Type III Inheritance   
(End Product from Certified Component)

|  |  |
| --- | --- |
| **Inheritance Profile** | |
| **Child Certification Type** | ☒ End Product (Built on Thread) |
| **Parent Product**  **Certification ID (CID)** | CID should be obtained from Nordic Semiconductor Infocenter Compatibility Matrix:   * nRF52840: <https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcomp_matrix_nrf52840%2FCOMP%2Fnrf52840%2Fnrf52840_thread_cids.html> * nRF5340 <https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcomp_matrix_nrf5340%2FCOMP%2Fnrf5340%2Fnrf5340_thread_cids.html&cp=3_0_3_5>   Click here to enter text.  **Must be provided by the owner of the Parent Product.**  **Thread Group cannot supply it** |
|  | |
| **Thread specification level**  To be selected according to the designed product.  Nordic does support the newest version of the standard which still passes tests for V1.1 and V1.2. | ☐ Thread V1.3 for Matter 1.0 |
| ☐ Thread V1.2 |
| ☐ Thread V1.1 |
| **Thread device roles supported by the Child**  To be selected according to the designed product.  Note that Nordic does not provide libraries for Border Router or On-Mesh Commissioner.Thread Joiner is not used in Matter application by default. | **Mark ALL that apply:** |
| ☐ Minimal Thread Device – MED (Minimal End Device) role |
| ☐ Minimal Thread Device – SED (Sleepy End Device) role |
| ☐ Minimal Thread Device – SSED (Synchronous Sleepy End Device) role |
|  |
| ☐ Full Thread Device [REED behavior] - Leader, Router, REED roles |
| ☐ Full Thread Device [FED behavior] - FED (Full End Device) role |
|  |
| ☐ Border Router |
|  |
| ☐ On-Mesh Commissioner |
| ☐ Thread Joiner |
|  |
| **Explain any difference from the Parent’s Thread device role support** | In case you use the OpenThread MTD or FTD libraries (pre-compiled and not modified) and exact same configuration as in the nRF Connect SDK release, there are no differences between Thread stack we certified vs one used by you. In other case you should clearly describe the changes being made.  Nordic comment: “The Parent’s Thread stack has passed the certification for all the device role defined support. The Child device uses MED/SED/SSED/REED/FED role”.  Click here to enter text. |
|  | |
| **Note/explain any differences from the Parent’s Thread hardware** | Nordic Semiconductor uses nRF52840 DK and nRF5340 DK during ATL and internal testing. You should describe the Radio PCB differences if any (e.g., in antenna design).  Click here to enter text. |
| **Note/explain any differences from the Parent’s Thread stack software** | In case you use the OpenThread MTD or FTD libraries (pre-compiled and not modified) and exact same configuration as in the nRF Connect SDK release, there are no differences between Thread stack we certified vs one used by you. In other case you should clearly describe the changes being made.  Nordic comment: “The Child product’s application is built on top of nRF Connect SDK X.Y.Z. The Thread stack (OpenThread) included in the Child product is taken “as-is” in pre-compiled (binary) form. There are no differences between Paren’t Thread stack and Child’s Thread stack.”  Click here to enter text. |
|  | |
| **Describe how the certified Thread software stack revision, distribution and Child implementation are controlled to insure the Child product truly contains the specified, certified Thread stack** | In case you use the OpenThread MTD or FTD libraries (pre-compiled and not modified) and exact same configuration as in the nRF Connect SDK release, there are no differences between Thread stack we certified vs one used by you. In other case you should clearly describe the changes being made.  Nordic comment: “The Child product’s application is built on top of nRF Connect SDK X.Y.Z. The Thread stack (OpenThread) included in the Child product is taken “as-is” in pre-compiled (binary) form. No modification has been made to original libraries.”  Click here to enter text. |
| **Describe how the certified Thread software stack functionality is called/controlled/abstracted in the Child’s operation** | This depends on the product designed. In case of Matter or HomeKit the Thread stack is controlled directly by those stacks. For example, for Matter application:  Nordic comment: “The Thread stack data path is integrated inside Zephyr Networking stack – in the same way as it is integrated in the Parent’s stack. The Thread device configuration is limited to basic functionalities as managing Thread credentials, starting/stopping Thread stack, configuring Thread role using OpenThread API, or adding IPv6 multicast addresses.”  Click here to enter text. |
|  | |
| **Describe/include any test/analytical data which demonstrates the Child’s equivalent Thread functionality** | Nordic comment: “The Child product’s application is built on top of nRF Connect SDK X.Y.Z. The Thread stack (OpenThread) included in the Child product is taken “as-is” in pre-compiled (binary) form. No modification has been made to original libraries  To ensure that the stack has not been accidently modified, the following diff check has been made and showed no differences.  $ cd nrfconnect/nrfxlib $ git diff --stat v2.2.0 openthread/  Additionally to ensure that the pre-build libraries were used in the build process, the final map file was inspected:  $ cat build/zephyr/zephyr.map | grep -E "(^LOAD).\*libopenthread-mtd" LOAD /home/user/git/nrfconnect/nrfxlib/openthread/lib/cortex-m4/hard-float/v1.3/mtd/oberon/libopenthread-mtd.a”  Optional: Finally, the output of the provided `ot version` shell command matches the one present in the parent’s testing report.”  Click here to enter text. |