AN13690

Enable More USB Examples on M4 Core for RT1170 or M7 Core for RT1180

Rev. 1 — 20 November 2022   **Application note**

1   Introduction

RT1170 works with M7 and M4, on which all USB examples can run. However, all USB examples in the release package are enabled on M7. Only example `host_hid_mouse` and `dev_hid_mouse` are enabled on M4. As for RT1180 works with M7 and M33, all USB examples in the release package are enabled on M33. Only example host_hid_mouse and dev_hid_mouse are enabled on M7. Since USB projects on both cores share demo source files but with different project settings, examples can be migrated.

This document describes how to migrate USB examples from M7 to M4 or from M33 to M7 with different toolchains. There are four main steps required:

1. Create an RT1170 M4/RT1180 M7 project.
2. Rearrange source files.
3. Rearrange project files.
4. Adjust project settings.

This document takes the `dev_msc_disk_bm` as an example to show how to enable USB examples on M4 core of RT1170 or on M7 core of RT1180 with IAR, MDK, ARMGCC, and MCUXpresso IDE.
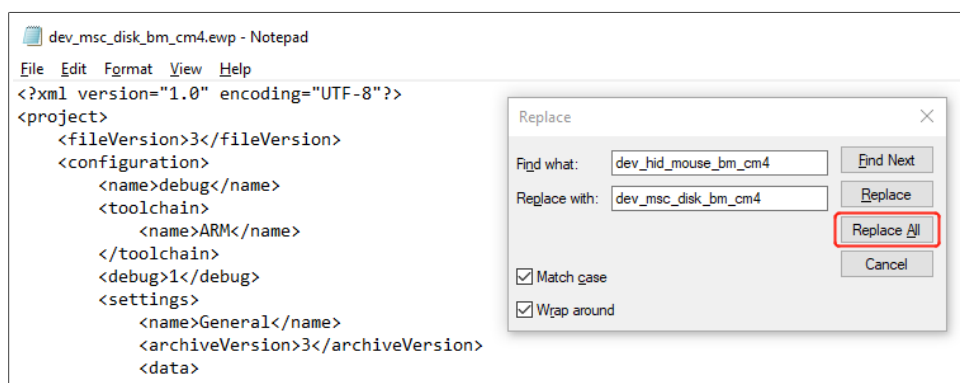
2   IAR

1. **Create an RT1170 M4/RT1180 M7 project**
   a. For RT1170, copy the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_hid_mouse/bm/cm4* to the folder where the example must be enabled.
   In this case, copy the *cm4* folder to *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_msc_disk/bm*.
   For RT1180, copy the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_hid_mouse/bm/cm7* to the folder where the example must be enabled.
   In this case, copy the *cm7* folder to *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_msc_disk/bm*.
   b. For RT1170, open the *iar* folder under *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_msc_disk/bm/cm4/iar*.
   For RT1180, open the *iar* folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_msc_disk/bm/cm7/iar*.
   c. For RT1170, change the filename `dev_hid_mouse_bm_cm4` to `dev_msc_disk_bm_cm4` respectively.
   For RT1180, change the filename `dev_hid_mouse_bm_cm7` to `dev_msc_disk_bm_cm7` respectively.

d. For RT1170, open the `dev_msc_disk_bm_cm4.eww` and `dev_msc_disk_bm_cm4.ewp` files with a text editor, such as notepad, notepad++, sublime or Visual studio code.
For RT1180, open the `dev_msc_disk_bm_cm7.eww` and `dev_msc_disk_bm_cm7.ewp` files with a text editor, such as notepad, notepad++, sublime or Visual studio code.

e. For RT1170, search and replace all `dev_hid_mouse_bm_cm4` with `dev_msc_disk_bm_cm4`, and then save the files.

For RT1180, search and replace all `dev_hid_mouse_bm_cm7` with `dev_msc_disk_bm_cm7`, and then save the files.

2. **Rearrange source files**

   a. For RT1170, open the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_msc_disk/bm/cm4*, and delete all files with the `.c` and `.h` extension.
   For RT1180, open the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_msc_disk/bm/cm7*, and delete all files with the `.c` and `.h` extension.

   b. For RT1170, copy files with the `.c` and `.h` extension in the *cm7* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/ cm7* to the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/ usb_device_msc_disk/bm/cm4*.

For RT1180, copy files with the .c and .h extension in the *cm33* folder under
*<install_dir>/boards/evkmimxrt1180/usb_examples/usb_device_msc_disk/bm/
cm33* to the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_examples/
usb_device_msc_disk/bm/cm7*.

3. **Rearrange project files**

Note: *The following steps are described for RT1170. However, these steps are also applicable for RT1180.*

a. Open `dev_msc_disk_bm_cm7` and `dev_msc_disk_bm_cm4` IAR project respectively. The two workspaces are respectively located under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm7/iar* and *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/iar*.

b. Compare the whole project directory. Find file groups that are in the `cm7` project but not `cm4` and add them to the `cm4` project.

c. Compare the two groups with the same name. Delete files that are in the `cm4` project but not `cm7`. Find files that are in the `cm7` project but not `cm4` and add them to the `cm4` project.

In this case,

- For the `board` group, add `sdmmc_config` source files to `board` group in the `cm4` project.
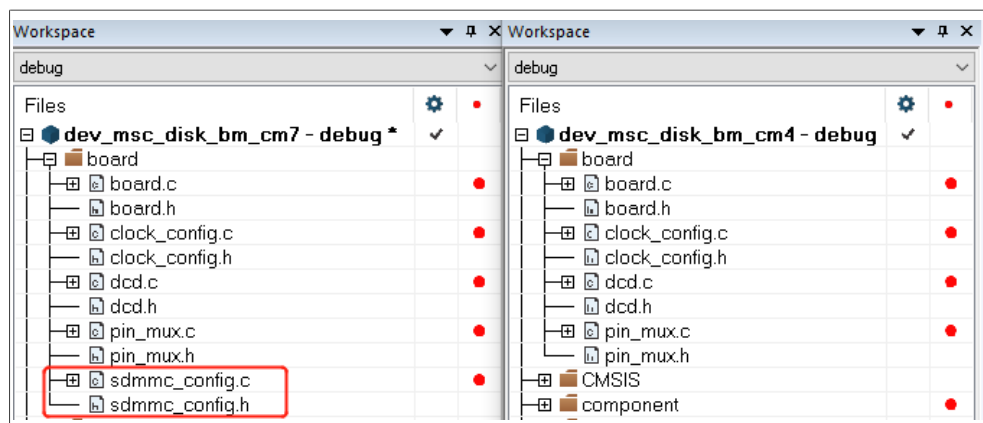
- For the `component` group, delete files associated with timer from `cm4` project and add no files.
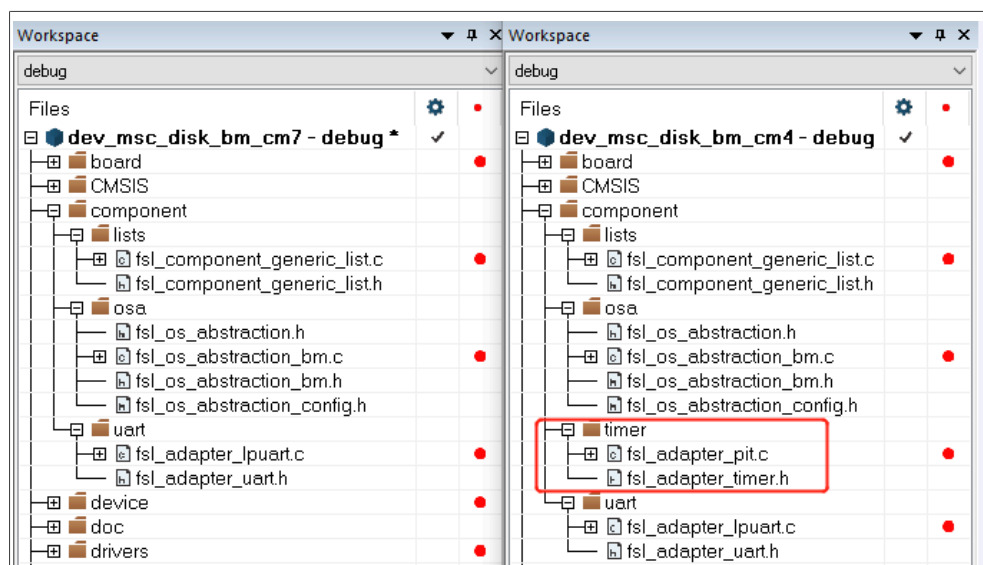
- For the `drivers` group, delete `fsl_pit` source files from the `cm4` project and add `fsl_usdhc` source files to the `cm4` project.

Rev. 1 — 20 November 2022
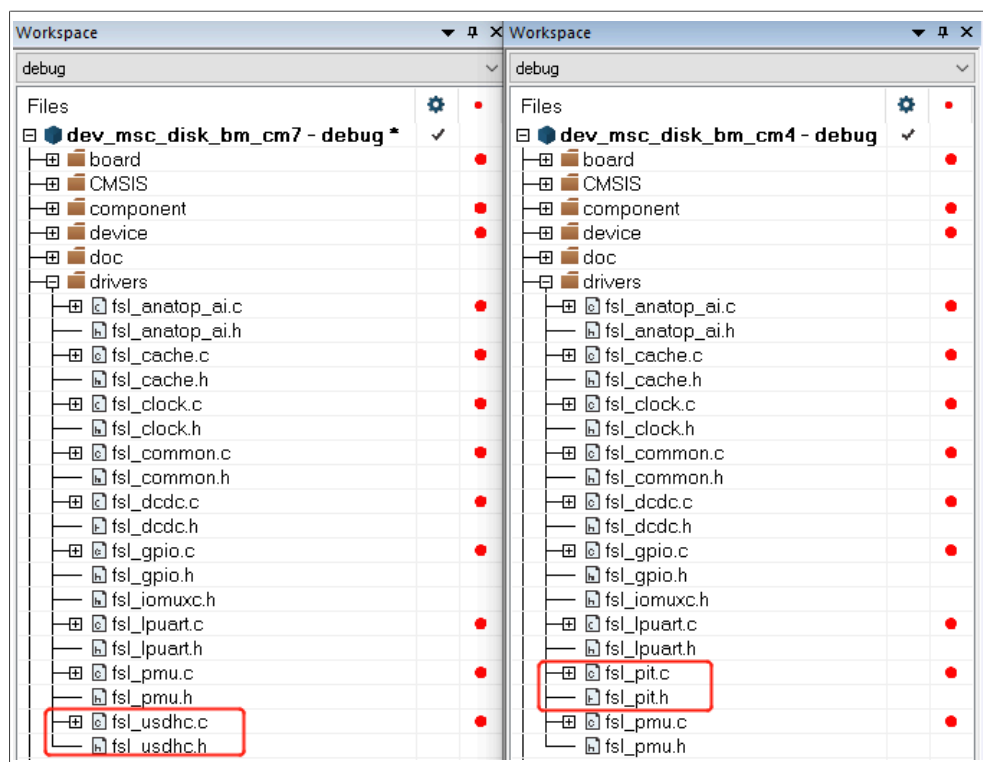
- For the `sdmmc` group, there is not `sdmmc` group in the `cm4` project. Add `sdmmc` group and its children to the `cm4` project.
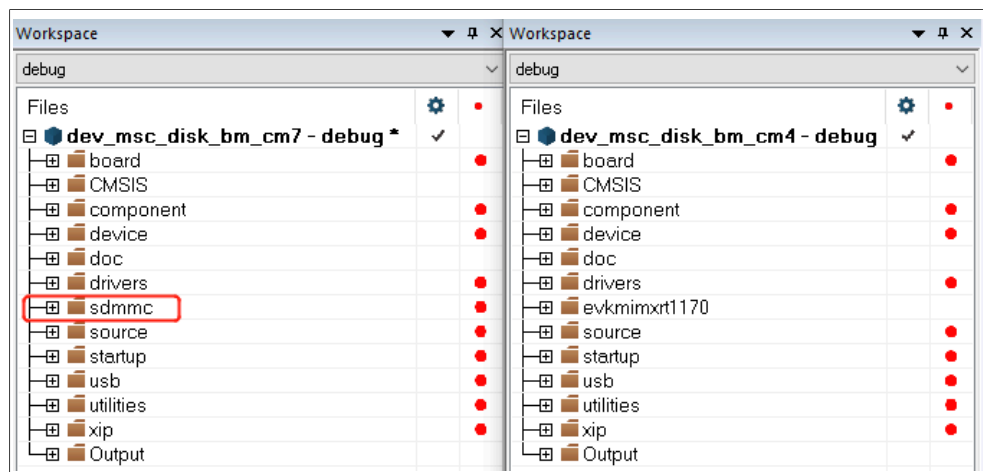
- For the `source` group, delete the files in the `cm4` source group and add the files under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/ bm/cm4* with the same name as in the `cm7` project.
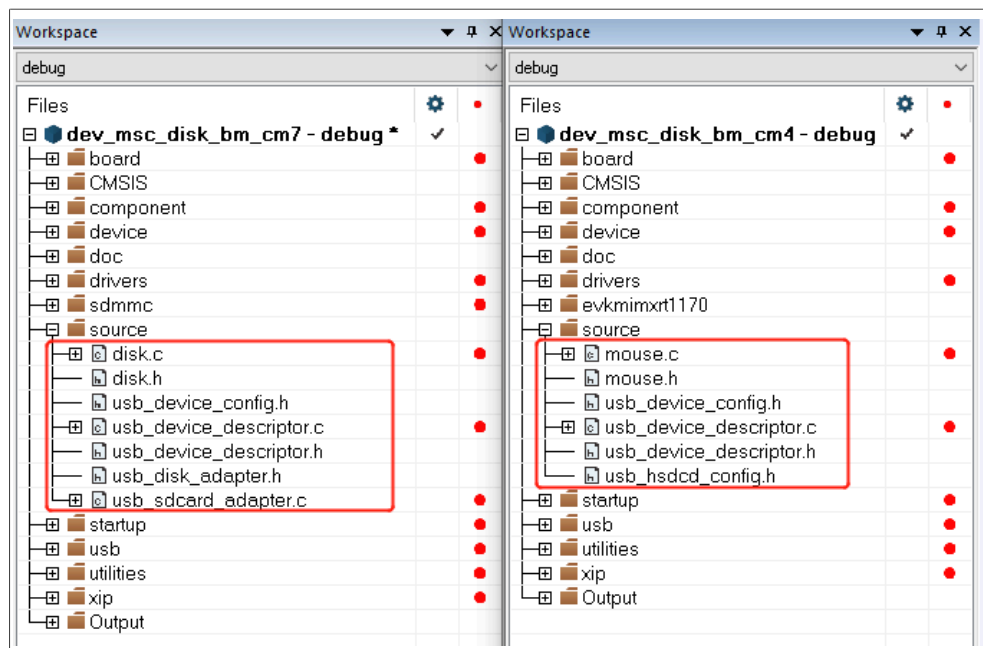
- For the `usb` group, replace the `hid` group and its children in the `cm4` project with `msc` group in the `cm7` project and delete the `dcd` group.
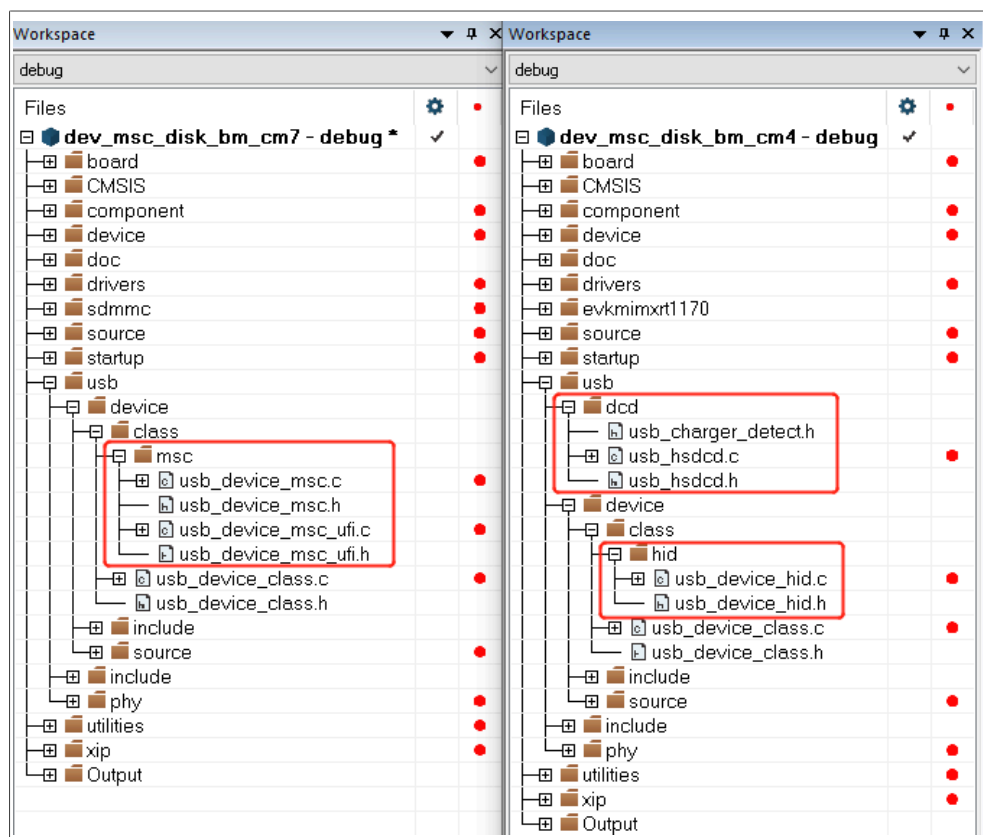
4. **Adjust project settings**

Note: *The following steps are described for RT1170. However, these steps are also applicable for RT1180.*

a. Compare the macro in the project settings: **Option - C/C++ compiler** > **Preprocessor**. Add the macro that is in the `cm7` project but not `cm4` to the `cm4` project.

In this case, add the `SD_ENABLED` macro to the `cm4` project.
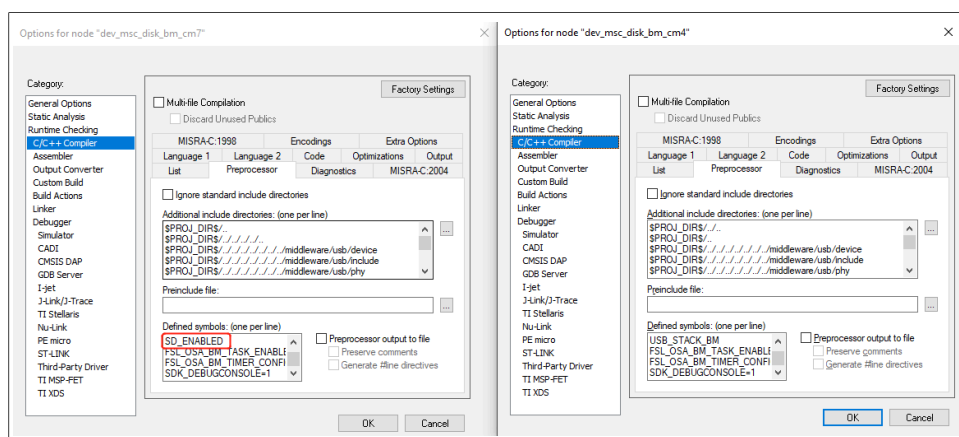
b. Change the include directories in the project settings, by clicking **Option** > **C/C++ compiler** > **Preprocessor**, according to the changes of project files.

In this case, delete the directories framed in red and add the directories in green.
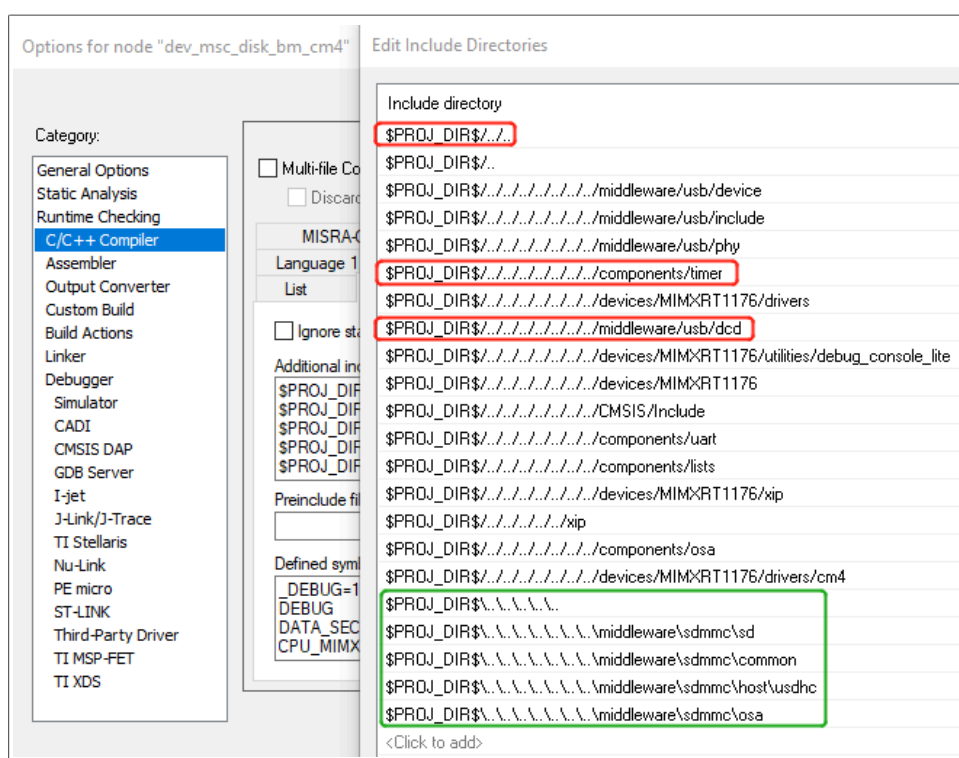
With all above steps done, the RT1170 M7 project successfully changes to an M4 project and the M4 example USB project is available for downloading and debugging. The RT1180 M33 project successfully changes to an M7 project and the M7 example USB project is downloading and debugging.

Rev. 1 — 20 November 2022

3 MDK

1. **Create an RT1170 M4/RT1180 M7 project**

   a. For RT1170, copy the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_hid_mouse/bm/cm4* to the folder where the example must be enabled.

   In this case, copy the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_hid_mouse/bm/cm4* to the *bm* folder under *<install_dir>/ boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm*.

   For RT1180, copy the *cm7*folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_hid_mouse/bm/cm7* to the folder where the example must be enabled.

   In this case, copy the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_hid_mouse/bm/cm7* to the *bm* folder under *<install_dir>/ boards/evkmimxrt1180/usb_examples/usb_device_msc_disk/bm*.

   b. For RT1170, open the *mdk* folder under *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_msc_disk/bm/cm4/mdk*.

   For RT1180, open the *mdk* folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_msc_disk/bm/cm7/mdk*.

   c. For RT1170, change the filename *dev_hid_mouse_bm_cm4* to *dev_msc_disk_bm_cm4* respectively.

   For RT1180, change the filename *dev_hid_mouse_bm_cm7* to *dev_msc_disk_bm_cm7* respectively.

   d. For RT1170, open the *dev_msc_disk_bm_cm4.uvmpw*, *dev_msc_disk_bm_cm4.uvoptx, and dev_msc_disk_bm_cm4.uvprojx* files with a text editor, such as notepad, notepad++, sublime or Visual studio code.

   For RT1180, open the *dev_msc_disk_bm_cm4.uvmpw*, *dev_msc_disk_bm_cm7.uvoptx, and dev_msc_disk_bm_cm7.uvprojx* files with a text editor, such as notepad, notepad++, sublime or Visual studio code.

   e. For RT1170, search and replace all `dev_hid_mouse_bm_cm4` with `dev_msc_disk_bm_cm4`, and then save the files.

For RT1180, search and replace all `dev_hid_mouse_bm_cm7` with `dev_msc_disk_bm_cm7`, and then save the files.

2. **Rearrange source files**

   a. For RT1170, open the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4*, and delete all files with the `.c` and `.h` extension.
   For RT1180, open the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_examples/usb_device_msc_disk/bm/cm7*, and delete all files with the `.c` and `.h` extension.

   b. For RT1170, copy files with the `.c` and `.h` extension in the *cm7* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm7* to the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4*.

For RT1180, copy files with the `.c` and `.h` extension in the *cm33* folder under *<install_dir>/boards/evkmimxrt1180/usb_examples/usb_device_msc_disk/bm/cm33*

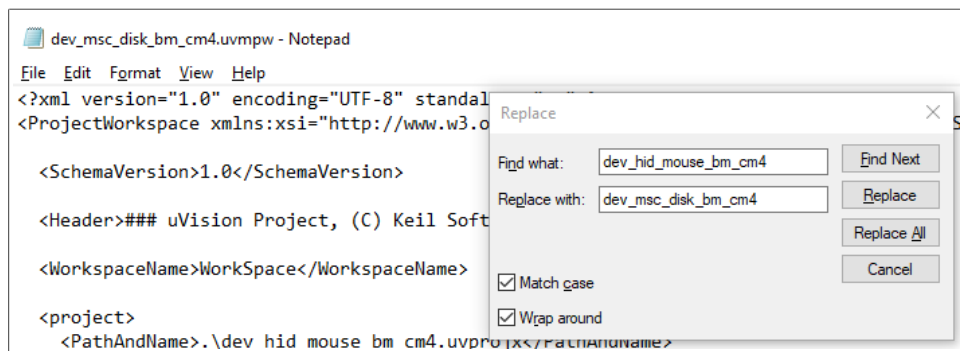to the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_examples/usb_ device_msc_disk/bm/cm7*.

3. **Rearrange project files**

   Note: *The following steps are described for RT1170. However, these steps are also applicable for RT1180.*

   a. Open `dev_msc_disk_bm_cm7` and `dev_msc_disk_bm_cm4` MDK project respectively. The two workspaces are respectively located in *<install_dir>/boards/ evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm7/mdk* and *<install_ dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/ mdk*.

   b. Compare the whole project directory, delete groups that are in the `cm4` project but not `cm7`. In the `cm4` project, add groups that are in the `cm7` project but not `cm4`.

   c. Compare the two groups with the same name. Delete files that are in the `cm4` project but not `cm7`. Add files that are in the `cm7` project but not `cm4`.

   In this case,

   • Delete the groups framed in the red box in the `cm4` project and add the directories framed in the green box to the `cm4` project.
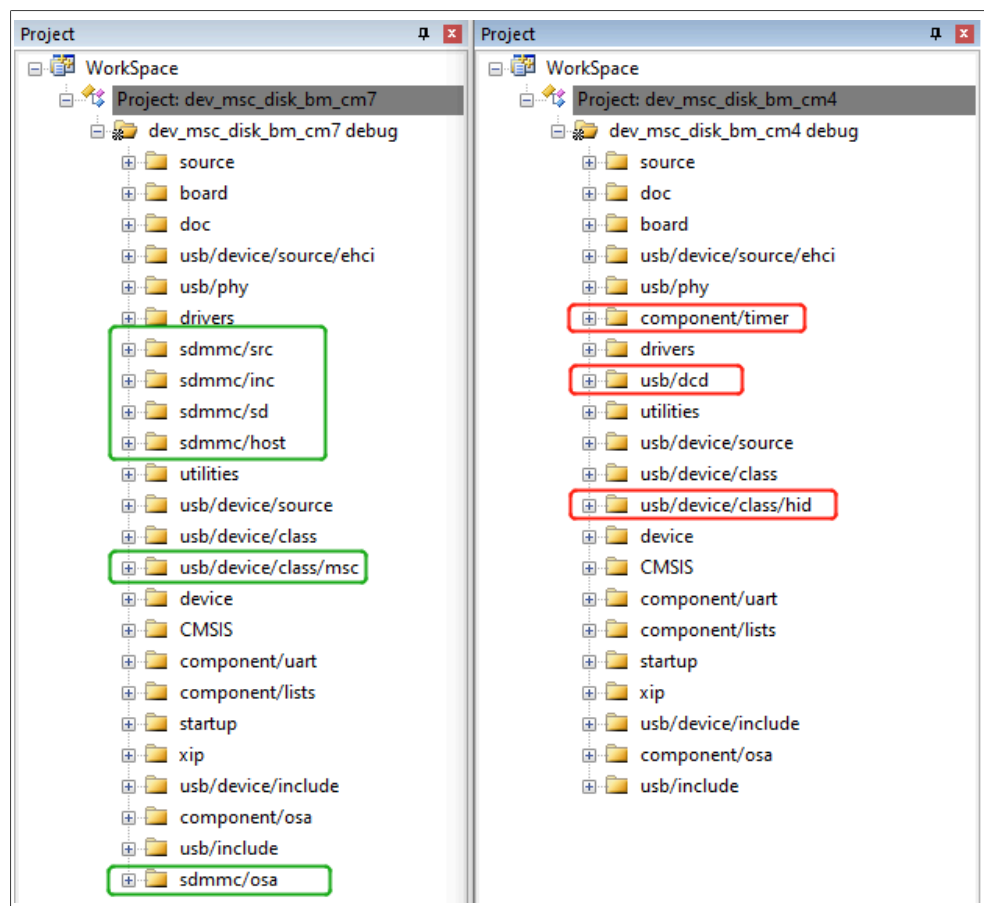
   • For the `source` group, delete the files in the *source* group in the `cm4` project and add files under *<install_dir>boards/evkmimxrt1170/usb_examples/usb_device_ msc_disk/bm/cm4* with the same name as in `cm7` project.
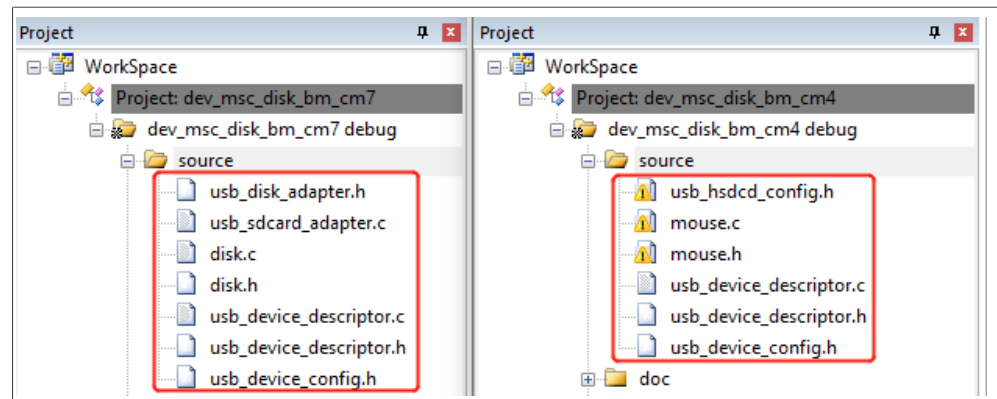
- For the `board` group, add *sdmmc_config* source files to the *board* group in the `cm4` project.

- For the `drivers` group, delete `fsl_pit` source files in the `cm4` project and add `fsl_usdhc` source files to the `cm4` project.

4. **Adjust project settings**
   Note: The following steps are described for RT1170. However, these steps are also applicable for RT1180.

   a. Compare the macro in the project settings by clicking **Option** -> **C/C++**. Add a macro that is in the `cm7` project but not `cm4`.
      In this case, add the *SD_ENABLED* macro to the `cm4` project.
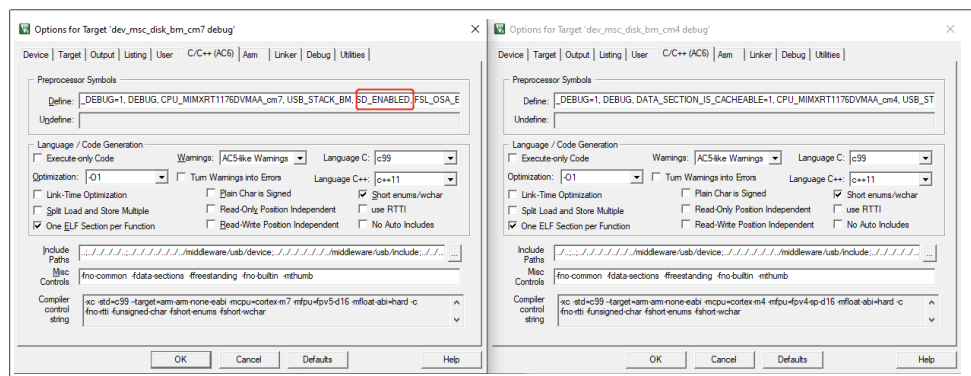
AN13690

Application note

All information provided in this document is subject to legal disclaimers.

Rev. 1 — 20 November 2022

© 2022 NXP B.V. All rights reserved.

13 / 24

b. Change the include paths in the project settings, by clicking **Option** > **C/C++**, according to the changes of project files.
In this case, delete the paths framed in the red box and add the paths framed in the green box.

With all above steps done, the RT1170 M7 project successfully changes to an M4 project and the M4 example USB project is available for downloading and debugging. The RT1180 M33 project successfully changes to an M7 project and the M7 example USB project is downloading and debugging.

4 ARMGCC

1. **Create an RT1170 M4/RT1180 M7 project**
   a. For RT1170, copy the *cm4* folder under *<install_dir> /boards/evkmimxrt1170/usb_examples/usb_device_hid_mouse/bm/cm4* to the folder where the example must be enabled.
   In this case, copy *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_ device_hid_mouse/bm/cm4* to *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_msc_disk/bm*.
   For RT1180, copy the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_hid_mouse/bm/cm7* to the folder where the example must be enabled.
   In this case, copy *<install_dir>/boards/evkmimxrt1180/usb_examples/usb_ device_hid_mouse/bm/cm7* to *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_msc_disk/bm*.
   b. For RT1170, open the *CMakeLists.txt* of the example. The file is located under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/ cm4/ armgcc*.
   For RT1180, open the *CMakeLists.txt* of the example. The file is located under *<install_dir>/boards/evkmimxrt1180/usb_examples/usb_device_msc_disk/bm/ cm7/ armgcc*.
   c. For RT1170, search and replace all `dev_hid_mouse_bm_cm4` with `dev_msc_disk_bm_cm4`, and then save the files.
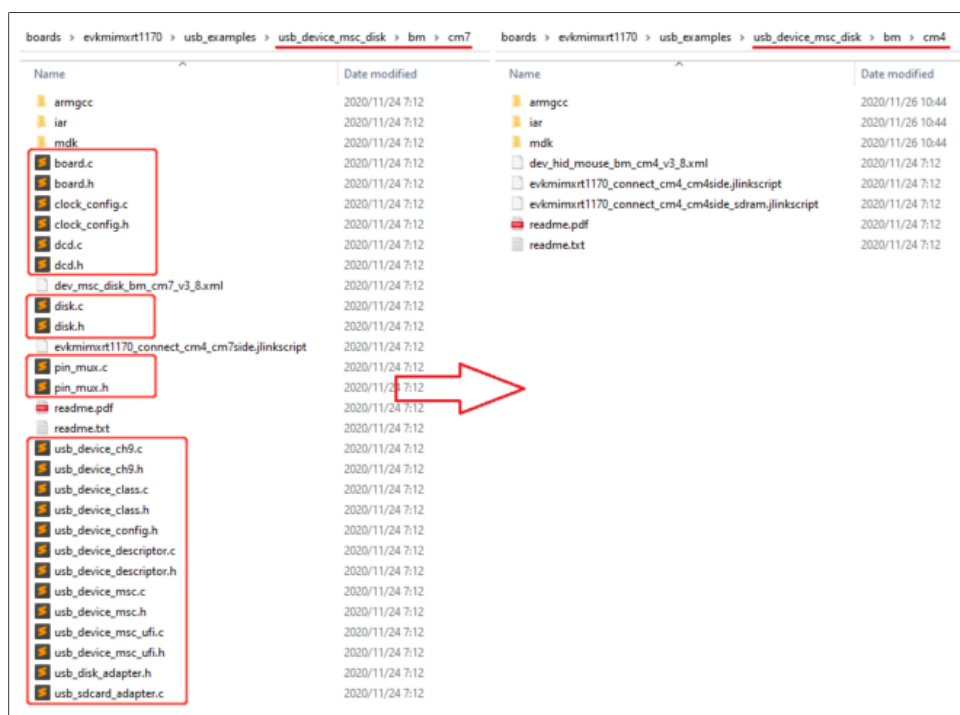   For RT1180, search and replace all `dev_hid_mouse_bm_cm7` with `dev_msc_disk_bm_cm7`, and then save the files.
2. **Rearrange source files**
   a. For RT1170, open the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_ examples/usb_device_msc_disk/bm/cm4* folder and delete all files with the `.c` and `.h` extension.

For RT1180, open the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_ examples/usb_device_msc_disk/bm/cm7* folder and delete all files with the `.c` and `.h` extension.

b. For RT1170, copy files with the `.c` and `.h` extension in the *cm7* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/ cm7* to the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/ usb_device_msc_disk/bm/cm4*.

For RT1180, copy files with the `.c` and `.h` extension in the *cm33* folder under *<install_dir>/boards/evkmimxrt1180/usb_examples/usb_device_msc_disk/bm/ cm33* to the *cm7* folder under *<install_dir>/boards/evkmimxrt1180/usb_examples/ usb_device_msc_disk/bm/cm7*.
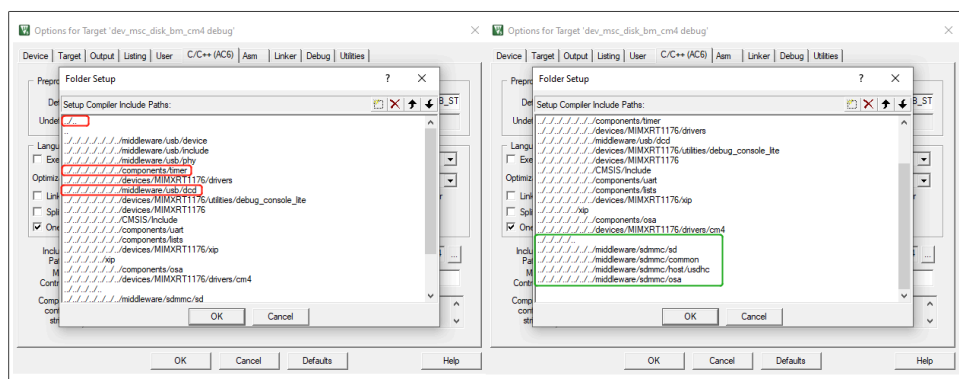

3. **Rearrange project files**

   Note: The following steps are described for RT1170. However, these steps are also applicable for RT1180.

   a. Open the *CMakeLists.txt* of the two examples respectively. The two files are respectively located under *<install_dir>/boards/evkmimxrt1170/usb_examples/ usb_device_msc_disk/bm/cm7/armgcc* and *<install_dir>/boards/evkmimxrt1170/ usb_examples/usb_device_msc_disk/bm/cm4/ armgcc*.

   b. Search the **add_executable** section in the *cm7* and *cm4* projects. Delete files that are in the `cm4` project but not `cm7`. Add files that are in the `cm7` project but not `cm4` to the `cm4` project.

   In this case, delete the files in red box and add the directories in the green box to the `cm4` project.
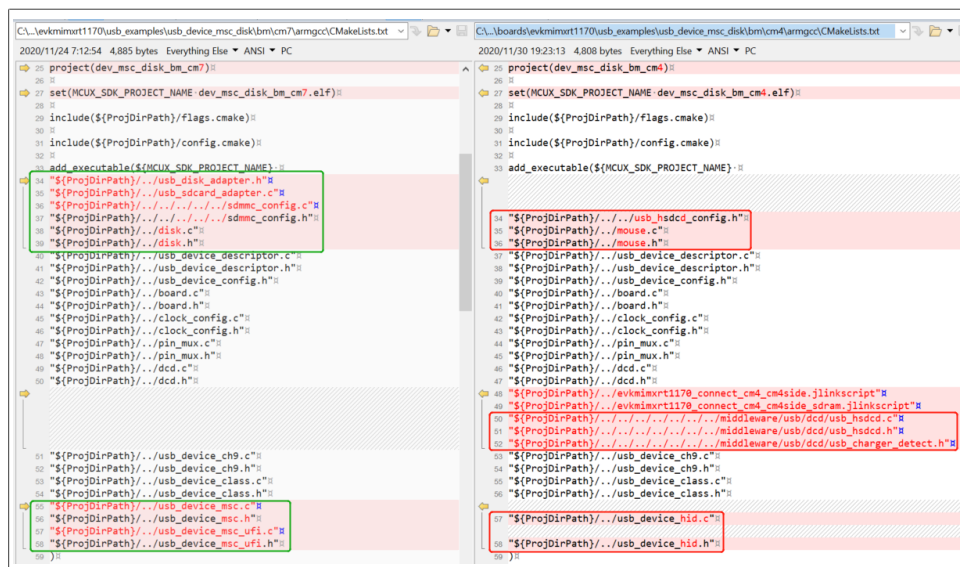
4. **Adjust project settings**

Note: The following steps are described for RT1170. However, these steps are also applicable for RT1180.

a. Search the **target_include_directories**, **CMAKE_MODULE_PATH**, and **# include modules** section one by one. Add macros that are in the `cm7` project but not `cm4` to the `cm4` project.

In this case, remove the paths framed in the red box from the `cm4` project and add the directories framed in the green box.

Note: Modify the include modules with the `cm7` suffix to `cm4`.

For example, modify `driver_usdhc_MIMXRT1176_cm7` *to* `driver_usdhc_MIMXRT1176_cm4.`

AN13690

Application note

All information provided in this document is subject to legal disclaimers.

Rev. 1 — 20 November 2022

© 2022 NXP B.V. All rights reserved.

16 / 24
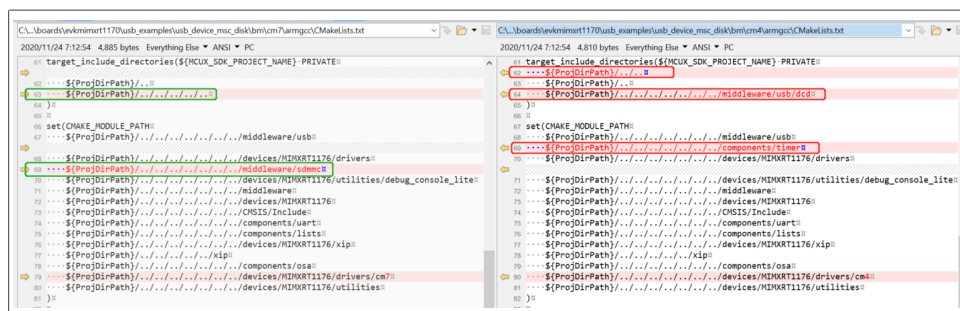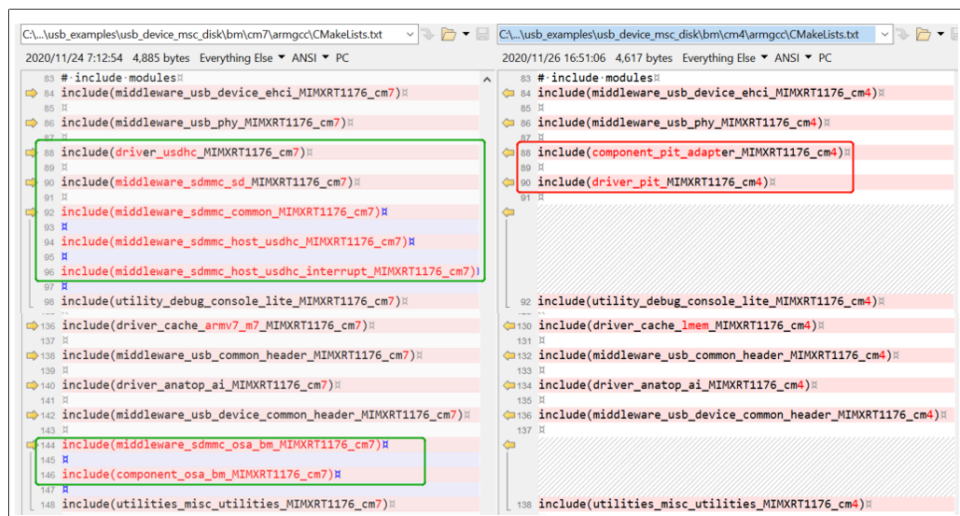
b. Open the `flags.cmake` of the two examples respectively. The two files are respectively located in *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm7/armgcc* and *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/ armgcc*.

c. Search the **CMAKE_C_FLAGS_DEBUG** section in the `cm4` and `cm7` projects. Compare the macros and add the macro that is in the `cm7` project but not `cm4` to the `cm4` project.
In this case, add the *SD_ENABLED* macro to the `cm4` project.
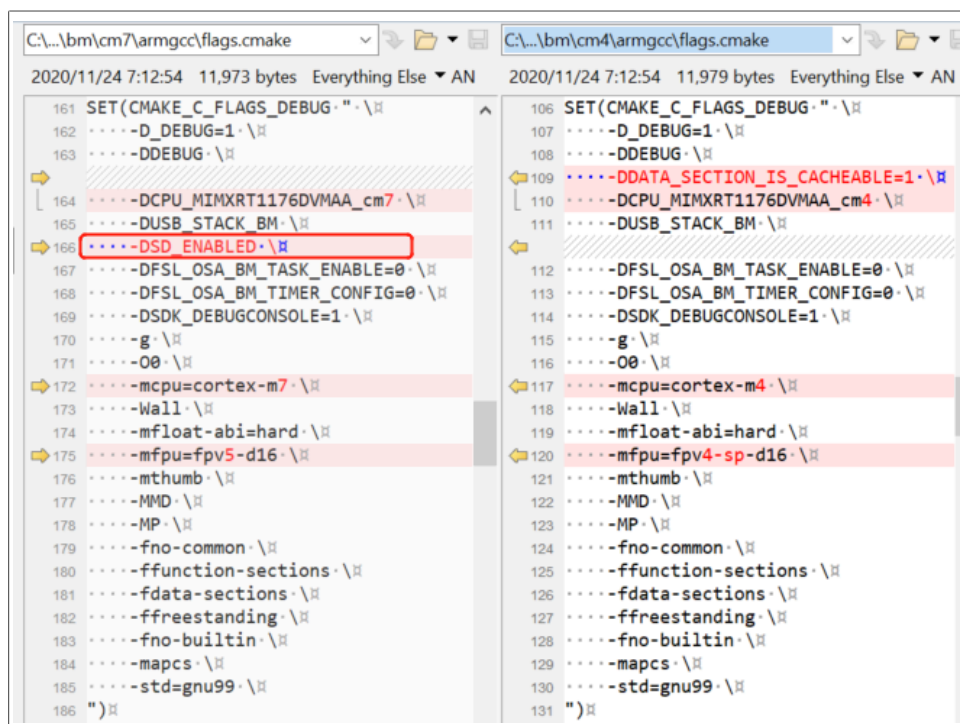
With all above steps done, the RT1170 M7 project successfully changes to an M4 project and the M4 example USB project is available for downloading and debugging. The RT1180 M33 project successfully changes to an M7 project and the M7 example USB project is downloading and debugging.

5 MCUXpresso IDE

1. **Rearrange source files**

   a. For RT1170, import `dev_msc_disk_bm_cm7` and `dev_hid_mouse_bm_cm4` example. For RT1180, import `dev_msc_disk_bm_cm33` and `dev_hid_mouse_bm_cm4` example.

   b. Open the directory of the two examples in the file explorer.

   c. For RT1170, compare the whole project directory and add the folders that are in the `cm7` project but not `cm4` to the `cm4` project. For RT1180, compare the whole project directory and add the folders that are in the `cm33` project but not `cm7` to the `cm7` project.

   d. Compare the two folders. For RT1170, delete the files that are in the `cm4` project but not `cm7` and add the files that are in the `cm7` project but not `cm4` to the `cm4` project. For RT1180, delete the files that are in the `cm7` project but not `cm33` and add the files that are in the `cm33` project but not `cm7` to the `cm7` project.

   In the RT1170 case,

   - For the `board`, `source`, and `usb` folders, replace them in the `cm4` project with the ones in the `cm7` directory.

   - For the `component` folder, delete the `timer` folder in the `cm4` directory.

   - For the `drivers` folder, delete `fsl_pit` source files in the `cm4` project and add `fsl_usdhc` source files to the `cm4` directory.

- For the `sdmmc` folder, copy it to the `cm4` project.

2. **Rearrange project files**

   After the source files are updated, the project files are loaded to the `cm4` project of RT1170 or the `cm7` project of RT1180 automatically. No manual updates to the project files are required.

   Due to the deletion in Rearrange source files, for the *board*, *source* and *usb* folders, an option must be deleted in the properties. Right click **board** -> **properties** -> **C/C++build** and uncheck **Exclude resource from build**. Perform the same operation in the **source** and **usb** folders. Take RT1170 as an example:
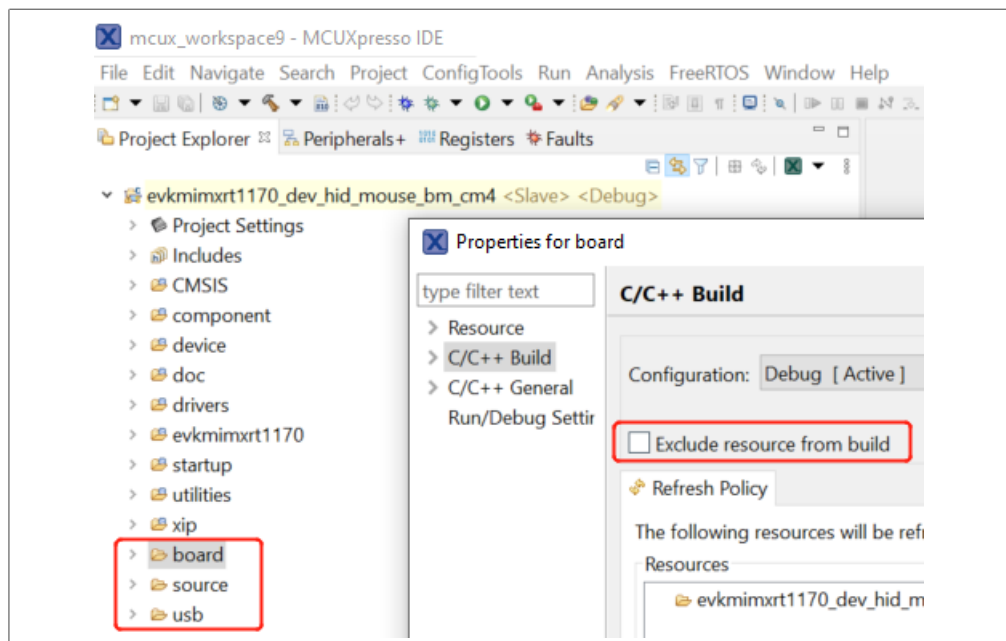
3. **Adjust project settings**
   Note: The following steps are described for RT1170. However, these steps are also applicable for RT1180.

   a. Compare the macro in the project settings, by clicking **Properties** -> **C/C++ Build** -> **Setting** -> **MCU C Compiler** -> **Preprocessor**. Add the macro that is in the cm7 project but not in cm4 to the cm4 project.
   In this case, the macro is SD_ENABLED.
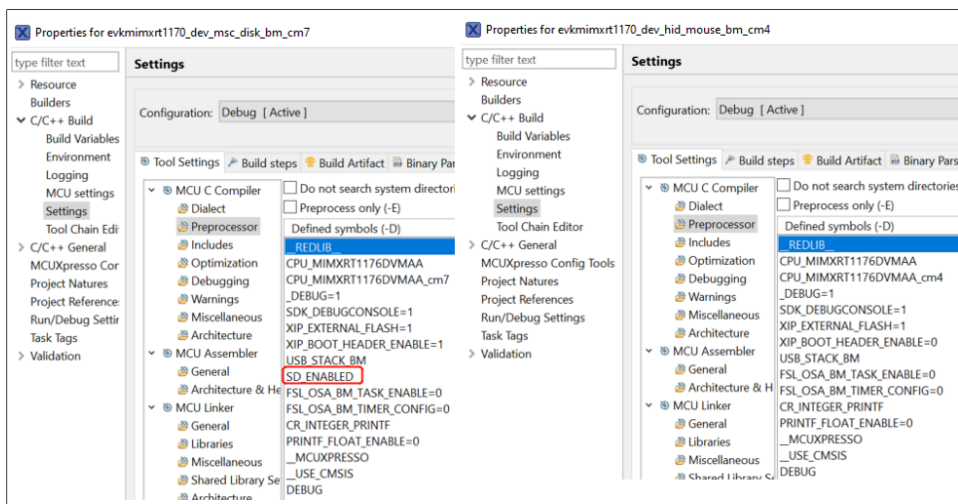
   b. Change the include paths in the project settings, by clicking **Properties** -> **C/C++ Build** -> **Setting** -> **MCU C Compiler** -> **Includes**, according to the changes of project files.
   In this case, delete the paths framed in the red box and add the paths in the green box.
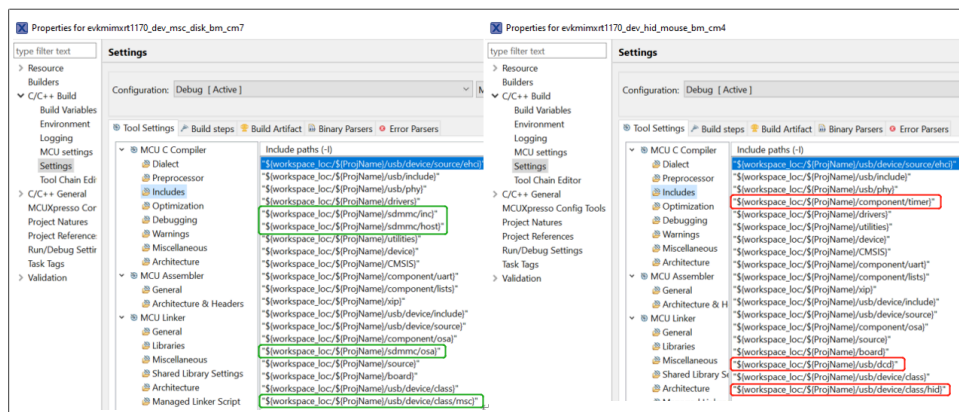
4. **Change project name**

   a. Right-click **project name** > **rename**, change the filename `dev_hid_mouse_bm_cm4` to `dev_msc_disk_bm_cm4`, and then click **OK**.
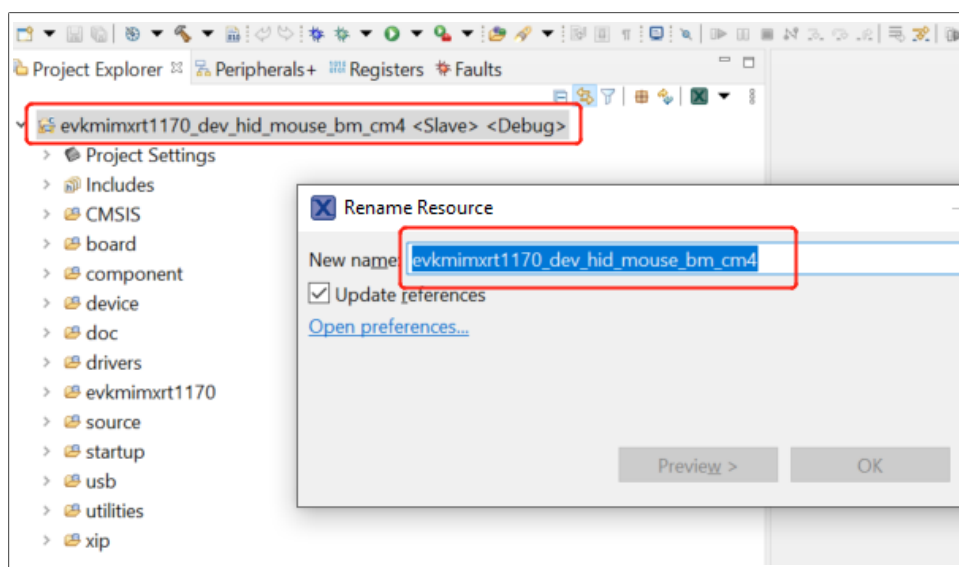
With all above steps done, the RT1170 M7 project successfully changes to an M4 project and the M4 example USB project is available for downloading and debugging. The RT1180 M33 project successfully changes to an M7 project and the M7 example USB project is downloading and debugging.

6 Note

1. The above steps provide the method based on the `dev_hid_mouse_bm` example to enable the `dev_msc_disk_bm` example on the M4 core of RT1170 or on the M7 core of RT1180. For other examples of RT1170, similar way applies to change an M7 project to an M4 project. For other examples of RT1180, similar way applies to change an M33 project to an M7 project.
2. To enable FreeRTOS examples, refer to *dev_hid_mouse_freertos*.
3. To enable USB host examples, refer to *host_hid_mouse_bm* or *host_hid_mouse_freertos*.

7   Revision history

This table summarizes revisions to this document.

Table 1. Revision history

Revision number	Date	Substantive changes
0	10 July 2022	Initial release
1	20 November 2022	Editorial and layout update.

AN13690

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

Application note

Rev. 1 — 20 November 2022

22 / 24

8 Legal information

8.1 Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document 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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

8.3 Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

Contents