vdd rises from below 0.8 V to above 2.0 V in less than 200 ms.

3.13 WDT.1: Accessing non-Watchdog APB registers during the feed sequence causes a reset

Introduction:

The Watchdog timer can reset the microcontroller within a reasonable amount of time if it enters an erroneous state.

Problem:

After writing 0xAA to WDFEED, any APB register access other than writing 0x55 to WDFEED may cause an immediate reset.

Workaround:

Avoid APB accesses in the middle of the feed sequence. This implies that interrupts and the GPDMA should be disabled while feeding the Watchdog.

3.14 ADC.1: External sync inputs not operational

Introduction:

In software-controlled mode (BURST bit is 0), the 10-bit ADCs can start conversion by using the following options in the A/D Control Register:

26:24	START		When the BURST bit is 0, these bits control whether and when an A/D conversion is started:	0
		000	No start (this value should be used when clearing PDN to 0).	
		001	Start conversion now.	
		010	Start conversion when the edge selected by bit 27 occurs on P0.16/EINT0/MAT0.2/CAP0.2 pin.	
		011	Start conversion when the edge selected by bit 27 occurs on P0.22/AD1.7/CAP0.0/MAT0.0 pin.	
		100	Start conversion when the edge selected by bit 27 occurs on MAT0.1.	
		101	Start conversion when the edge selected by bit 27 occurs on MAT0.3.	
		110	Start conversion when the edge selected by bit 27 occurs on MAT1.0.	
		111	Start conversion when the edge selected by bit 27 occurs on MAT1.1.	

Fig 1. A/D control register options

Problem:

The external start conversion feature, ADxCR:START = 0x2 or 0x3, may not work reliably and ADC external trigger edges on P0.16 or P0.22 may be missed. The occurrence of this problem is peripheral clock (pclk) dependent. The probability of error (missing an ADC trigger from GPIO) is estimated as follows:

- For PCLK_ADC = 60 MHz, probability error = 12 %
- For PCLK_ADC = 50 MHz, probability error = 6 %
- For PCLK_ADC = 12 MHz, probability error = 1.5 %

The probability of error is not affected by the frequency of ADC start conversion edges.

Workaround:

In software-controlled mode (BURST bit is 0), the START conversion options (bits 26:24 set to 0x1 or 0x4 or 0x5 or 0x6 or 0x7) can be used. The user can also start a conversion by connecting an external trigger signal to a capture input pin (CAPx) from a Timer peripheral to generate an interrupt. The timer interrupt routine can then start the ADC conversion by setting the START bits (26:24) to 0x1. The trigger can also be generated from a timer match register.

4. AC/DC deviations detail

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

Philips Quality Spec specifies ESD-HBM should be above 2.0 kV. The LPC2134 passed ESD-HBM Stress up to 2.5 kV but without VDD-to-VDD zapping (e.g. Vref to VBat, or Vref to V3A). Units zapped according to JEDEC Standard (JESDA22-A114-B) did not pass the ESD Post Stress Test.

From Revision C onwards, this issue has been fixed and the ESD-HBM limit has been improved (except ESD.2 as described below).

The LPC2134 ESD-HBM is now 4.0 kV.

4.2 ESD.2: The device does not meet the 2 kV ESD requirements on the RTCX1 pin

Introduction:

The LPC2134 is rated for 2 kV ESD. The RTCX1 pin is the input pin for the RTC oscillator circuit.

Problem:

The LPC2134 does not meet the required 2 kV ESD specified.

Workaround:

Observe proper ESD handling precautions for the RTCX1 pin.

5. Errata notes detail

5.1 Note.1: Increased power consumption from battery while RTC is running from the main 3.3 V supply

Introduction:

The RTC is powered by its own power supply pin, Vbat, which can be connected to a battery or to the same 3.3 V supply used by the rest of the device.

Problem:

If the VBAT is connected to an external battery, RTC will consume more power from the battery if the core is running and the selected clock source is the prescaler.

Workaround:

Switch the clock source such that the RTC takes the clock from the 32 KHz oscillator that is connected to the RTCX1 and RTCX2 pins. After initialization of the RTC, clear the PCRTC bit in the PCONP register to switch off the peripheral clock (pclk) to RTC. Any further writes to the RTC would require this bit to be set.This will reduce the power consumed from VBAT.

5.2 Note.2

Port pin P0.31 must not be driven low during reset. If low on reset the device behaviour is undetermined.

5.3 Note.3

ADC1 is not functional on the LPC2134 which has datecodes up to 0548.

5.4 Note.4

On port pin P0.25 (when configured as general purpose input pin), leakage current increases when the input voltage is Vi ≥ Vdd I/O + 0.5 V. Care must be taken to limit the current to less than 4 mA by using a series limiting resistor.

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

6.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

7. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 1 March 2011
Document identifier: ES_LPC2134