AN13114

Enable More USB Examples on M4 Core for RT1170

Rev. 1 — 01 June 2021 Application Note

Contents

1 Introduction....1
2 IAR....1
3 MDK....7
4 ARMGCC....11
5 MCUXpresso IDE....14
6 Note....18
7 Revision history....18

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. Since USB projects on both cores share the same demo source files but with different project settings, examples are migratable.

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

1. Create an M4 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 with IAR, MDK, ARMGCC and MCUXpresso IDE.

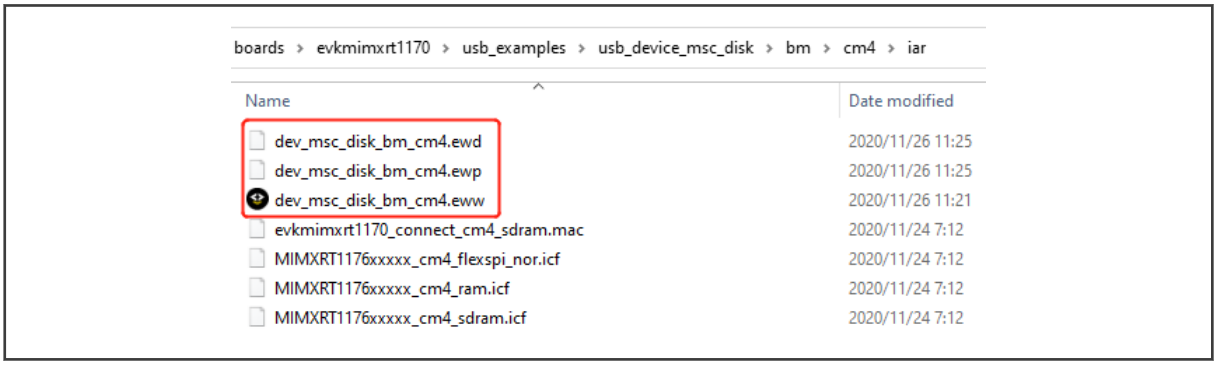
2 IAR

1. **Create an M4 project**
   a. Copy the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_hid_mouse/bm/cm4* to the folder where the example needs to be enabled.

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

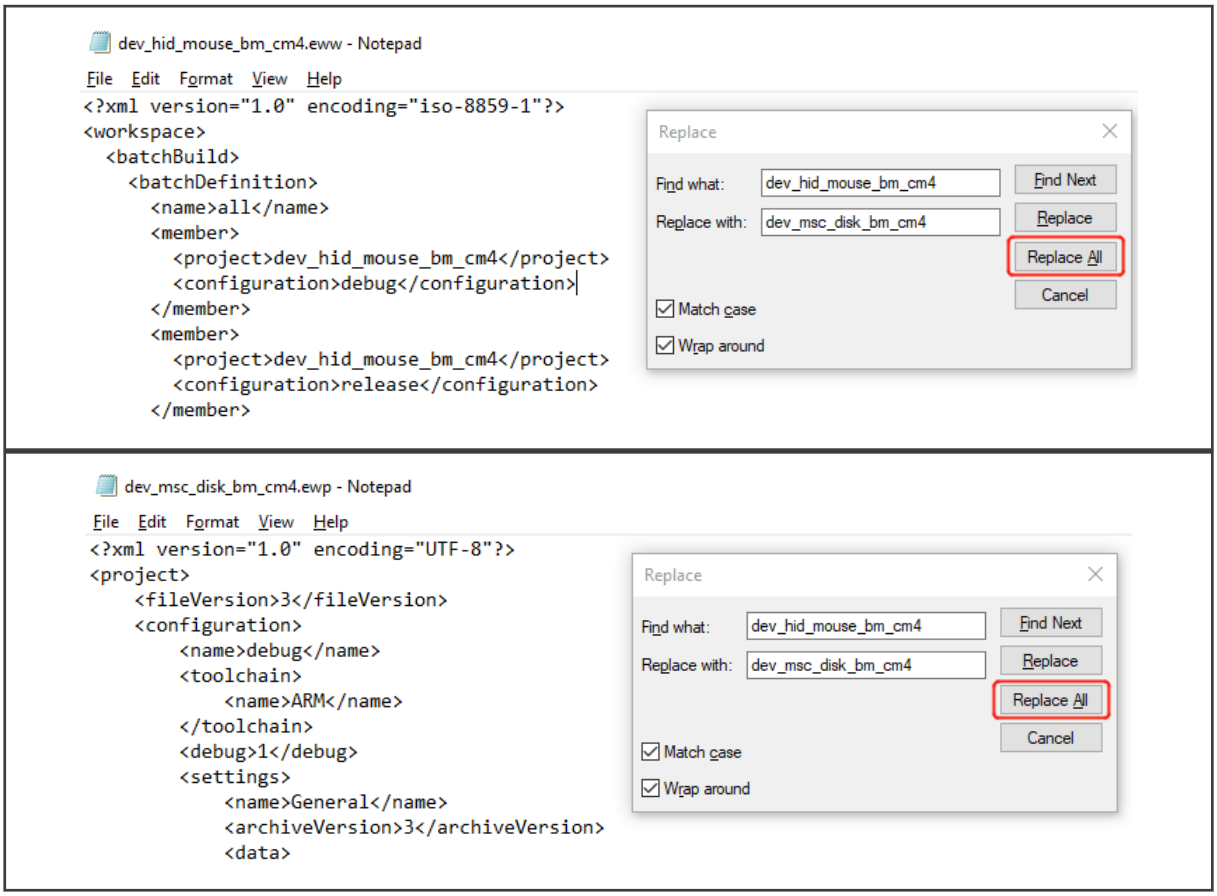
   b. Open the *iar* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/iar*.

   c. Change the file name `dev_hid_mouse_bm_cm4` to `dev_msc_disk_bm_cm4` respectively.



   d. 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.

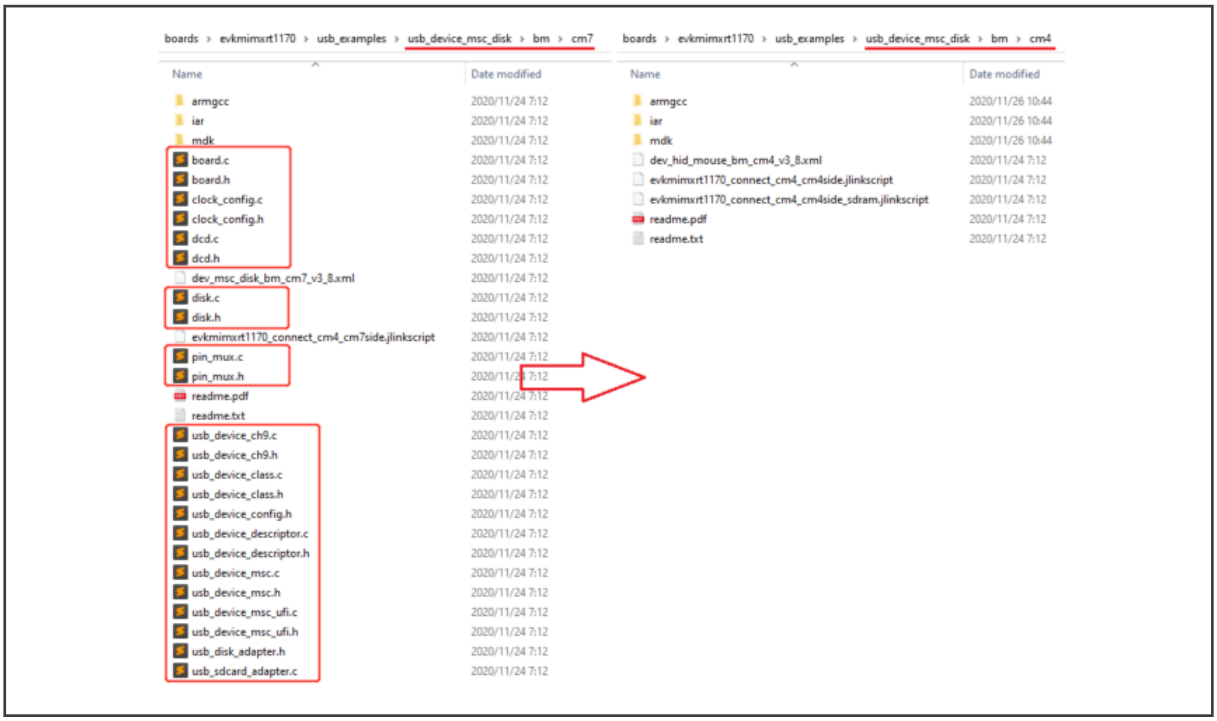
   e. Search and replace all `dev_hid_mouse_bm_cm4` with `dev_msc_disk_bm_cm4`, and then save the files.



2. **Rearrange source files**

   a. 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.

   b. 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*.



3. **Rearrange project files**

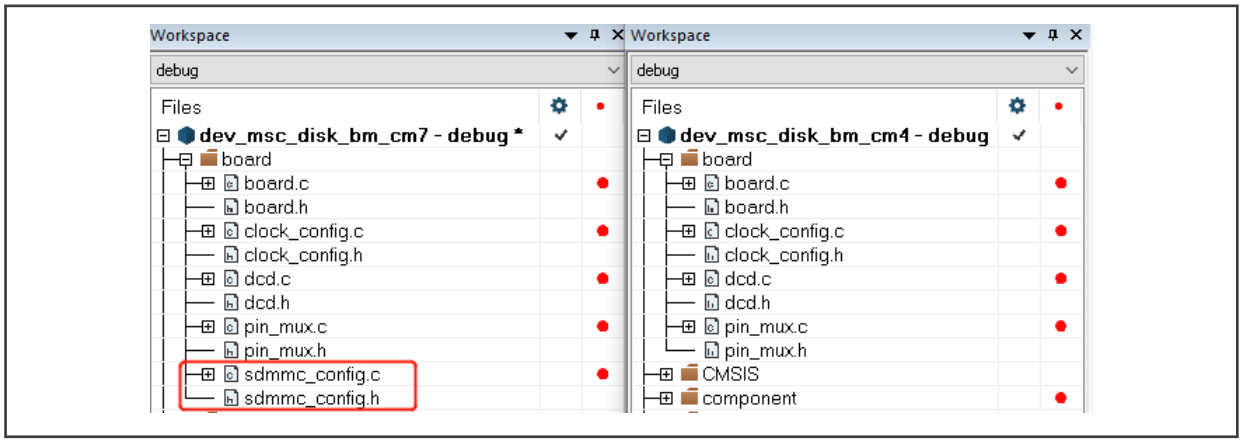
   a. Open `dev_msc_disk_bm_cm7` and `dev_msc_disk_bm_cm4` IAR project respectively. The two workspaces are respectively located under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm7/iar* and *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/iar*.

   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.

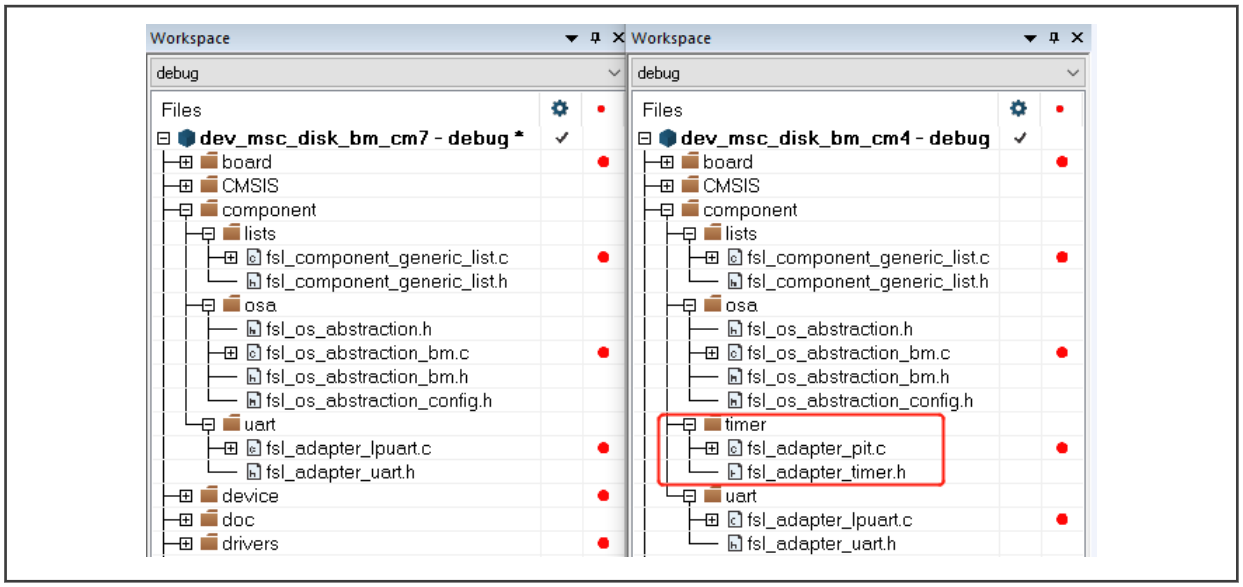
   c. Compare the two groups with the same name. Delete files that are in the `cm4` project but not `cm7`. Find files that are in the `cm7` project but not `cm4` and add them to the `cm4` project.

   In this case,

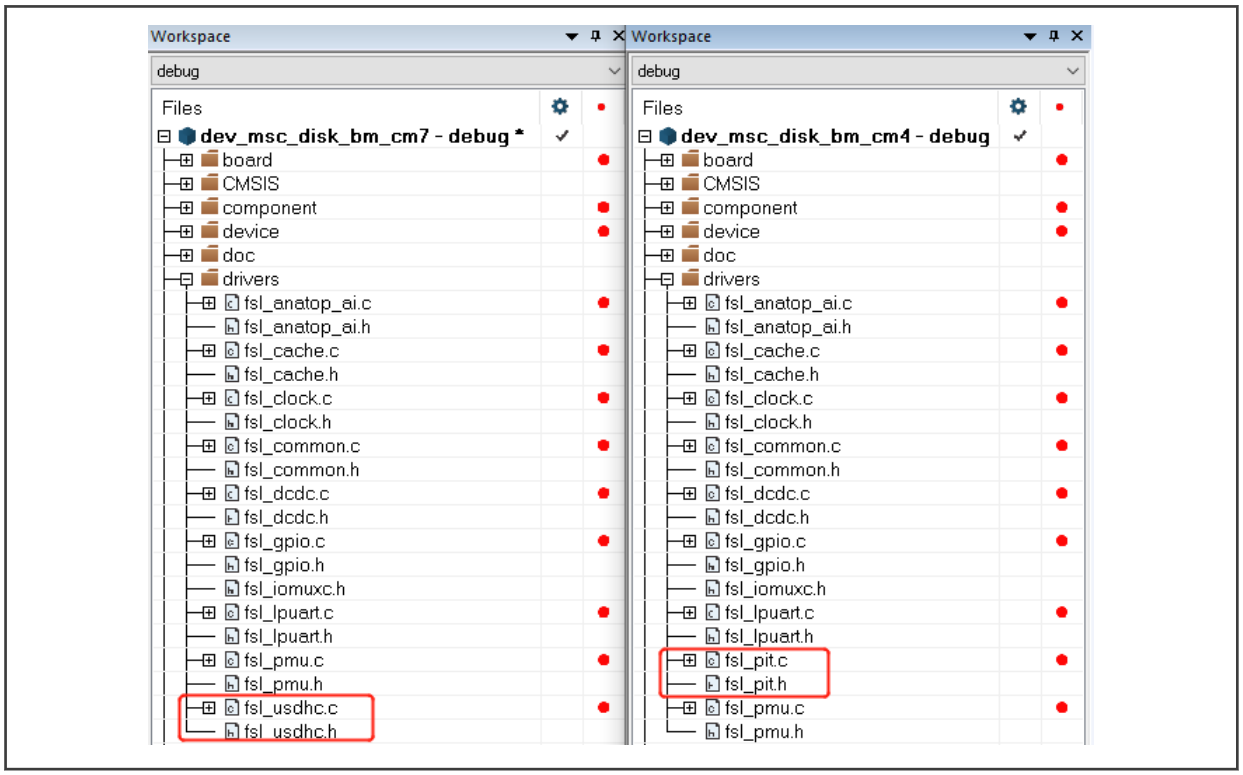
   - For the `board` group, add `sdmmc_config` source files to `board` group in the `cm4` project.



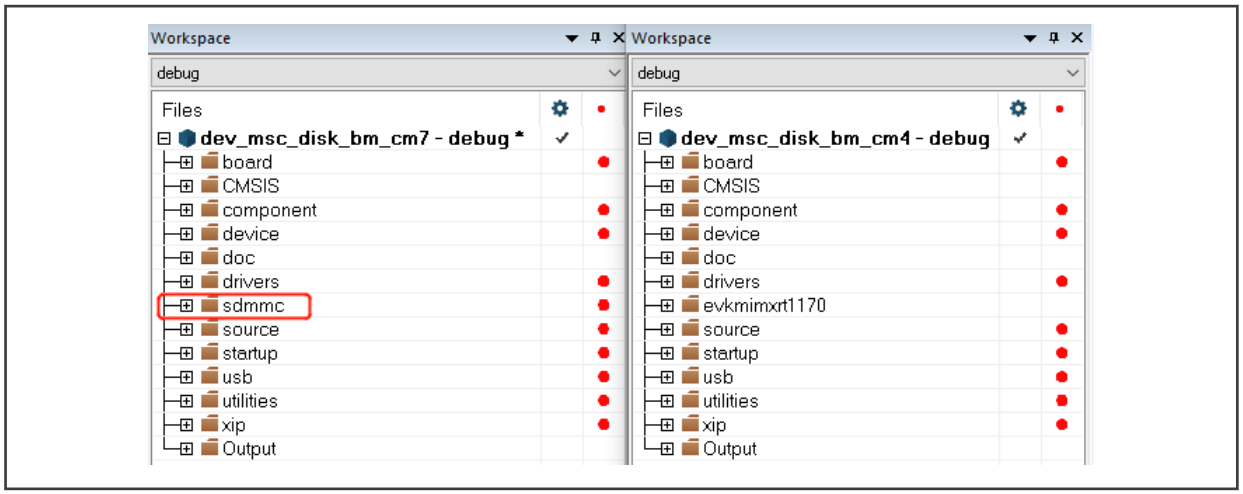
   - For the `component` group, delete files associated with timer from `cm4` project and add no files.



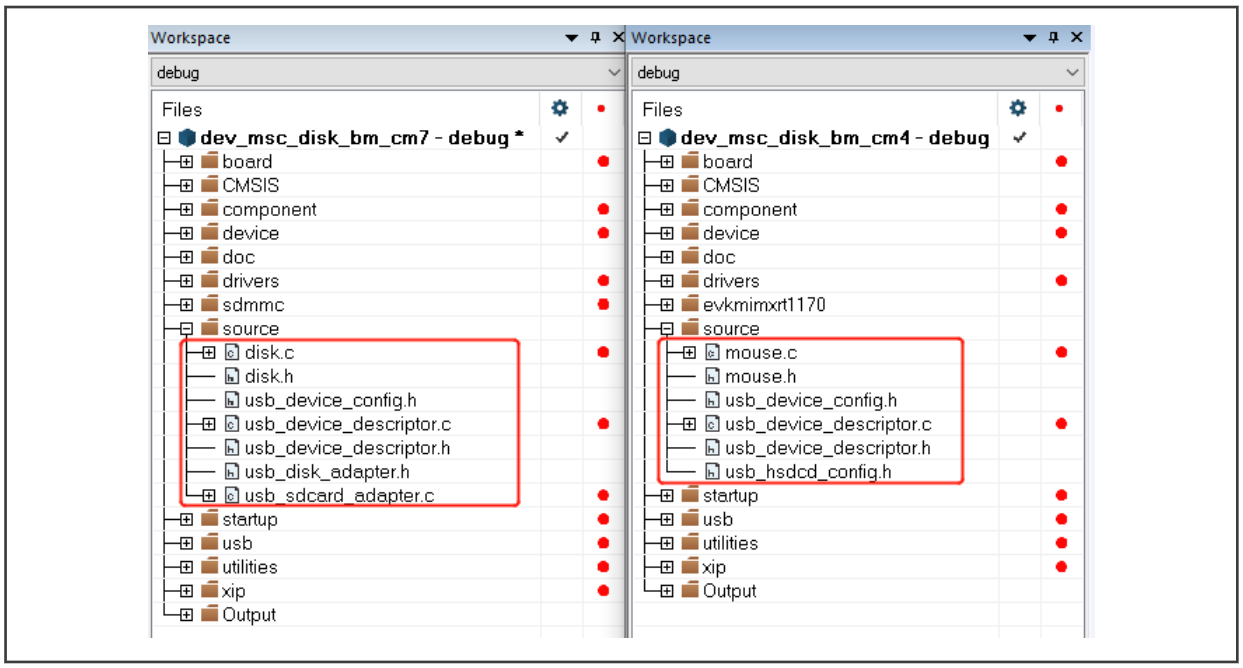
- For the `drivers` group, delete `fsl_pit` source files from the `cm4` project and add `fsl_usdhc` source files to the `cm4` project.



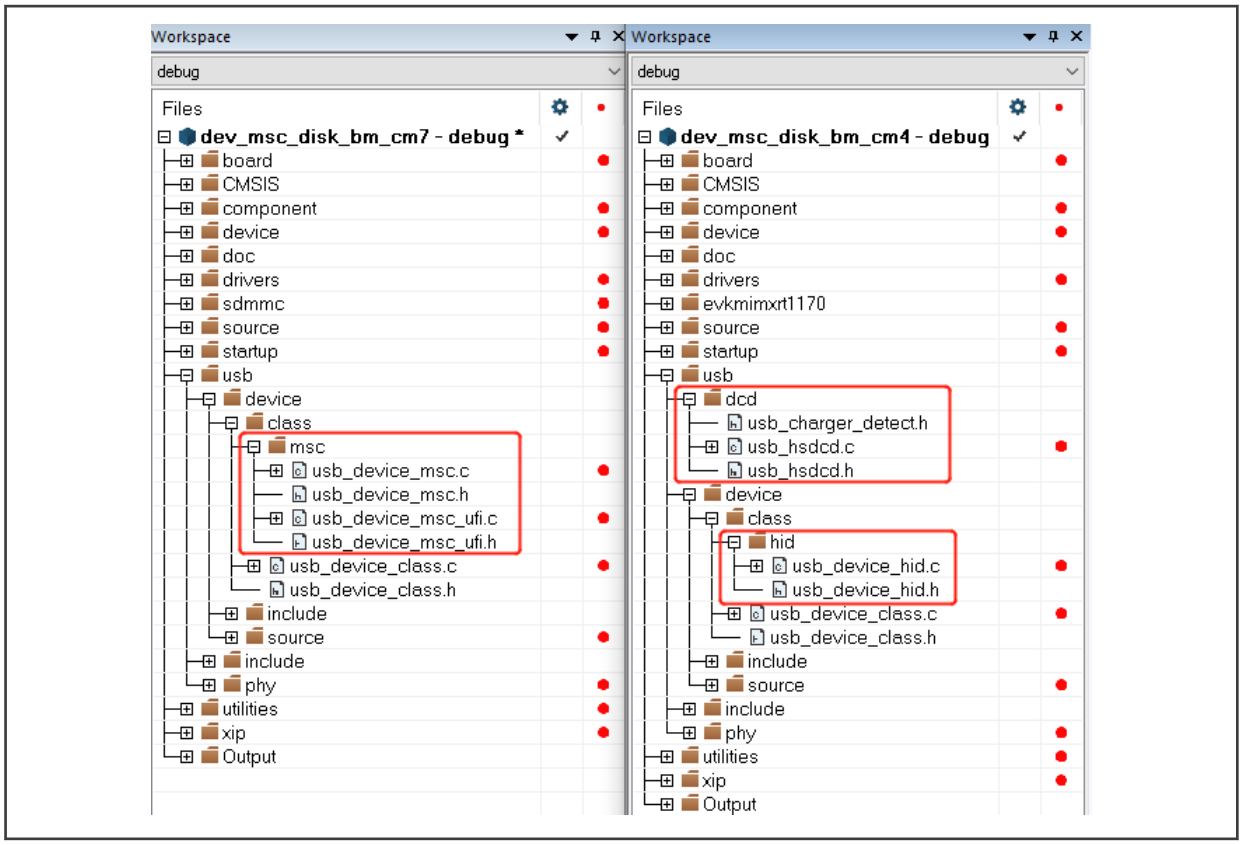
- For the `sdmmc` group, there isn't `sdmmc` group in the `cm4` project. Add `sdmmc` group and its children to the `cm4` project.



- 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.



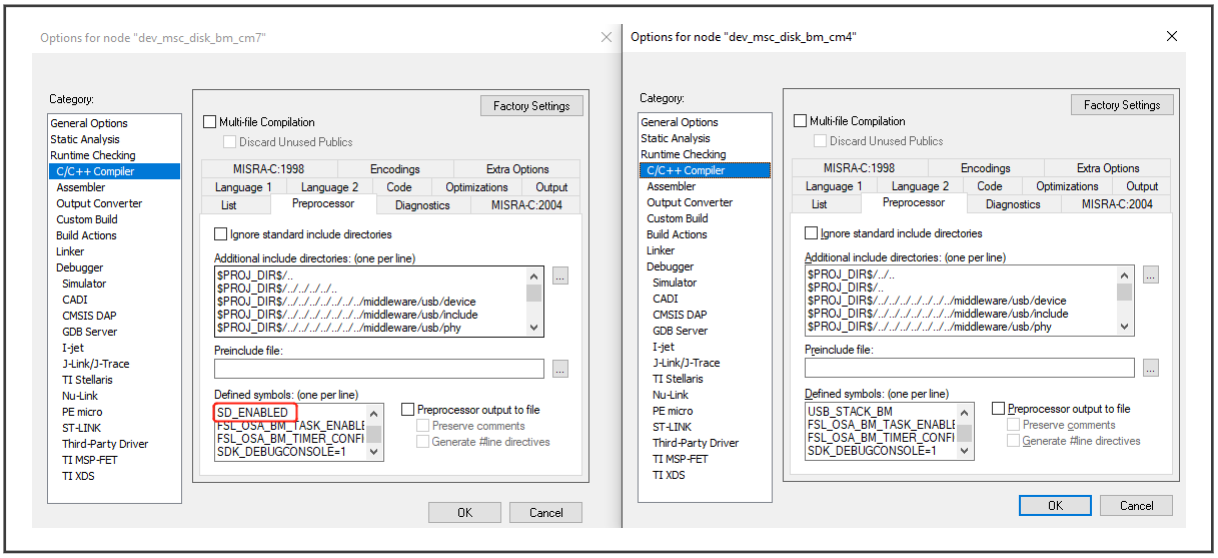
- For the `usb` group, replace the `hid` group and its children in the `cm4` project with `msc` group in the `cm7` project and delete the `dcd` group.



4. **Adjust project settings**

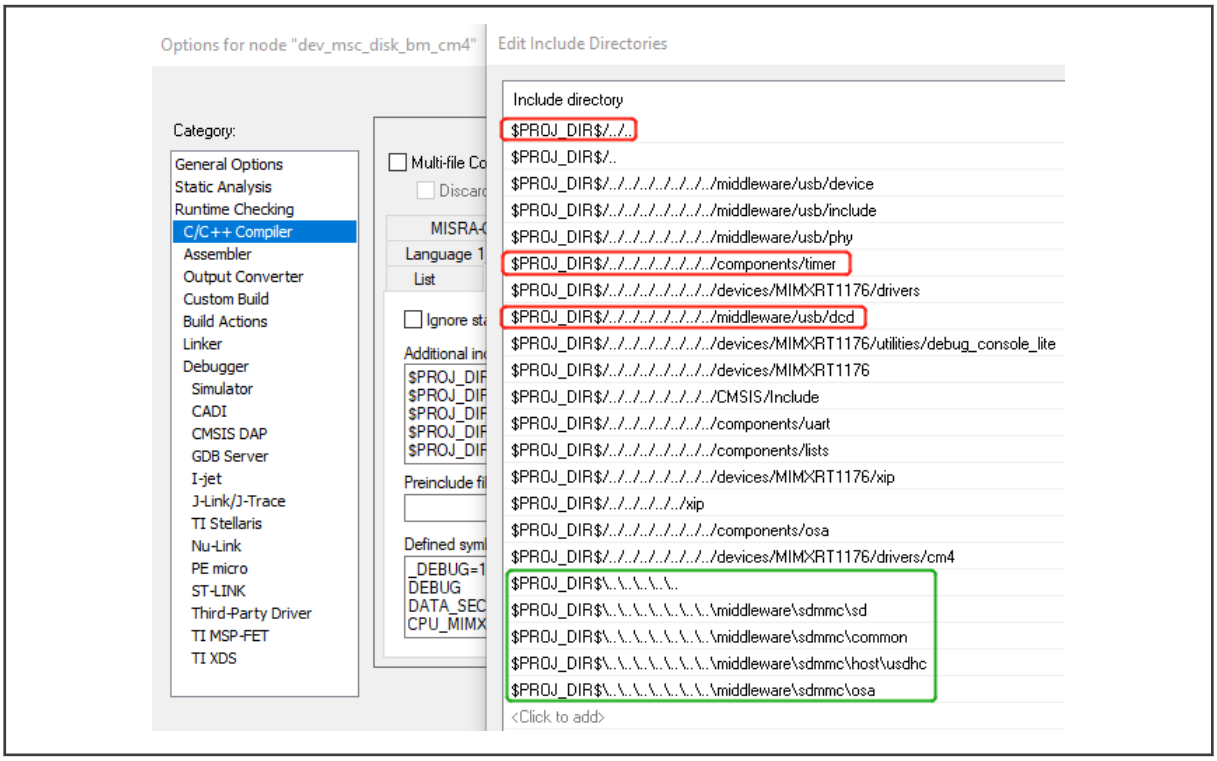
   a. Compare the macro in the project settings: **Option** -> **C/C++ compiler** -> **Preprocessor**. Add the macro that are in the `cm7` project but not `cm4` to the `cm4` project.

   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 those in green.



With all above steps done, the M7 project is successfully changed to an M4 project and the M4 example USB project can be downloaded and debugged.

3 MDK

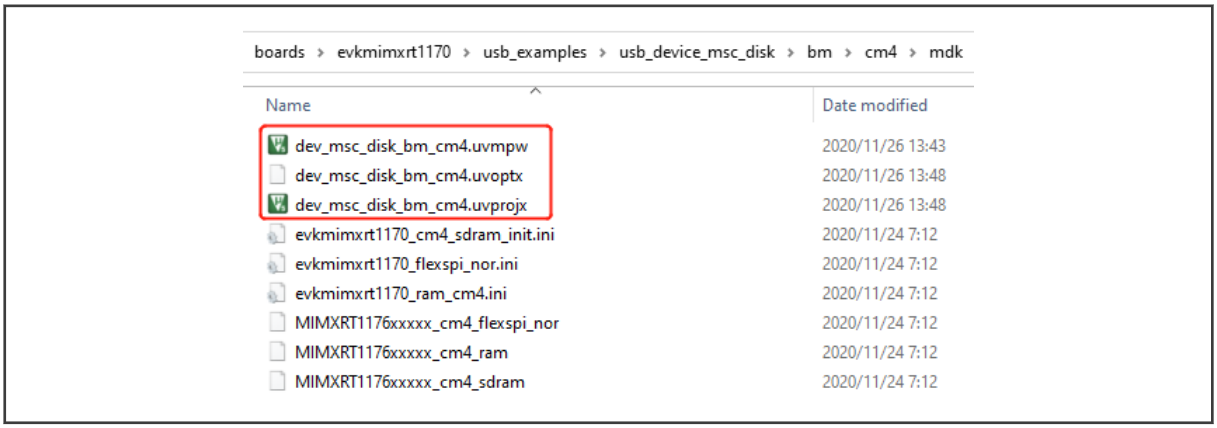
1. **Create an M4 project**

   a. Copy the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_hid_mouse/bm/cm4* to the folder where the example needs to 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*.

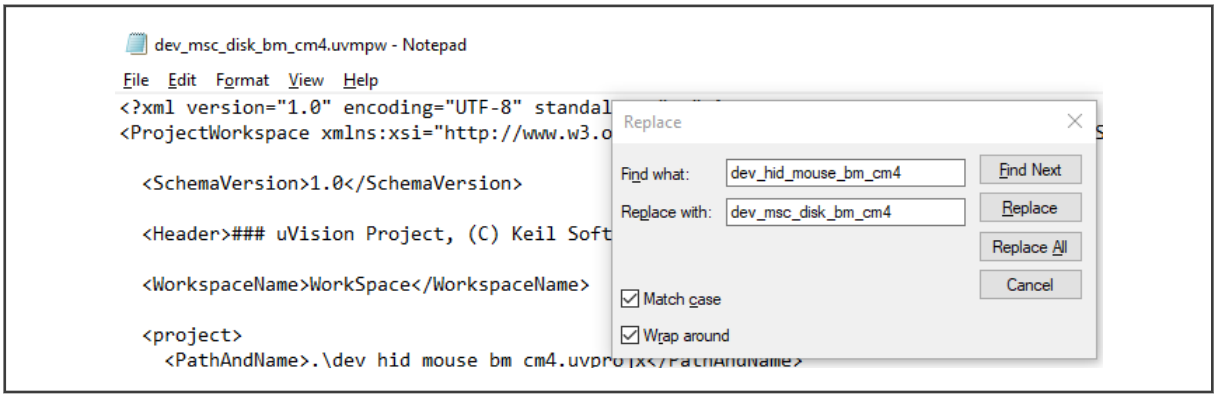
   b. Open the *mdk* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/mdk*.

   c. Change the file name *dev_hid_mouse_bm_cm4* to *dev_msc_disk_bm_cm4* respectively.



d. 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.

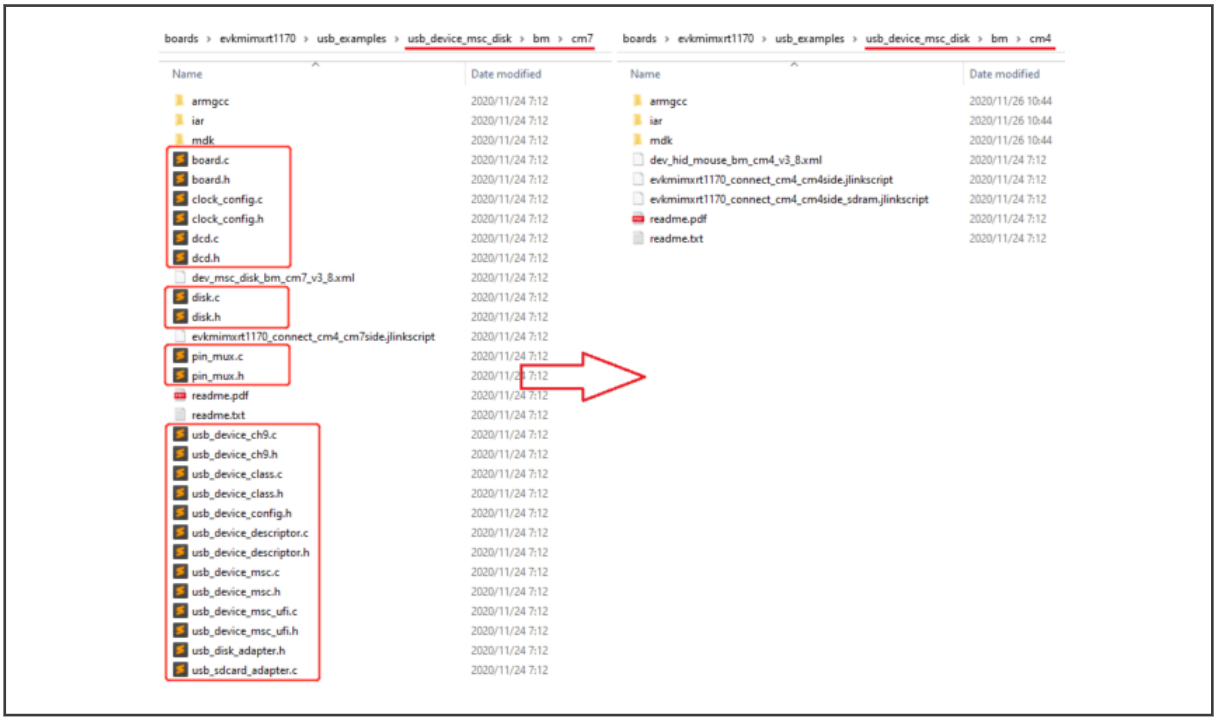
e. Search and replace all `dev_hid_mouse_bm_cm4` with `dev_msc_disk_bm_cm4`, and then save the files.



2. **Rearrange source files**

a. 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.

b. 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*.



3. **Rearrange project files**

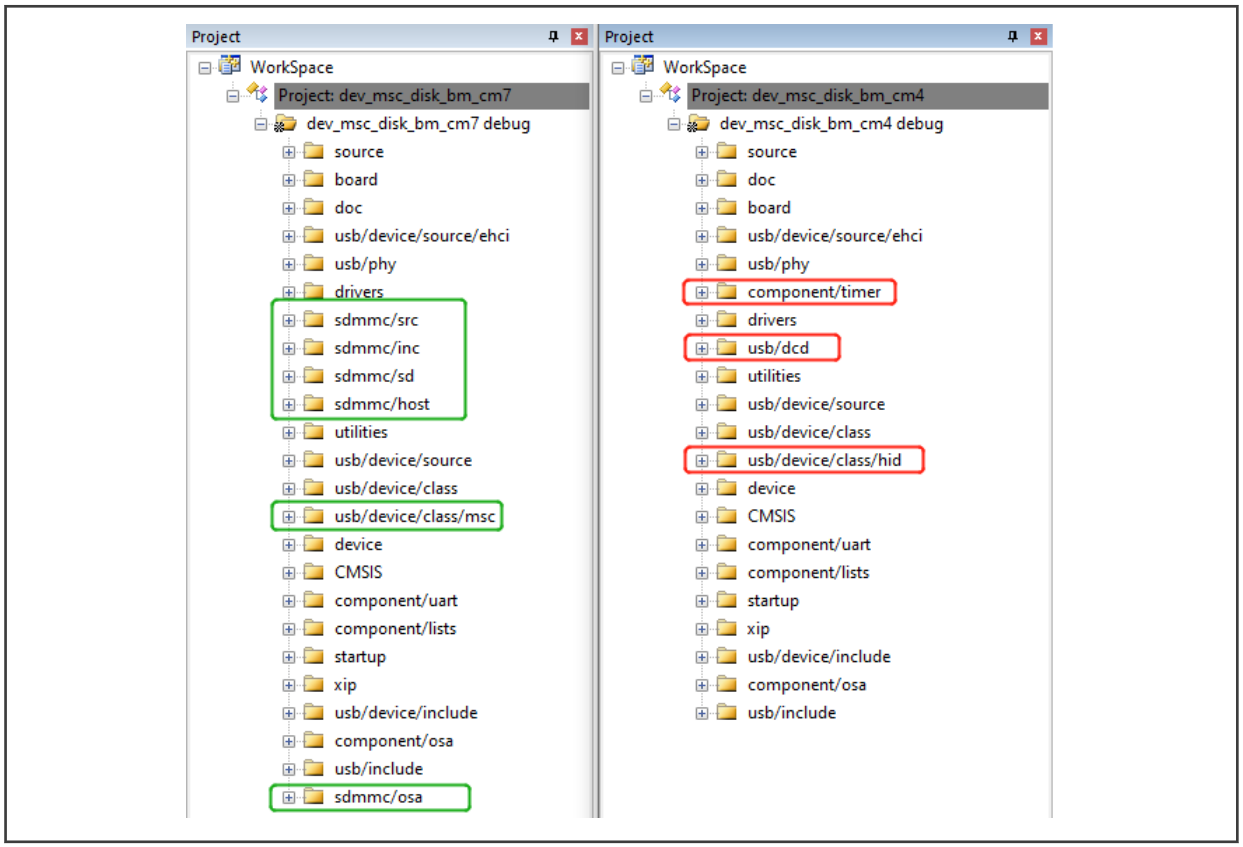
a. Open `dev_msc_disk_bm_cm7` and `dev_msc_disk_bm_cm4` MDK project respectively. The two workspaces are respectively located in *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm7/mdk* and *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/ mdk*.

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`.

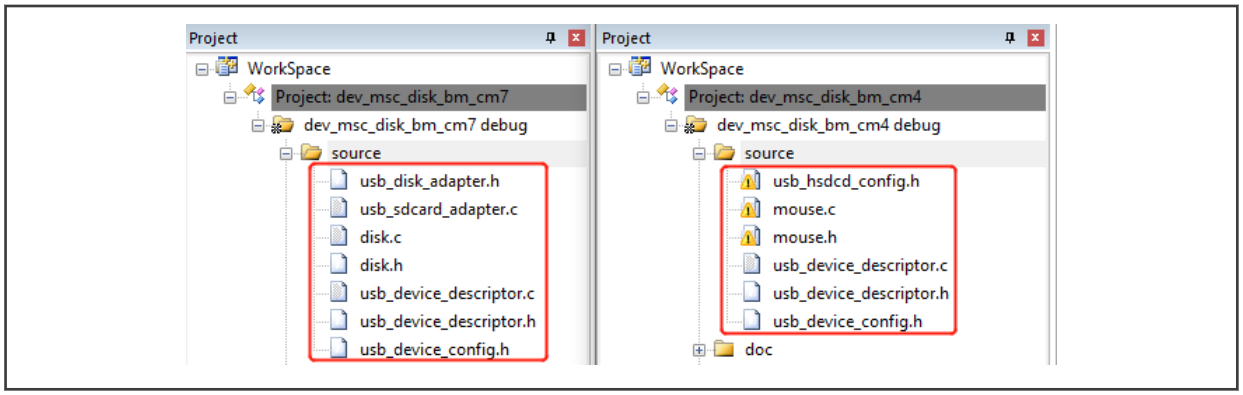
c. Compare the two groups with the same name. Delete files that are in the `cm4` project but not `cm7`. Add files that are in the `cm7` project but not `cm4`.

In this case,

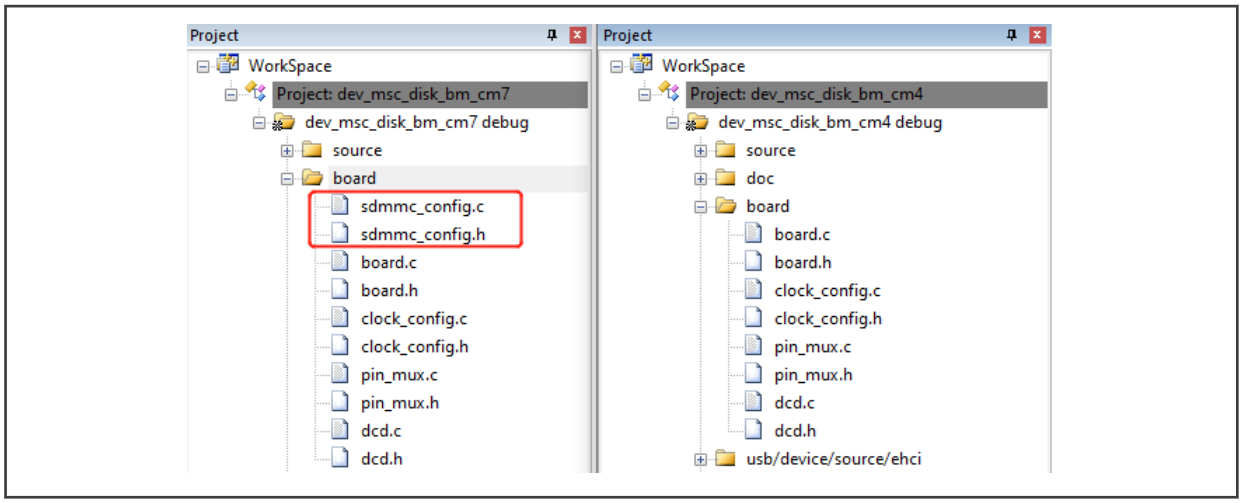
- Delete the groups framed in the red box in the `cm4` project and add those framed in the green box to the `cm4` project.



- 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 those in `cm7` project.



- For the `board` group, add *sdmmc_config* source files to the *board* group in the `cm4` project.

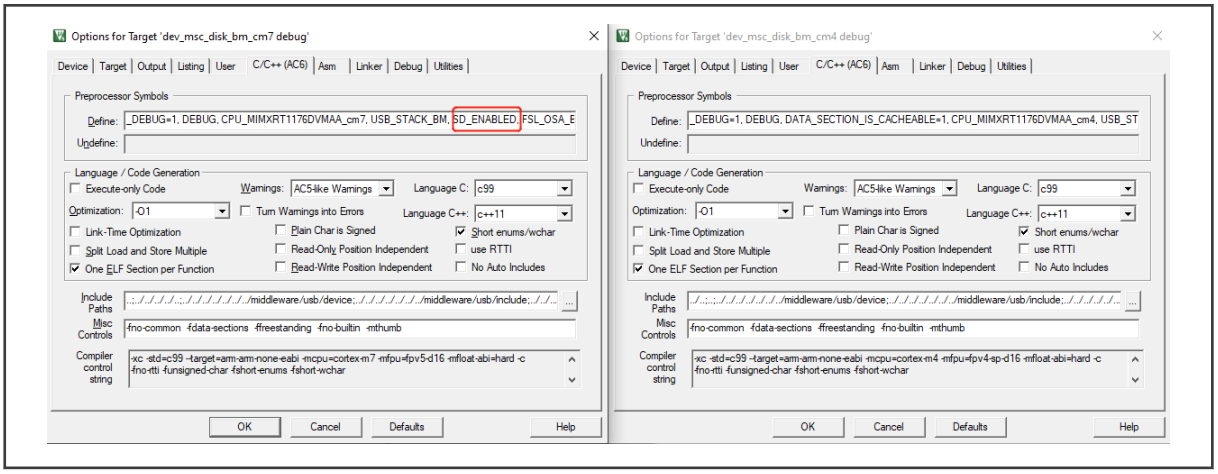


- For the `drivers` group, delete `fsl_pit` source files in the `cm4` project and add `fsl_usdhc` source files to the `cm4` project.

4. **Adjust project settings**

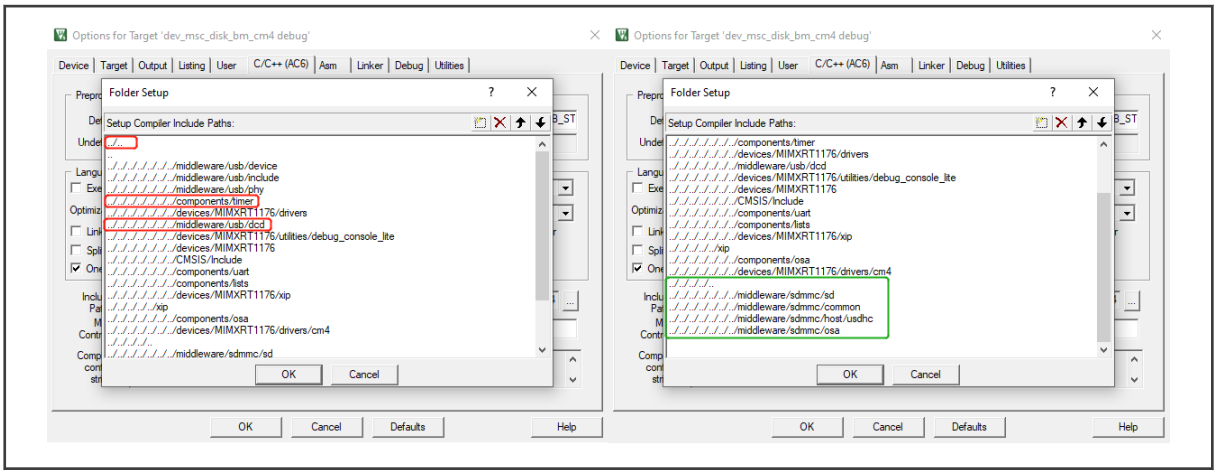
   a. Compare the macro in the project settings by clicking **Option -> C/C++**. Add a macro that is in the `cm7` project but not `cm4`.

   In this case, add the *SD_ENABLED* macro to the `cm4` project.



   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 those framed in the green box.



With all above steps done, the M7 project is successfully changed to an M4 project and the M4 example USB project can be downloaded and debugged.

4 ARMGCC

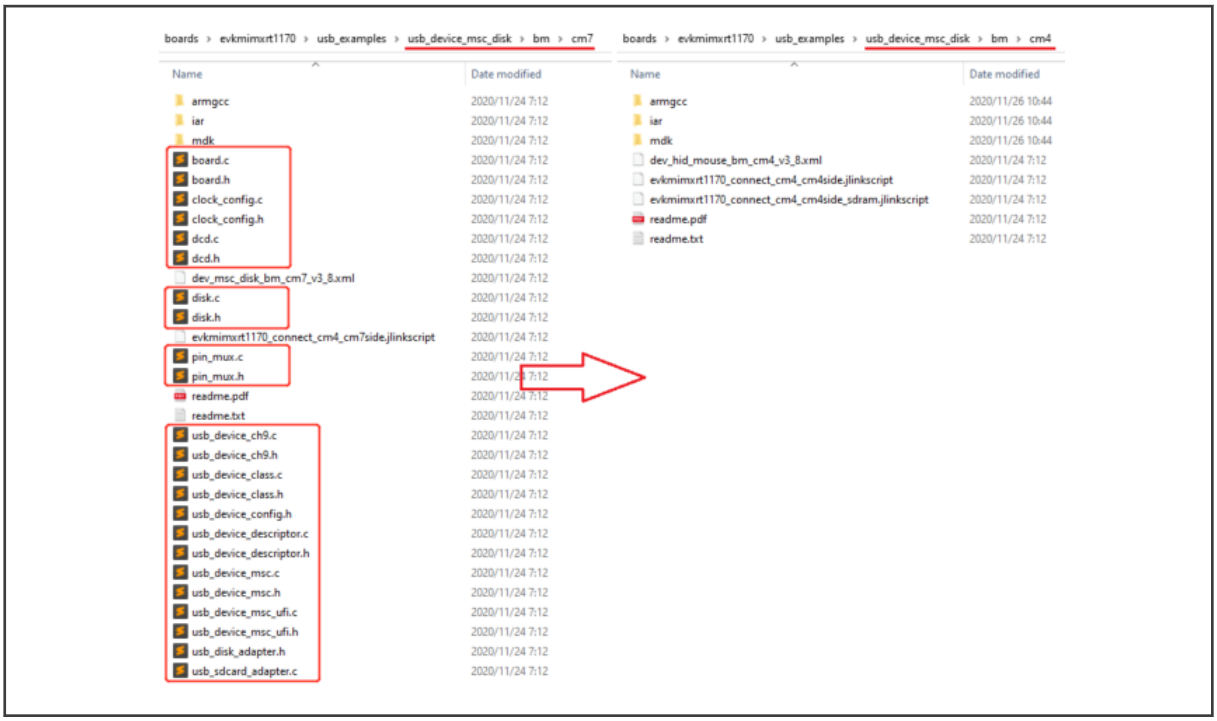
1. **Create an M4 project**
   a. Copy the *cm4* folder under *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_hid_mouse/bm/cm4* to the folder where the example needs to be enabled.

      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*.

   b. 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*.

   c. Search and replace all `dev_hid_mouse_bm_cm4` with `dev_msc_disk_bm_cm4`, and then save the files.
2. **Rearrange source files**
   a. 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.

   b. 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*.

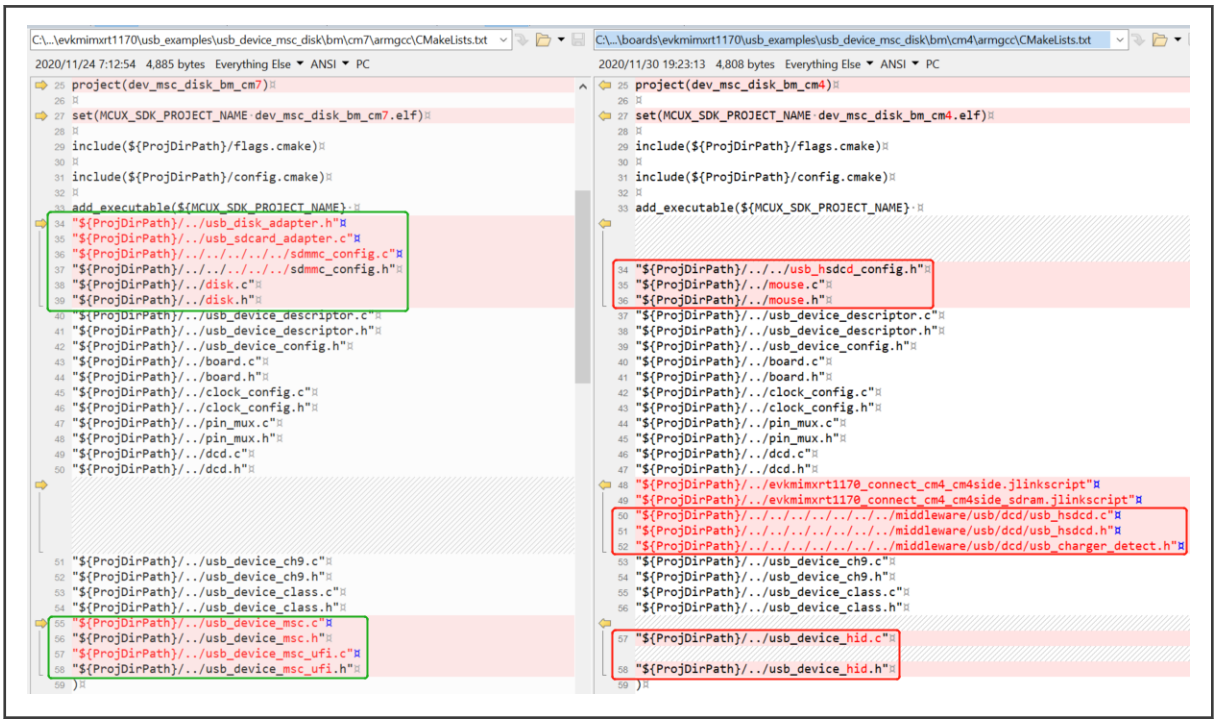


3. **Rearrange project files**

   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 those in the green box to the `cm4` project.



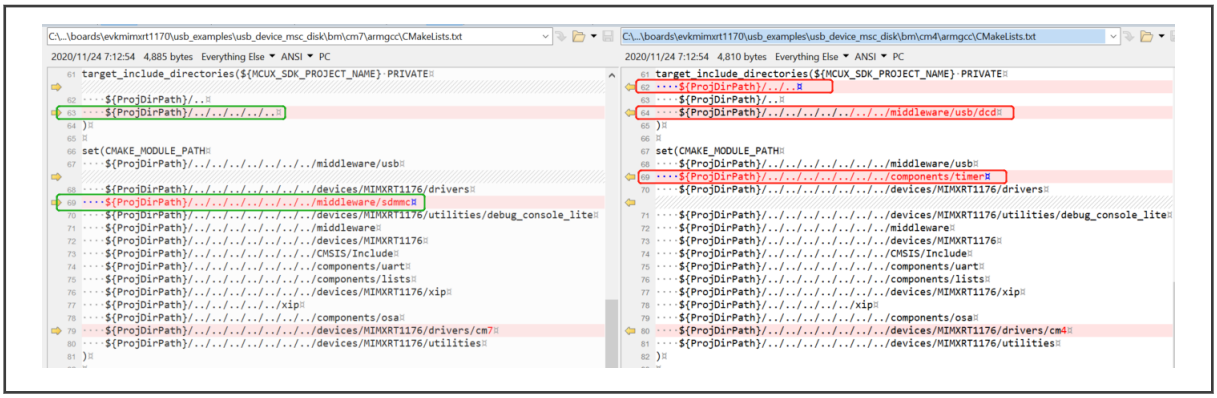
4. **Adjust project settings**

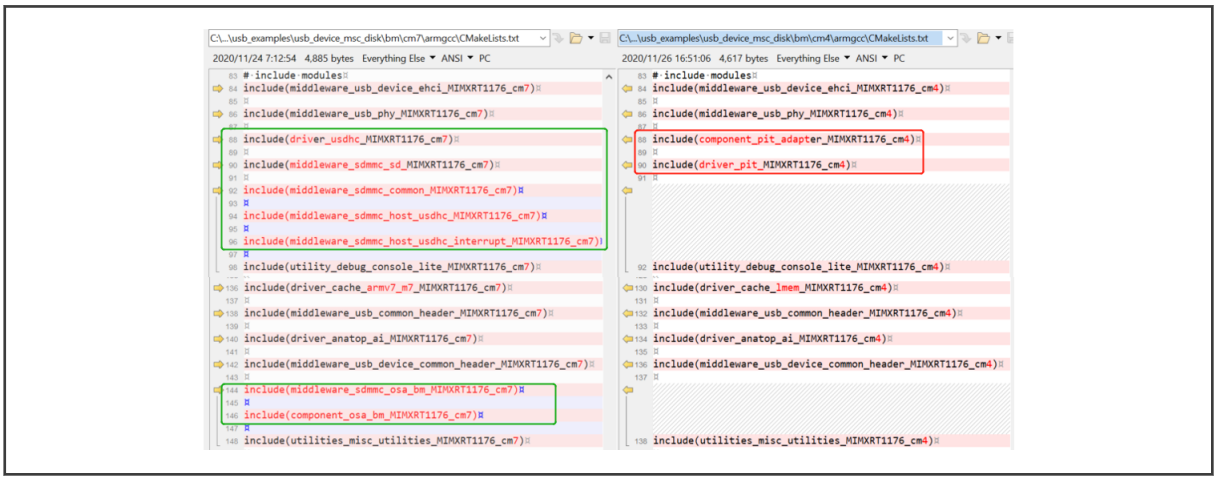
   a. Search the **target_include_directories**, **CMAKE_MODULE_PATH**, and **# include modules** section one by one. Add macros that are in the `cm7` project but not `cm4` to the `cm4` project.

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

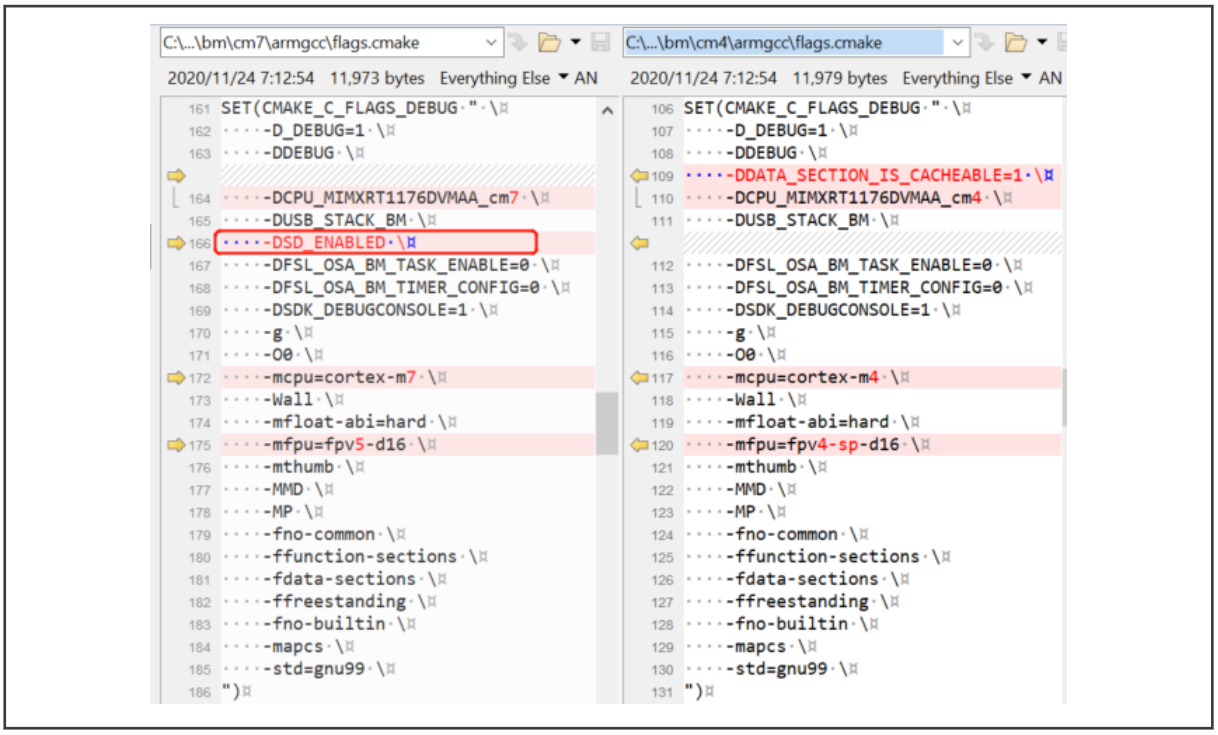




   b. Open the `flags.cmake` of the two examples respectively. The two files are respectively located in *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm7/armgcc* and *<install_dir>/boards/evkmimxrt1170/usb_examples/usb_device_msc_disk/bm/cm4/ armgcc*.

   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.



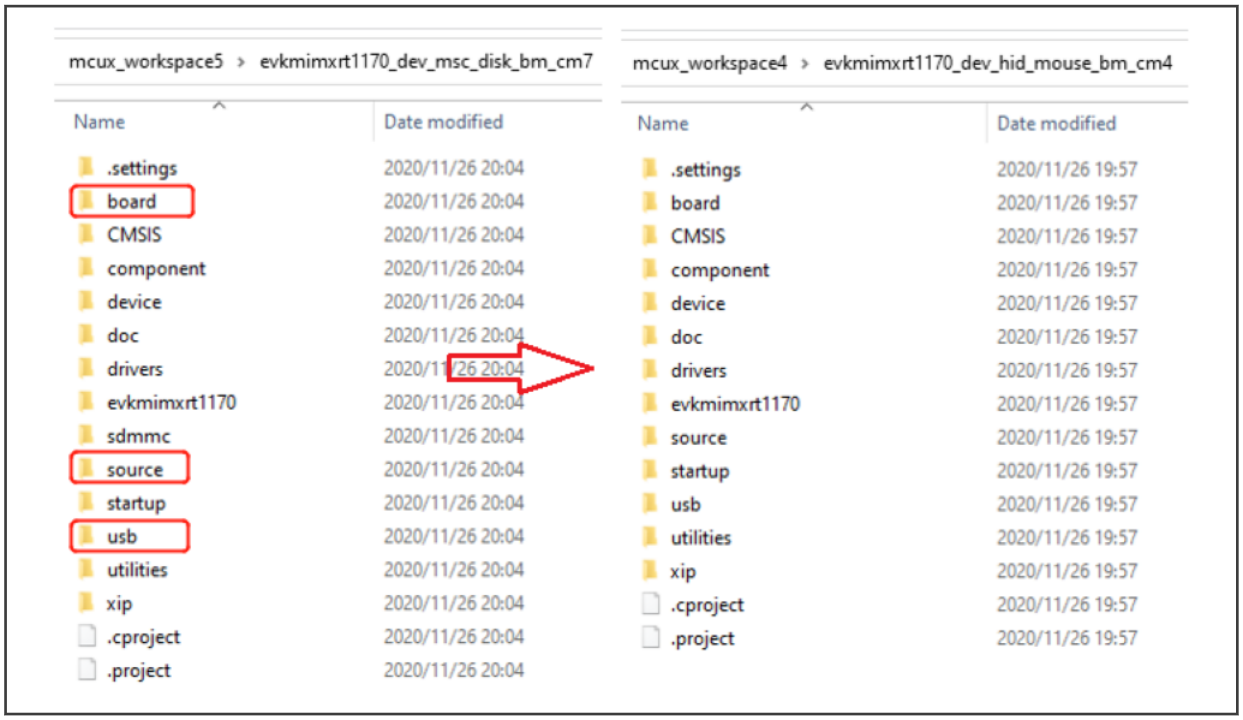
With all above steps done, the M7 project is successfully changed to an M4 project and the M4 example USB project can be downloaded and debugged.

5 MCUXpresso IDE

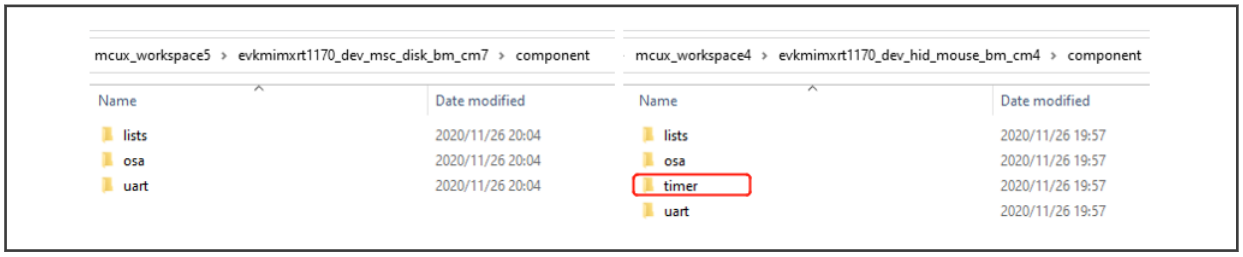
1. **Rearrange source files**
   a. Import `dev_msc_disk_bm_cm7` and `dev_hid_mouse_bm_cm4` example.
   b. Open the directory of the two examples in the file explorer.
   c. Compare the whole project directory and add the folders that are in the `cm7` project but not `cm4` to the `cm4` project.
   d. Compare the two folders. 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.

   In this case,

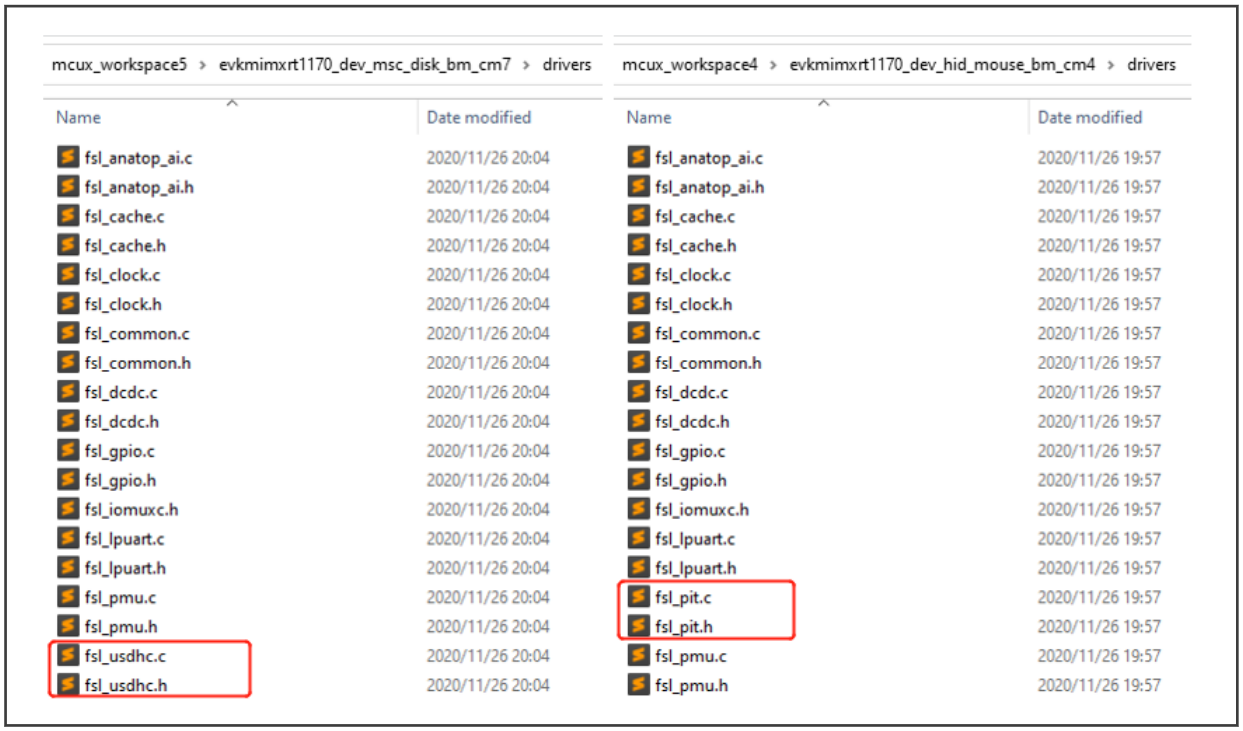
   - For the `board`, `source` and `usb` folders, replace them in the `cm4` project with those in the `cm7` directory.



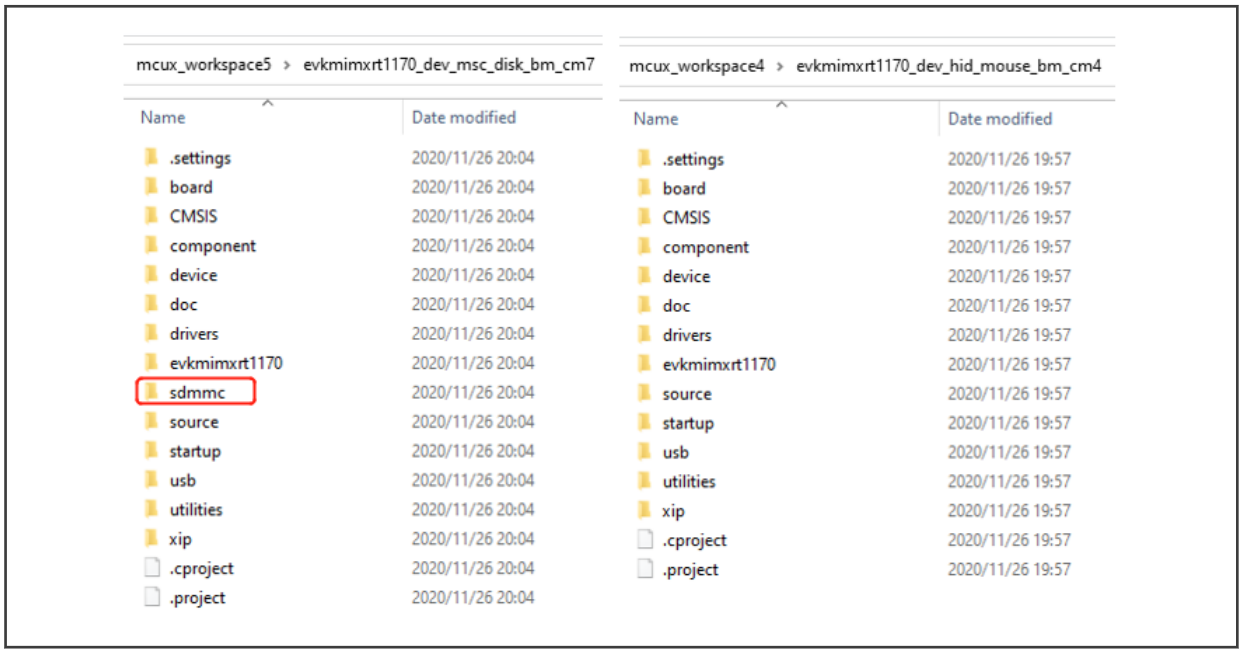
- For the `component` folder, delete the `timer` folder in the `cm4` directory.



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



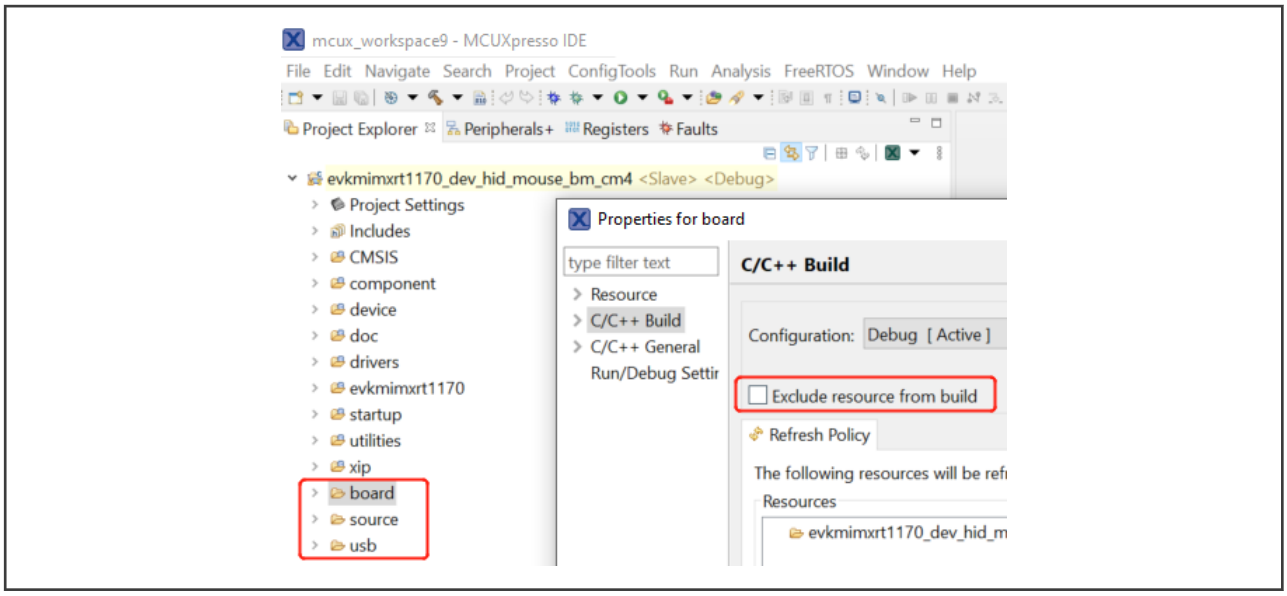
- For the `sdmmc` folder, copy it to the `cm4` project.



2. **Rearrange project files**

   After the source files are updated, the project files are loaded to the `cm4` project automatically. No manual updates to the project files are required.

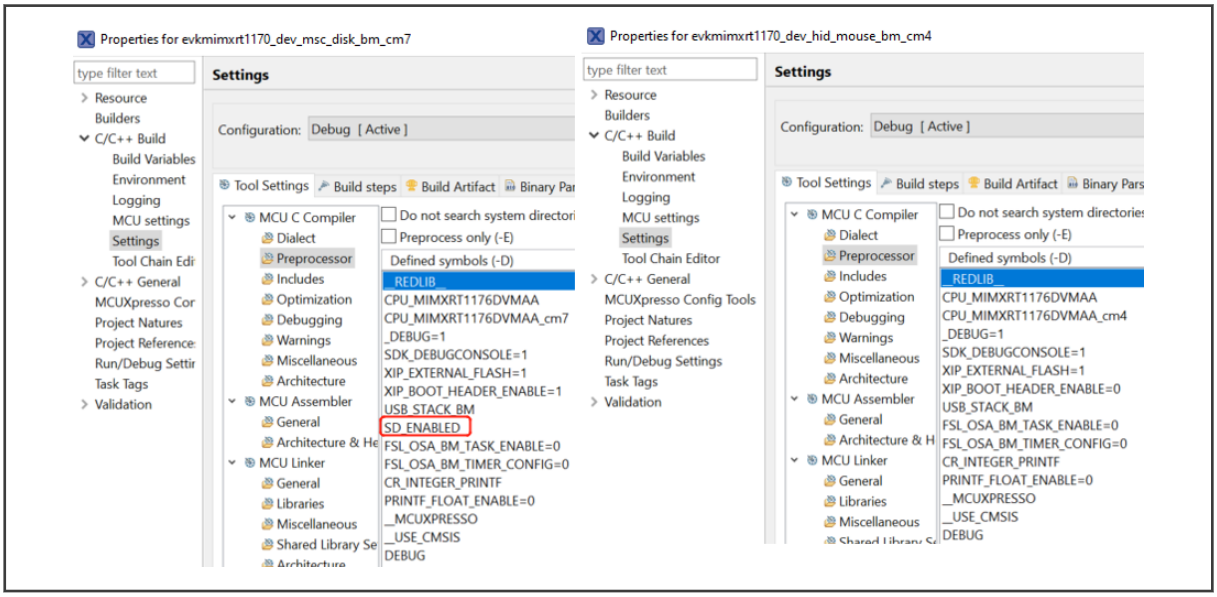
   Due to the deletion in Rearrange source files, for the *board*, *source* and *usb* folders, an option needs to be deleted in the properties. Right click **board** -> **properties** -> **C/C++build** and uncheck **Exclude resource from build**. Perform the same operation in the **source** and **usb** folders.



3. **Adjust project settings**

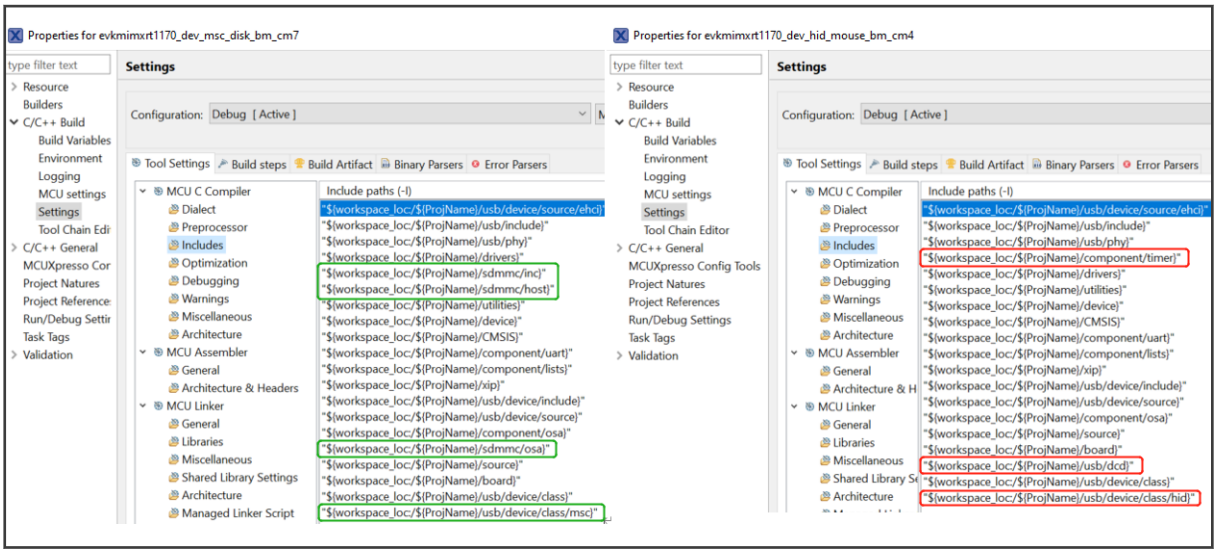
   a. Compare the macro in the project settings, by clicking **Properties** -> **C/C++ Build** -> **Setting** -> **MCU C Compiler** -> **Preprocessor**. Add the macro that is in the `cm7` project but not in `cm4` to the `cm4` project.

   In this case, the macro is `SD_ENABLED`.



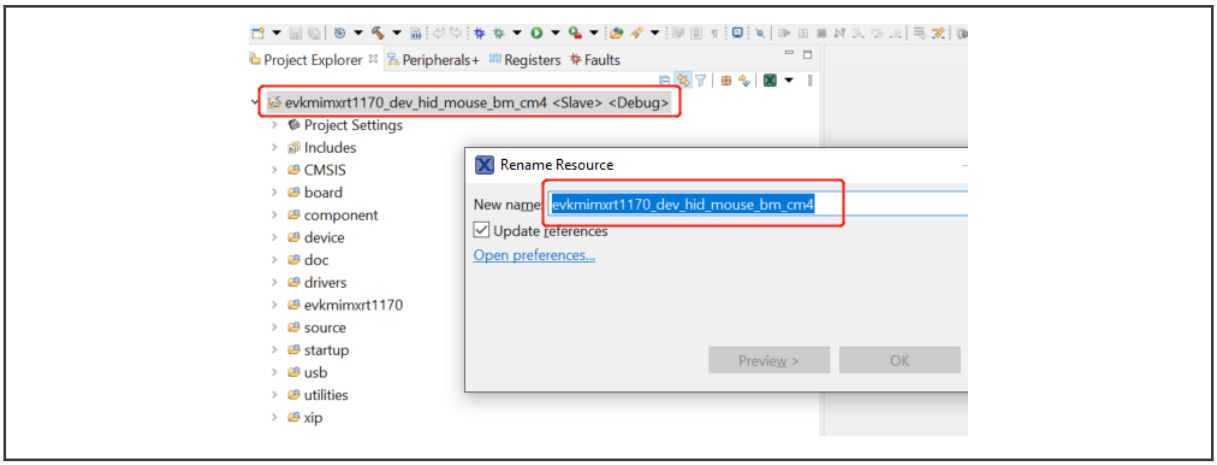
   b. Change the include paths in the project settings, by clicking **Properties** -> **C/C++ Build** -> **Setting** -> **MCU C Compiler** -> **Includes**, according to the changes of project files.

   In this case, delete the paths framed in the red box and add those in the green box.



4. **Change project name**

   a. Right click **project name -> rename**, change the file name `dev_hid_mouse_bm_cm4` to `dev_msc_disk_bm_cm4`, and then click **OK**.



With all above steps done, the M7 project is successfully changed to an M4 project and the M4 example USB project can be downloaded and debugged.

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 M4 core. For other examples, similar way applies to change an M7 project to an M4 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	20 January 2021	Initial release
1	01 June2021	Updated for MCUXpresso SDK v2.10.0

How To Reach Us

Home Page:

nxp.com

Web Support:

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.

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX,EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2021. **All rights reserved.**

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 01 June 2021

Document identifier: AN13114

arm