1 Introduction

IEC60730B_CM4_CM7_4_1 is the actual version of the core self-test library for NXP devices with the CM4 and CM7 core. The library is certified by VDE. It is dedicated for use in applications compliant with the Safety class B standard (specified by IEC 60730, IEC60335 and/or UL 60730, and UL 1998).

The library is released in a precompiled format, together with functional example projects and documentation describing the respective tests.

The library is created in close cooperation with the application team, who have vast experience in customer projects. The customer feedback is also taken into consideration.

2 What is new

When compared to the previous version of the library, the main changes are:

- The WDOG function is consolidated - there is now only one function for one reference timer with parameters that enable you to select a correct WDOG refresh sequence.
- The following new devices are added: I.MX8MM, I.MX8MN, MK2xF, MIMXRT117x.

2.1 Description

The supported devices are as follows:
- MKV3x
- MKV4x
- MKV5x
- MKE1xF
- MK2xF
- MIMXRT10xx
- MIMXRT117x
- MIMX8MNx
- MIMX8MMx

The supported/recommended IDEs are as follows:
- IAR v8.50 and higher
- Keil µVision V5.33(C compiler V6) and higher
- MCUXpresso IDE V11.3 and higher

The tested components are as follows:
- CPU registers
• Program counter
• Variable memory (RAM)
• Invariable memory (flash)
• Clock
• Digital I/O
• Analog I/O
• Stack
• Watchdog
• Touch Sensing Interface (TSI)

3 Optimizations, improvements, and changes

3.1 Library

The WDOG test functions are consolidated. The functions which use the same reference timer are merged into one with a parameter to select the refresh sequence.

The WDOG check function is consolidated.

The backup WDOG type `fs_wdog_test_t` is extended (the previous parts are in bold):

```c
typedef struct {
    uint32_t counter;
    uint32_t resets;
    uint32_t wdTestIncompleteFlag;
    uint32_t RefTimerBase;
    uint32_t WdogBase;
    uint32_t pResetDetectRegister;
    uint32_t ResetDetectMask;
} volatile fs_wdog_test_t;
```

- `RefTimerBase` - The base address of the reference timer
- `WdogBase` - The base address of the WDOG used
- `pResetDetectRegister` - The address of the reset-detect register
- `ResetDetectMask` - The mask for the WDOG reset flag

Table 1. Conversion table for WDOG function

<table>
<thead>
<tr>
<th>New/edited function</th>
<th>Covered old function with new parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>FS_WDOG_Setup_LPTMR()</code></td>
<td>• <code>FS_WDOG_Setup()</code> - Parameters &quot;FS_KINETIS_WDOG&quot;</td>
</tr>
<tr>
<td></td>
<td>• <code>FS_WDOG_Setup_COP()</code> - Parameters &quot;FS_COP_WDOG&quot;</td>
</tr>
<tr>
<td></td>
<td>• <code>FS_WDOG_Setup_KE1XZ()</code> - Parameters &quot;FS_WDOG32&quot;</td>
</tr>
</tbody>
</table>

Table continues on the next page...
### Table 1. Conversion table for WDOG function (continued)

<table>
<thead>
<tr>
<th>New/edited function</th>
<th>Covered old function with new parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS_WDOG_Setup_KE1XF()</td>
<td>• FS_WDOG_Setup_KE1XF() - Parameters &quot;FS_WDOG32&quot;</td>
</tr>
<tr>
<td>FS_WDOG_Setup_IMX_GPT()</td>
<td>• FS_WDOG_Setup_RT() - Parameters &quot;FS_IMXRT&quot;</td>
</tr>
<tr>
<td></td>
<td>• Also suitable for I.MX8mx - Parameters &quot;FS_IMX8M&quot;</td>
</tr>
<tr>
<td>FS_WDOG_Check()</td>
<td>The function has the following new parameters:</td>
</tr>
<tr>
<td></td>
<td>• &quot;clear_flag&quot; - clears the flag of the WDOG reset</td>
</tr>
<tr>
<td></td>
<td>• RegWide8b - if 1 SRS register is read/write as 8b, otherwise as 32b</td>
</tr>
</tbody>
</table>

### 3.2 Documentation

The documents for all test routines are merged into one document.

### 3.3 Examples

The example projects are available only in the MCUXpresso SDK as middleware.

To open an example, use the link from the table at [http://www.nxp.com/iec60730](http://www.nxp.com/iec60730) or perform the following steps:

- Go to [http://mcuxpresso.nxp.com](http://mcuxpresso.nxp.com).
- Click **Select Development Board**.
- Select the supported board and click to add the **Safety** middleware.
- Build and download the SDK package.