

**Core self-test library compliant with IEC 60730, IEC 60335 UL 60730,  
UL 1998 documentation**

**Library architecture document**

## Document revision history

Date	Author	Version	Notes
11/2015	Jozef Sedlak	0.1	Initial release
11/2015	Jozef Sedlak	1.0	Version for certification
11/2015	Jozef Sedlak	1.1	Correction of variable memory test
10/2016	Jozef Sedlak	1.2	NXP, MKE1xF added
11/2018	Jozef Sedlak	3.0	Keil, MCUXpresso support, new devices: MK32W0x, MIMXRT

<b>1</b>	<b>CORE SELF-TEST LIBRARY ARCHITECTURE .....</b>	<b>4</b>
<b>2</b>	<b>CORE SELF-TEST LIBRARY – SOURCE CODE VERSION .....</b>	<b>5</b>
<b>3</b>	<b>CORE SELF-TEST LIBRARY – OBJECT CODE VERSION .....</b>	<b>9</b>
<b>4</b>	<b>LIBRARY STRUCTURE VALIDATION.....</b>	<b>9</b>

# 1 Core self-test library architecture

---

The intention of core self-test library is to provide functions performing microcontroller core self-test. Library consists of independent functions performing tests compliant with international standards (IEC 60730, IEC 60335 UL 60730, UL 1998). Library supports IAR, Keil and MCUXpresso IDEs. NXP core self-test library performs the following tests:

- CPU Registers test
- CPU Program Counter test
- Variable Memory test
- Invariable Memory test
- Stack test
- Clock test
- Digital Input/Output test
- Analog Input/Output test
- Watchdog test

Test architecture, implementation, test and validation of corresponding tests is comprehensively described in the independent documents for each test. The library can be distributed in both versions: source code and object code.

The library supports MKV3x, MKV4x, MKV5x, MKE1xF, MK32Wx/cm4, MIMXRT families based on ARM CM4 and CM7 core.

Core self-test library has two versions: source code version and object code version. Three object codes – one for each IDE - were compiled from the common source code version. Header files are the same for both versions.

## 2 Core self-test library – source code version

---

The library name is IEC60730\_Kinetis\_M4\_M7\_Class\_B. Main header file is IEC60730\_B\_CM4\_CM7.h.

The list of library files:

1. Registers test:  
The source code is implemented in assembler. The implementation of functions can be found in “IEC60730\_B\_CM4\_CM7\_reg.S” and functions declaration, necessary macros and data types can be found in corresponding header file “IEC60730\_B\_CM4\_CM7\_reg.h”. Test functions for FPU registers are placed in separated file IEC60730\_B\_CM4\_CM7\_reg\_fpu.S
2. Invariable memory (Flash) test:  
The source code is implemented in assembler. The implementation of functions can be found in “IEC60730\_B\_CM4\_CM7\_flash.S” and functions declaration, necessary macros and data types can be found in corresponding header file “IEC60730\_B\_CM4\_CM7\_flash.h”. Additional file with test using the DCP unit is IEC60730\_B\_CM4\_CM7\_flash.c
3. Variable memory (RAM) test:  
The source code is implemented in assembler. The implementation of functions can be found in “IEC60730\_B\_CM4\_CM7\_ram.S” and functions declaration, necessary macros and data types can be found in corresponding header file “IEC60730\_B\_CM4\_CM7\_ram.h”.
4. Stack test:  
The source code is implemented in assembler. The implementation of functions can be found in “IEC60730\_B\_CM4\_CM7\_Stack.S” and functions declaration, necessary macros and data types can be found in corresponding header file “IEC60730\_B\_CM4\_CM7\_Stack.h”.
5. Program counter test:  
The source code is implemented in assembler. The implementation of functions can be found in “IEC60730\_B\_CM4\_CM7\_pc.S” and functions declaration, necessary macros and data types can be found in corresponding header file “IEC60730\_B\_CM4\_CM7\_pc.h”. For new test version, there is one function in additional file IEC60730\_B\_CM4\_CM7\_pc\_object.S.
6. Digital input/output test:  
The source code is implemented in c language. The implementation of functions can be found in “IEC60730\_B\_CM4\_CM7\_dio.c” and functions declaration, necessary

macros and data types can be found in corresponding header file "IEC60730\_B\_CM4\_CM7\_dio.h".

7. Extended tests for Digital Input and Output are placed in IEC60730\_B\_CM4\_CM7\_dio\_ext.c and IEC60730\_B\_CM4\_CM7\_dio\_ext.h files
  8. Analog input/output test:  
The source code is implemented in c language. The implementation of functions can be found in "IEC60730\_B\_CM4\_CM7\_aio.c" and functions declaration, necessary macros and data types can be found in corresponding header file "IEC60730\_B\_CM4\_CM7\_aio.h".
  9. Clock test:  
The source code is implemented in c language. The implementation of functions can be found in "IEC60730\_B\_CM4\_CM7\_clock.c" and functions declaration, necessary macros and data types can be found in corresponding header file "IEC60730\_B\_CM4\_CM7\_clock.h".
  10. Watchdog test:  
The source code is implemented in c language. The implementation of functions can be found in "IEC60730\_B\_CM4\_CM7\_wdg.c" and functions declaration, necessary macros and data types can be found in corresponding header file "IEC60730\_B\_CM4\_CM7\_wdg.h".
- File needed because of assembler directives: asm\_mac\_common.h.
  -

**Table 1. List of library items**

File Name	Test Type	Function Name
IEC60730_B_CM4_CM7_reg.S	Register test	IEC60730B_CM4_CM7_CPU_RegisterTest()
	Register test	IEC60730B_CM4_CM7_CPU_NonStackedRegisterTest()
	Register test	IEC60730B_CM4_CM7_CPU_PrimaskTest()
	Register test	IEC60730B_CM4_CM7_CPU_SPmainTest()
	Register test	IEC60730B_CM4_CM7_CPU_SPprocessTest()
	Register test	IEC60730B_CM4_CM7_CPU_ControlTest()
	Register test	IEC60730B_CM4_CM7_CPU_SpecialTest()
	Register test	IEC60730B_CM4_CM7_CPU_SpecialTest_8PriorityLevels()
IEC60730_B_CM4_CM7_reg_fpu.S	Register test	IEC60730B_CM4_CM7_CPU_ControlTest_fpu()
	Register test	IEC60730B_CM4_CM7_CPU_FloatTest1()

	Register test	IEC60730B_CM4_CM7_CPU_FloatTest2()
IEC60730_B_CM4_CM7_flash.S	Invariable memory test	IEC60730B_CM4_CM7_Flash_HWTest()
	Invariable memory test	IEC60730B_CM4_CM7_Flash_SWTest()
	Invariable memory test	IEC60730B_CM4_CM7_Flash_SWTest_32()
IEC60730_B_CM4_CM7_flash_DCP.c	Invariable memory test	IEC60730B_CM7_Flash_HWTest_DCP()
IEC60730_B_CM4_CM7_ram.S	Variable memory test	IEC60730B_CM4_CM7_RAM_AfterResetTest()
	Variable memory test	IEC60730B_CM4_CM7_RAM_RuntimeTest()
	Variable memory test	IEC60730B_CM4_CM7_RAM_SegmentMarchC()
	Variable memory test	IEC60730B_CM4_CM7_RAM_SegmentMarchX()
	Variable memory test	IEC60730B_CM4_CM7_RAM_CopyToBackup()
	Variable memory test	IEC60730B_CM4_CM7_RAM_CopyFromBackup()
IEC60730_B_CM4_CM7_clock.c	Clock test	IEC60730B_CM4_CM7_CLK_SYNC_Init()
	Clock test	IEC60730B_CM4_CM7_CLK_Check()
	Clock test	IEC60730B_CM4_CM7_CLK_SYNC_LPTMR_Isr()
	Clock test	IEC60730B_CM4_CM7_CLK_SYNC_GPT_Isr()
IEC60730_B_CM4_CM7_dio.c	Digital I/O test	IEC60730B_CM4_CM7_DIO_OutputTest_RT()
	Digital I/O test	IEC60730B_CM4_CM7_DIO_OutputTest()
IEC60730_B_CM4_CM7_dio_ext.c	Extended Digital I/O test	IEC60730B_CM4_CM7_DIO_InputTest_Ext()
	Extended Digital I/O test	IEC60730B_CM4_CM7_DIO_ShortToSupplyTest_Set()
	Extended Digital I/O test	IEC60730B_CM4_CM7_DIO_ShortToAdjTest_Set()
	Extended Digital I/O test	IEC60730B_CM4_CM7_DIO_InputTest_Ext_RT()
	Extended Digital I/O test	IEC60730B_CM4_CM7_DIO_ShortToSupplyTest_Set_RT()
	Extended Digital I/O test	IEC60730B_CM4_CM7_DIO_ShortToAdjTest_Set_RT()
IEC60730_B_CM4_CM7_aio.c	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputInit()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputTrigger()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputSet()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputCheck()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputCheck_ke()

	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputSet_k3s()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputCheck_k3s()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputInit_Cyclic()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputSet_Cyclic()
	Analog I/O test	IEC60730B_CM4_CM7_AIO_InputCheck_Cyclic()
IEC60730_B_CM4_CM7_pc.S	Program counter Test	IEC60730B_CM4_PC_Init()
	Program counter Test	IEC60730B_CM4_PC_Test()
	Program counter Test	IEC60730B_CM7_PC_Test()
IEC60730_B_CM4_CM7_pc_object.S	Program counter Test	IEC60730B_PC_object()
IEC60730_B_CM4_CM7_Stack.S	Stack test	IEC60730B_CM4_CM7_Stack_Init()
	Stack test	IEC60730B_CM4_CM7_Stack_Test()
IEC60730_B_CM4_CM7_wdg.c	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_setup()
	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_check()
	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_setup_ke()
	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_check_ke()
	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_setup_k32w()
	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_check_k32w()
	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_setup_RT()
	Watchdog test	IEC60730B_CM4_CM7_watchdog_test_check_RT()



### 3 Core self-test library – object code version

---

The object code version of the library consists of the precompiled binary file and the same list of header files as for source code version of the library, see chapter 2.

Object files for given IDEs:

IAR: IEC60730\_Kinetis\_CM4\_CM7\_Class\_B\_IAR\_v3\_0.a  
 Keil: IEC60730\_Kinetis\_CM4\_CM7\_Class\_B\_Keil\_v3\_0.lib  
 MCUXpresso: IEC60730\_Kinetis\_CM4\_CM7\_Class\_B\_MCUX\_v3\_0.a

### 4 Library structure validation

---

The intention of the validation is to check the content of the source code library version and the object code library version. We proved that the object code version is compiled from valid source code library version.

Validation

Date	Validated by	Validated Revision of Document	Validated Versions of Library	Validation Result
11/2015	Jaroslav Lepka	1.0	1.0	P
11/2015	Jaroslav Lepka	1.1	1.1	P
11/2016	Pavel Sustek	1.2	1.2	P
11/2018	Jaroslav Lepka	3.0	3.0	P

Validation result options:

P – Passed

F – Failed

N/A – Not applicable

**How to Reach Us:**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including **without limitation consequential or incidental damages**. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," **must be validated for each customer application by customer's technical experts**. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:

[nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, Freescale, and the Freescale logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners.

ARM, the ARM logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2016 NXP B.V.

