MARTe2-core - Test #165

MARTe2 QA-audit (v0.2)

08.07.2015 10:12 - Ivan Herrero

Status:	Closed	**Start date:**	05.07.2015
Priority:	Normal	**Due date:**	23.07.2015
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	0.2	**Spent time:**	0.00 hour

Description

Requirements review

Date of the review: 31/07/2015

Person who did the review: André Neto

Version of requirements: 0.2

Result of review: N/A

List of non-conformities: N/A

Comments: As per the original MARTe implementation, the user stories of this sprint are support functions and do not require formal requirements.

Architecture & design review

Date of the review: 31/07/2015

Person who did the review: André Neto

Version of architecture & design document: 0.2

Result of review: N/A

List of non-conformities: N/A

Comments: As per the original MARTe implementation, the user stories of this sprint are support functions and do not require formal design.

Code and documentation review

Date of the review: 31/07/2015

Person who did the review: Ivan Herrero

Version of source code: 0.2

Result of review: PASS

List of non-conformities: N/A

- The file Source\Core\L0Portability\Architecture\x86_cl\HighResolutionTimerA.h has not been reviewed, because it is pending further discussion.
- The file Source\Core\L0Portability\Architecture\x86_gcc\HighResolutionTimerA.h has not been reviewed, because it is pending further discussion.
- The file Source\Core\L0Portability\OperatingSystem\Linux\HighResolutionTimerCalibratorOS.h has not been reviewed, because it is pending further discussion.
- The file for Source\Core\L0Portability\OperatingSystem\Windows\HighResolutionTimerCalibratorOS.h has not been reviewed, because it is pending further discussion.

- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:92:85: Note 9025: More than two pointer indirection levels used for type 'ThreadInformation ***' [MISRA C++ Rule 5-0-19]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:92:85: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:195:78: Note 925: cast from pointer to pointer [MISRA C++ Rule 5-2-8], [MISRA C++ Rule 5-2-9]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:107: Note 9025: More than two pointer indirection levels used for type 'ThreadInformation ***' [MISRA C++ Rule 5-0-19]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:107: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadsDatabase.cpp:207:118: Note 925: cast from pointer to pointer [MISRA C++ Rule 5-2-8], [MISRA C++ Rule 5-2-9]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.h:47:25: Note 9109: type 'ThreadInformation' previously declared at location 'line 47' [MISRA C++ Rule 3-2-3]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.cpp:93:49: Note 929: cast from pointer to pointer [MISRA C++ Rule 5-2-7]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.h:47:25: Note 9109: type 'ThreadInformation' previously declared at location 'line 47' [MISRA C++ Rule 3-2-3]
- MARTe2-dev/Source/Core/L0Portability/ThreadInformation.h:47:25: Note 9109: type 'ThreadInformation' previously declared at location 'line 47' [MISRA C++ Rule 3-2-3]
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::maxNOfEntries'
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::nOfEntries'
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::entries'
- Warning 459: Function 'SystemThreadFunction(ThreadInformation *)' whose address was taken has an unprotected access to variable 'ThreadsDatabase::internalMutex'
- Threads::name declares that it returns a C style string (char8*) without specifying who is the responsible for the management of the memory used by this char array (it actually returns a pointer to the char array hosted by an instance of ThreadInformation calling its ThreadName method). Perhaps it should be created a copy of the char array by means of Memory::StringDup or changed the prototype putting the char array as an out parameter of the method.

Unit test review

Date of the review: 31/07/2015

Person who did the review: Ivan Herrero

Version of unit tests: 0.2

Result of coverage tests review: PASS

Result of functional tests review: PASS

Result of review: PASS

List of non-conformities:

- TimeoutType lacks unit tests.
- ThreadsDatabase class has not specific unit tests, but it is actually tested through the unit tests of Threads class.
- ThreadInformation class has not specific unit tests, but it is actually tested through the unit tests of Threads class.
- Unit testing coverage note: Some error paths not exercised on #116, #114, #162, #159, #157, #154, #151, #147, #142.
- Unit testing coverage note: In Processor::Family() the sentences inside the block protected by "if (family == 0xf)" has not been exercised, because the processor's family of the processor used in tests has not reached 0xf, so it does not need to use the Extended Family ID.
- Threads unit tests fails at:
  [FAILED] ThreadsGTest.TestPriority
  [FAILED] ThreadsGTest.TestGetThreadInfoCopy
  The reason why these tests fail, is that in Linux a regular user is not allowed to change the priority. This can be solved by either running the tests as the root user (not advisable), or by editing the file /etc/security/limits.conf and adding the following lines (change aneto to your username):

```
@aneto soft rtprio 100
@aneto hard rtprio 10
```

- The following functions test more than one thing and shall be broken down into different functions. This has no impact on the coverage/functional aspects of the test, but will greatly increase readability of the tests and in case of problems will greatly simplify the identification of the culprit :

```
MemoryTest::TestRealloc()
MemoryTest::TestCopy()
MemoryTest::TestMove()
MemoryTest::TestSet()
MemoryTest::TestSearch()
```

- Testing note: Not proper test can be done for the LoadableLibrary::Close() function. It will have to be discussed in the future.
- The following methods of BasicConsole do not have an explicit test, because the answer is different according to the target operating system:

```
bool ColourSupported()
bool ConsoleBufferSupported()
bool CursorPositionSupported()
bool TitleBarSupported()
bool WindowSizeSupported()
bool TimeoutSupported()
```

- The following methods of BasicConsole are not tested because they are not implemented in Linux:

```
ErrorType BasicConsole::ShowBuffer();
ErrorType BasicConsole::SetColour(const Colours &foregroundColour, const Colours &backgroundCo
lour);
ErrorType BasicConsole::SetTitleBar(const char8 * const title);
ErrorType BasicConsole::GetTitleBar(char8 * const title, const uint32 &size) const;
ErrorType BasicConsole::SetCursorPosition(const uint32 &column, const uint32 &row);
ErrorType BasicConsole::GetCursorPosition(uint32 &column, uint32 &row) const;
ErrorType BasicConsole::SetWindowSize(const uint32 &numberOfColumns, const uint32 &numberOfRow
s);
ErrorType BasicConsole::GetWindowSize(uint32 &numberOfColumns, uint32 &numberOfRows) const;
ErrorType BasicConsole::PlotChar(const char8 &c, const Colours &foregroundColour, const Colour
s &backgroundColour, const uint32 &column, const uint32 &row);
```

- Notes: The class HighResolutionTimerCalibratorOS does not have a counterpart on L0, so it is not directly tested because the target of the unit tests are only the classes on L0. Nevertheless, all its methods has been tested indirectly through HighResolutionTimer, except GetInitialTime and GetInitialTicks.
- Note: The tests for HighResolutionTimer::TestCounter, HighResolutionTimer::TestCounter32, and HighResolutionTimer::TestGetTimeStamp, sometimes pass and sometimes fail. The reason is that the operating system dynamically changes the CPU frequency (for power-saving reasons). As a consequence, the frequency retrieved by MARTe when the test starts might not be the same as when the test is executed, generating false errors. This can be resolved by issuing the following command in CentOS: cpupower frequency-set -g performance
- Note: The static declaration/definition of calibratedHighResolutionTimer (of type HighResolutionTimerCalibratorOS) will be refactored in next sprints.

Integration test review

Date of the review: 31/07/2015

Person who did the review: Ivan Herrero

Version of integration tests: 0.2

Result of review: N/A

List of non-conformities: N/A

Comments: As per the original MARTe implementation, the user stories of this sprint are support functions and do not require formal integration tests.

Acceptance test review

Date of the review: 31/07/2015

Person who did the review: Ivan Herrero

Version of acceptance tests: 0.2

Result of review: N/A

List of non-conformities: N/A

Comments: As per the original MARTe implementation, the user stories of this sprint are support functions and do not require formal acceptance tests.

History

#1 - 08.07.2015 10:14 - Ivan Herrero

- Description updated

#2 - 31.07.2015 11:53 - Ivan Herrero

- Description updated

- Target version set to 0.2

- % Done changed from 0 to 100

#3 - 26.10.2015 12:59 - Ivan Herrero

- Status changed from New to Closed

#4 - 30.11.2015 15:58 - Ivan Herrero

- Assignee deleted (Ivan Herrero)