AN11498

NXP LPC2000 IEC60335 Class B library

Rev. 1 — 20 January 2014

Application note

Document information

Info	Content
Keywords	IEC60335 Class B, ARM7, NXP LPC2000, LPC213x
Abstract	This application note describes the IEC60335 Class B certified library for the NXP LPC2000 family members. All tests implemented and the library usages are described in detail.

Revision history

Rev	Date	Description
1	20140120	Initial version.

Contact information

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

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

1. Introduction

Modern day home appliances require a certain level of protection in order to avoid hazardous situations if the appliance fails. Since 2007, home appliances must comply with the IEC60335 standard. Home appliance manufacturers therefore need to ensure that the requirements are met.

This document describes the IEC60335 standard requirements with respect to software for microcontrollers and the implementation of these requirements. NXP has developed a software library for the NXP LPC2000 family based on these requirements; this document discusses the tests and the usage of these tests in detail.

ATTENTION!

The usage of this library does not make a certified application of your project. It is still necessary to have the complete application software certified.

This library should not be changed, and it should be used as explained. Otherwise, a new certification for the changed parts will be necessary.

1.1 How to read this application note

This application note is a guide in using and implementing the library functions provided. It will first discuss the set requirements of the IEC60335 standard, and then briefly discuss the products the library is developed for.

The main part of the document describes how the Class B tests are done and how it can and should be implemented. Details on the tested peripherals are given in the last chapter.

2. IEC60335 Class B

The IEC60335 standard specifies design enhancements for home appliance manufacturers that design appliances with electronic controls and controls using software with respect to safe and reliable operation. This standard requires inclusion of features that will avoid or at least minimize the change of hazardous situations when the appliance fails.

Referring to IEC60730, this deals with standard various assets of safety and reliability precautions required to be taken for all home appliances. Annex H of the IEC60730 standard software and hardware requirements is defined to be taken in order to comply with this standard.

2.1 Software classification

Within the IEC60730 Annex H, details for testing and diagnostic implementation in microcontroller software are classified as A, B or C.

- Class **A**: Control functions which are not intended to be relied upon for the safety of the equipment
- Class **B**: Control functions intended to prevent unsafe operation of the controlled equipment.
- Class **C**: Control functions which are intended to prevent special hazards (e.g., explosion of the controlled equipment, such as burner controls).

The majority of the home appliances like white goods (refrigerator, dishwasher, cooker etc.) and personal appliances (electrical tooth brush, shaver etc.) require the Class B level of precautions.

The IEC60370 Class B specifies that measures must be taken to avoid software related faults and errors in data and segments of the software that are safety related. Periodic monitoring of the system therefore is required.

2.2 Class B components

Table H.11.12.7 of IEC60730 Annex H specifies the components to be tested and monitored during operation of the controller. Table 1 shows a summary of table H.11.12.7.

Table 1. IEC60335 Class B tests as defined by IEC60730 Annex H

Test number	Component	Fault/error	In library
1.1	CPU registers	Stuck at	YES
1.3.	Program Counter	Stuck at	YES
2.	Interrupt handling and execution	No interrupt or too frequent	YES
3.	Clock	Wrong frequency (for quartz synchronized clock: harmonics/subharmonics only)	YES
4.1	Invariable memory	All single bit faults	YES
4.2.	Variable memory	DC fault	YES
4.3.	Addressing (relevant to variable and invariable memory)	Stuck at	YES
5.1.	Internal data path	Stuck at	NO
5.2.	Addressing	Wrong address	NO
6.	External communications	Hamming distance 3	NO
6.3.	Timing	Wrong point in time and sequence	NO
7.[1]	Input/output periphery	Fault conditions specified in H.27	NO
7.2.1. [1]	A/D and D/A converters	Fault conditions specified in H.27	NO
7.2.2. [1]	Analog multiplexer	Wrong addressing	NO

[1] Only when using external memory

3. NXP ARM7 microcontrollers

This chapter gives a general description of the NXP ARM7 family members for which the IEC60335 Class B self-test libraries are written.

3.1 The NXP ARM7 microcontrollers

The LPC213x and LPC213x/01 microcontrollers are based on a 16/32 bit RM7TDMI-S CPU with real-time emulation and embedded trace support that combines the microcontroller with embedded high speed flash memory ranging from 32 kB to 512 kB and 8/16/32 kB of on-chip static RAM. CPU can run up to 60 MHZ.

3.1.1 The ARM7TDMI-S core

The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed complex instruction set computers. This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective processor core.

Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory.

The ARM7TDMI-S processor also employs a unique architectural strategy known as THUMB, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue.

4. IEC60335 Class B library

The chapter gives an overview about the functionality of the various functions. It gives you knowledge about the library and helps with understanding the self-test philosophy. Please note by changing any library functionality it needs to be re-certified again. If a special part needs to be modified, then there will be an explicit description and explanation.

4.1 POST and BIST

POST (Pre Operation System Test) means the testing as part of the start-up procedure. These tests are corruptible, which means that the data contents are not restored after executing the test. Also, in this state of application, there are normally no interrupts active.

Note, at start-up all tests must be executed: CPU registers, PC, RAM and ROM. For this reason special POST functions are available. The POST testing is developed such that it reduces test time and therefore is monolithic and corruptible.

The Built-In Self-Test (BIST) is designed such that it will not modify the content of program, data or registers. To avoid system failures in time critical applications, these test are not monolithic. Functions are implemented for testing the variable and non-variable in smaller blocks.

4.2 CPU register test (1.1)

The ARM7 core has a total of 37 registers with 32 bits wide including 31 general-purpose and six status registers which are arranged in partially overlapping banks, with a different register bank for each processor mode. The general-purpose registers R0-R15 can be split into three groups:

- The unbanked registers, R0-R7
- The banked registers, R8-R14
- Register 15, the PC

At any time, 15 general-purpose registers (R0 to R14), one or two status registers and the program counter are visible. So these visible registers (R0 to R14 and CPSR) are tested in privileged mode for stuck-at faults and direct coupling faults.

These tests are to be executed as POST and BIST.

POST testing is a corruptible test, so the CPU registers will not be retained. Since the POST CPU register tests don’t retain register data, it is mandatory to execute this test prior to any other application or system initialization. Preferably execute this test prior to branch to main.

CPU BIST testing isn’t corruptible, so all data is restored after testing. To decrease test time and therefore CPU resources, the CPU register BIST testing is divided into five separate tests. The first three test the general purpose registers (R0-R7, R4-R10 and R8-R12), the fourth tests the stack pointer (R13). To prevent the system from crashing, all interrupts and exceptions are disabled while running this part of the CPU register BIST. The fifth and last BIST test is testing the other special registers (R14 and CPSR).

The CPU register POST and BIST tests are developed in assembly code because most of the registers of the core are not directly accessible from C code. However, it allows C code to call the assembly source routines.

Both BIST and POST use the same test methodology when testing the registers. First, a pattern will be stored in the register, then read back and compared. Then, the inverse of that pattern is stored in the register, read and compared. If any register test can’t pass, a status flag in a defined test structure will be set to fail and exit from testing at once. Otherwise the flag will be set to pass when the tests are completed.

The test structures:

- `CPUregTestPOST_struct`
- `CPUregTestBIST_struct`

The basic pattern used for the CPU register tests:

- Normal: `0xAAAA.AAAA`
- Inverted: `0x5555.55555`

4.3 Program Counter (PC) test (1.3)

The PC test checks whether the PC is able to branch throughout the whole program and data memory space. To test the branching, dummy functions are allocated throughout the whole used program and data memory space.

The allocation of the PC dummy test functions are placed accordingly by use of sections defined in the linker file.

The PC test routines call the dummy functions and check the returned value. Each dummy function will return a unique value. Thereby it is possible to check if the PC has jumped to the correct address.

Note that an enabled memory protection unit may trigger an exception when dummy functions are executable code areas that are protected.

In principle, the test results always show as okay, because a defective program counter results in program crashes in most cases.

There are two different implementations for this test available: BIST and POST. The POST will check each dummy function at once. This is implemented by a loop. The BIST will only test one dummy function per call. All functions will be called after each other like a ring buffer. The implementation flowcharts are shown in Fig 1:

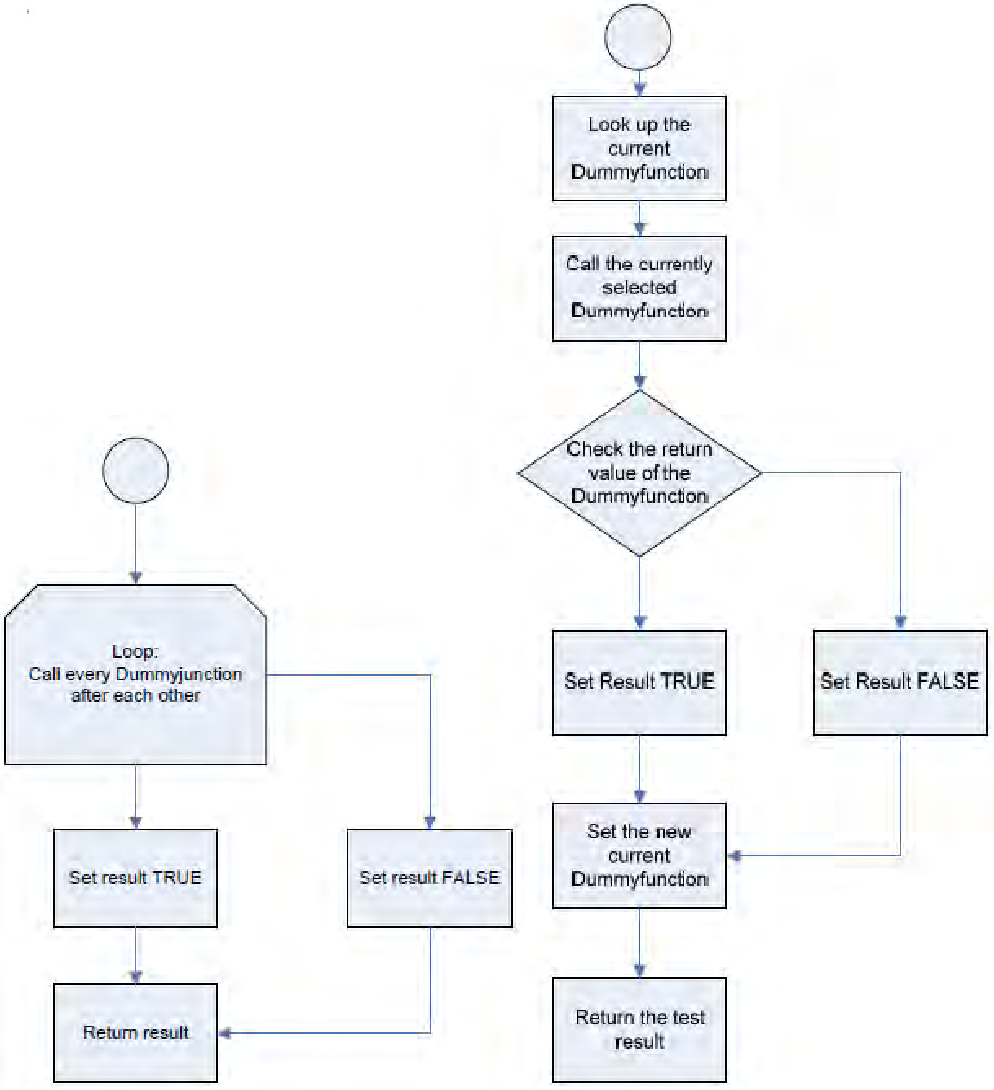

Look up the current Dummyfunction
Call the currently selected Dummyfunction
Check the return value of the Dummyfunction
Set Result TRUE
Set Result FALSE
Set the new current Dummyfunction
Return the test result
Loop:
Call every Dummyjunction after each other
Set result TRUE
Set result FALSE
Return result

Fig 1. **PC POST (left) and PC BIST (right) test flow diagram**

4.4 Interrupt handling and execution test (2)

The test for interrupt handling and execution is application dependent. In this test, the library delivers some templates to enable the users testing the functionality in an abstract way.

The interrupts will be checked with the aid of counter variables. The different interrupts, which are observed by counter mechanisms, should have individual up-counting values instead of simply adding one.

To check the interrupts, the counter value has to be checked cyclically in a known equidistant time and compared to boundaries estimated by the user. A timer interrupt service handler should assist with this.

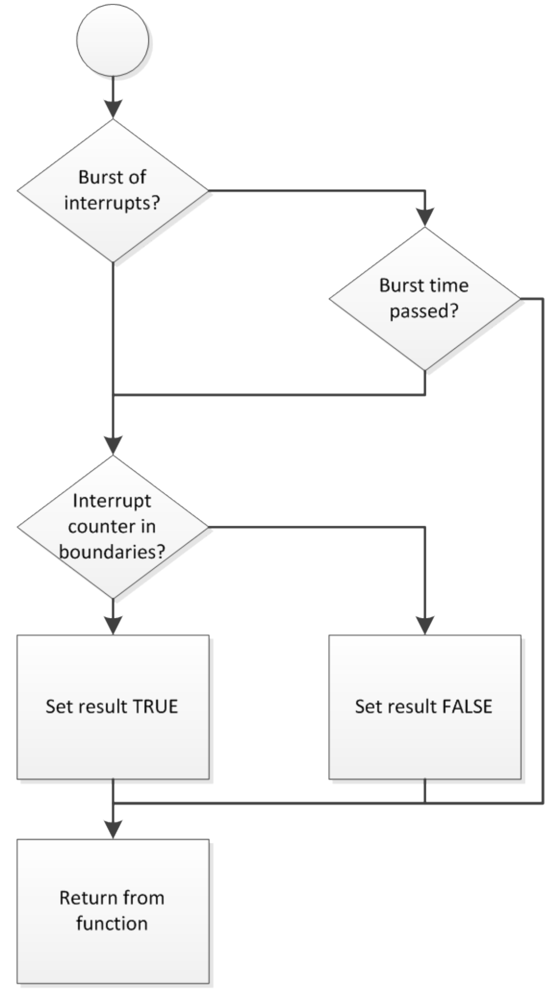


Fig 2. Interrupt

If the interrupt that needs to be checked is a burst of interrupts, the routine will check if the time to wait for all interrupts has elapsed. If the time has passed, it will check the interrupt count to be within the boundaries. If not, the check function will return directly, without setting any Result.

If the interrupt to check is not a burst of interrupts, the routine will check the interrupt counter to be within the bound directly.

4.5 Clock system test (3)

This test is intended to check the CPU clock source and frequency. This requires a second independent clock source. For a part of the NXP LPC2000 family, the only possibility to get interrupts triggered, sourced by an independent clock, is to use the RTC peripheral.

Three test functions are implemented, and the first one is cyclically called from the main loop of the user application.

The main loop function checks both the timer and RTC interrupt occurrence functions. If one or both of them are missing within a rough time frame, which has to be estimated empirically, the function will return failed as result. This function also checks the result of the timer check, which is performed by the RTC function.

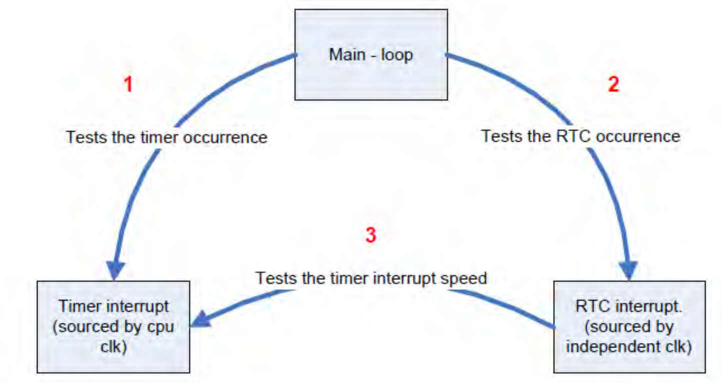

Main - loop
1
Tests the timer occurrence
2
Tests the RTC occurrence
3
Tests the timer interrupt speed
Timer interrupt (sourced by cpu clk)
RTC interrupt. (sourced by independent clk)

Fig 3. Clock system test flow

The second function is intended to be called from a timer interrupt service handler. This Timer needs to have the same clock source as the CPU.

The timer simply counts how often the timer interrupt has occurred. This value is then checked by the RTC function. Additionally, it sets the occurrence semaphore, which is used for occurrence recognition inside of the main function.

The last function is intended to be called from the RTC interrupt service handler. This function is intended to check the frequency of the timer interrupts.

The RTC function also sets an occurrence semaphore, to be tested from the main function. Then it checks the timer counter variable to be within the estimated boundaries.

The result of this check is stored into a result semaphore.

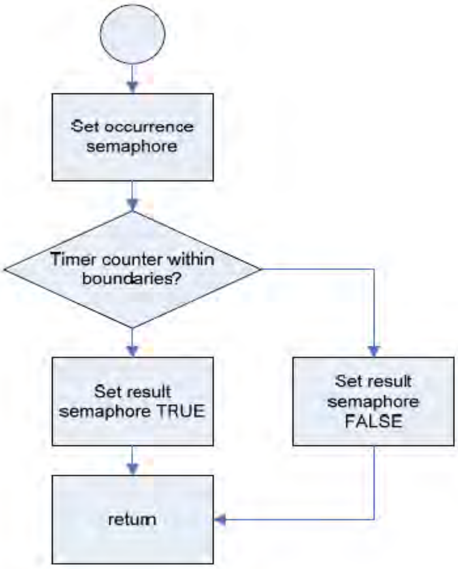

Set occurrence semaphore
Timer counter within boundaries?
Set result semaphore TRUE
Set result semaphore FALSE
return

Fig 4. Setting semaphore and boundary check

4.6 Invariable memory test (4.1)

4.6.1 Test description

The invariable memory must be checked for single bit faults. During POST testing the complete flash memory is tested. During BIST testing it is advisable to test the flash memory in smaller segments to prevent the CPU from being blocked.

4.6.1.1 Multiple input signature register

The NXP LPC2000 family has an integrated flash module that incorporates a 128-bit signature generator which is reserved. This generator can be used for generating a signature of the used safety critical memory region.

Since this module is integrated in the flash module, it generates a signature faster than when implemented in software.

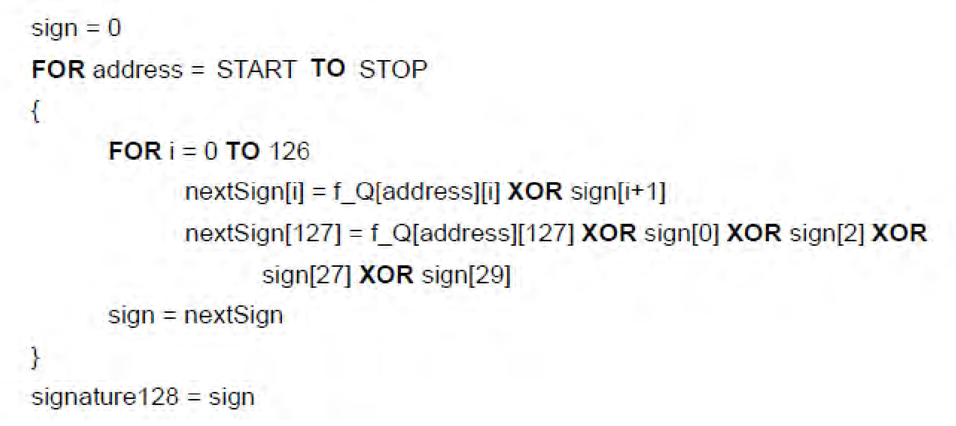
A signature can be generated for any part of the flash contents. The address range to be used for the signature generation is defined by writing the start address and the stop address to the respective registers.

The flash address should first be aligned with a flash word (128 bits). In the array this is done by right-shifting the start and stop address by 4.

Since the MISR is implemented in hardware, it is much faster than doing the same MISR check in software. The time that the signature generation takes is proportional to the address range for which the signature is generated.

4.6.1.2 Signature verification

The signatures generated by the hardware must be verified and be equal to the reference signatures which are generated by software. The algorithm to derive the reference signature is shown in Fig 5:



```
sign = 0
FOR address = START TO STOP
{
        FOR i = 0 TO 126
                nextSign[i] = f_Q[address][i] XOR sign[i+1]
                nextSign[127] = f_Q[address][127] XOR sign[0] XOR sign[2] XOR
                        sign[27] XOR sign[29]
        sign = nextSign
}
signature128 = sign
```

Fig 5. Algorithm for generating a 128-bit long digital signature

Important notification:

The hardware signature generator is *blocking* for the flash; this means no flash read or write access is possible during signature generation. It is therefore advisable to make sure while using the hardware signature generator the flash will not be accessed.

4.7 Variable memory (4.2)

The variable memory (SRAM) must be tested for direct coupling and stuck-at faults. A pattern therefore must be written and checked. This pattern is chosen such that it could determine not only stuck-at faults but also direct coupling and even retention faults.

The March test algorithm is developed for efficient testing and detecting direct coupling and stuck-at faults in the variable memory, or in this case RAM, array.

The algorithm can be divided in 8 individual tests, called March tests 0 to 8. Each test has an *even* and an *odd* test.

Even represents even addressing and odd represents odd addressing during the test, indicated as nnnn in Fig 6. Increasing and decreasing addressing is indicated by use of an up pointing or down pointing arrow. Read or write execution is indicated by r or w.

There are two patterns used during the variable memory test, the dbg (0x5555.5555) and the dbgN (0xAAAA.AAAA) pattern. The pattern layout depends on the invariable memory structure.

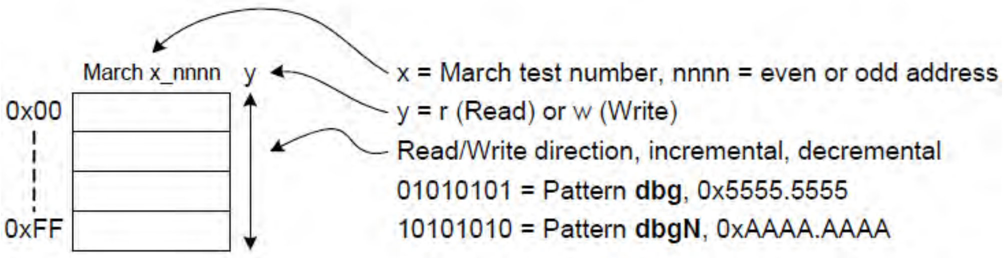

March x_nnnn
y
0x00
0xFF
x = March test number, nnnn = even or odd address
y = r (Read) or w (Write)
Read/Write direction, incremental, decremental
01010101 = Pattern **dbg**, 0x5555.5555
10101010 = Pattern **dbgN**, 0xAAAA.AAAA

Fig 6. Algorithm flow-diagram test example

This algorithm is designed to cover both stuck-at faults and direct coupling faults in the fastest possible way.

March tests 0 and 1 test in increasing addressing order whether the full tested variable memory region dbg pattern is written and read correctly. This covers stuck-at 0 faults at the even bits and stuck-at 1 faults at the odd bits in the data words.

March test 2 tests in decreasing addressing order the stuck-at 0 faults at the odd bits and the stuck-at 1 faults in the even bits. It also tests the retention of the charged cells when loaded with the dbg pattern. It also takes the direct coupling in account.

March tests 3 and 4 test in increasing order the inversion of March tests 0 and 1, where March test 5 does the same for test 2.

March tests 6, 7 and 8 are testing in increasing and decreasing addressing order the direct coupling more extensively.

5. Test usage

This section describes the test examples to show how to test the IEC components on a target board using the present IEC library. The test examples are implemented in the following development environment:

- MCB2130 board based on LPC2138
- ULink2 debugger
- KEIL IDE V4.7

The test examples can be active or inactive by setting the macro definition to 1 or 0 in test_config.h file. If the test is passed, the corresponding LED on board will be toggled. Otherwise, the LED will keep the state. The maps for definitions, test components and LEDs are shown in Table 2:

Table 2. Test example maps

Definition	Test component	LED port on board
CPU_REG_TEST	CPU registers	P1_16
PC_TEST	Program Counter	P1_17
INT_TEST	Interrupt handling and execution	P1_20
CLK_TEST	Clock	P1_21
FLASH_TEST	Invariable memory	P1_19
RAM_TEST	Variable memory	P1_18
INTEGRATED_TEST	All the above components except Clock&Interrupt	P1_16 & P1_17 & P1_18 & P1_19

Note: the interrupt and clock test can only be tested individually in the project.

How to test the examples:

- Open the project file named “IEC60335_B_example.Uv2” in the software package.
- Open the configuration file “test_config.h” and select the test components by setting the macro definition to 1.
- Build the project and download the output file to MCB2130 board through ULink2.
- Reset the MCB2130 board to run the test example
- The test is passed when the corresponding LED is toggled. Otherwise, it’s failed.

6. Conclusion

This application note describes the IEC60335 Class B certified library for the NXP LPC2000 family members.

This application note also describes the test examples for all components using the library.

7. References

[1] NXP LPC213xx User Manual UM10120, NXP Semiconductors

[2] ARM Architecture Reference Manual, DDI0100E_ARM_ARM, 2000

[3] AN10918, NXP LPC Cortex-M3 IEC60335 Class B library, Rev. 01 — 1 March 2010

8. Legal information

8.1 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.

8.2 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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

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 and its suppliers accept 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 competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

8.3 Trademarks

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

9. List of figures

10. List of tables

11. Contents

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

Date of release: 20 January 2014
Document identifier: AN11498