AN12671

Steps to migrate from M4 USB project to M0p USB example project for K32

Rev. 1 — 11 July 2022 Application note

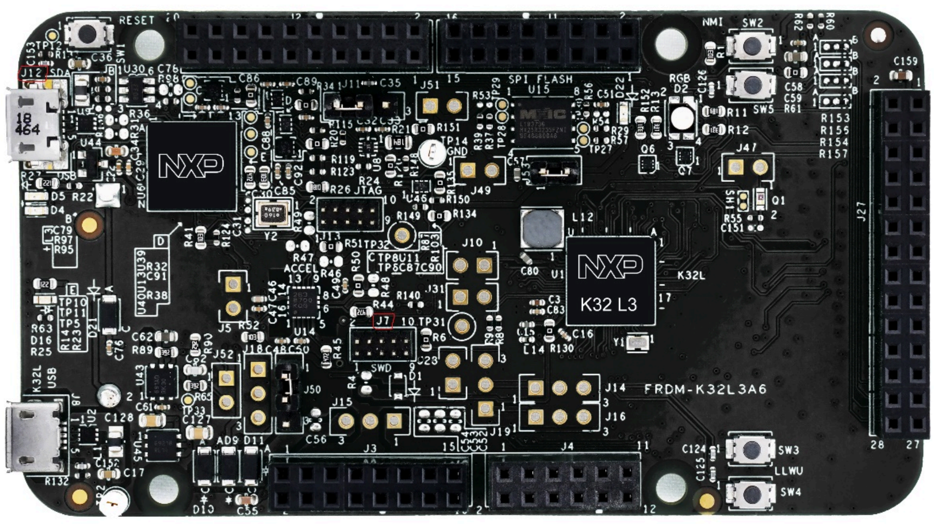
Document information

Information	Content
Keywords	AN12671, M4 USB project, M0p USB example project, K32
Abstract	This document lists the steps to migrate from M4 USB project to M0p USB example project for K32.

1 Introduction

MCU boots from M4 core, by default. However, it can be configured to boot from M0+ core. To do so, bit field BOOT_CORE in FTFE_FOPT register must be set to 0 (register value - 0xFFFFFFBF).

Follow the steps below to access (program, read, erase) the FTFE_FOPT register using J-Link commander:



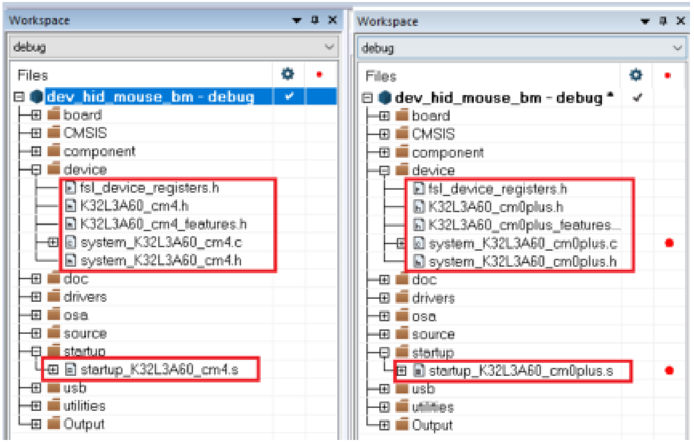
1. Connect J12 on board to PC using USB cable.
2. Connect J7 on board to PC using Jlink.
3. Open Jlink command. Once the following instructions are input, new configuration will become active after power-on-reset.
   FTFE_FOPT value should be 0xFFFFFFFF before.

```
w1 0x40023007 0x43 // FCCOB0: CMD_PROGRAM_ONCE (IFR)
w1 0x40023006 0x84 // FCCOB1: IFR Index of the FOPT
w1 0x40023005 0x00 // FCCOB2: Not used
w1 0x40023004 0x00 // FCCOB3: Not used
w1 0x4002300B 0xFF // FCCOB4: Record byte 0 value --> FOPT (Bit31:Bit24)
w1 0x4002300A 0xFF // FCCOB5: Record byte 1 value --> FOPT (Bit23:Bit16)
w1 0x40023009 0xFF // FCCOB6: Record byte 2 value --> FOPT (Bit15:Bit8)
w1 0x40023008 0xBF // FCCOB7: Record byte 3 value --> FOPT (Bit7:Bit0)
w1 0x40023000 0x80 // Trigger operation
```

4. Additional information is available on webpage https://wiki.segger.com/K32W#Boot_ROM. For download issue, see the “Getting Started with MCUXpresso SDK for K32W0X2S.pdf”.
   K32 has M4 and M0+ cores. USB example can run on both cores. The USB example in the release package runs on M4 core. However, this example project can be migrated from M4 to M0p core. M4 and M0p USB project use same source files with different project setting. Use the following steps to change M4 project to M0P project with IAR, MDK, and GCC compilers. Other compilers have similar way to migrate project from M4 to M0p.

2 IAR

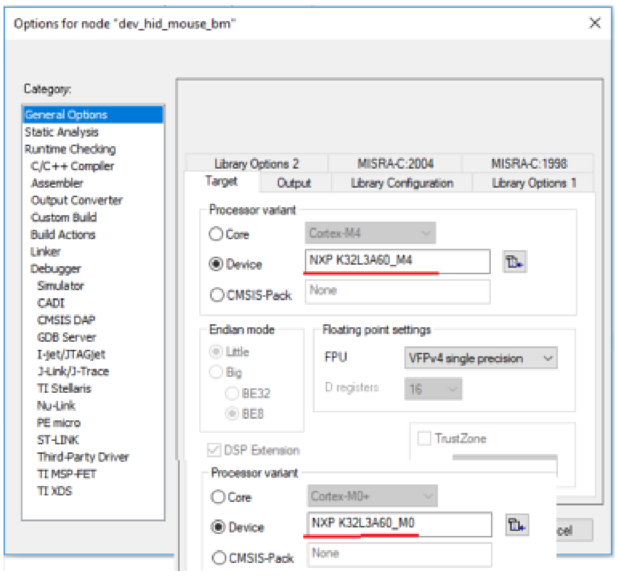
1. Update startup and system file from M4 platform files to M0p platform files. The files are available in folder: FRDM-K32L36\devices\K32L3A60. The bellow picture shows how to change m4 project files to M0p project file.



```
Workspace
debug
Files
dev_hid_mouse_bm - debug
  board
  CMSIS
  component
  device
    fsl_device_registers.h
    K32L3A60_cm4.h
    K32L3A60_cm4_features.h
    system_K32L3A60_cm4.c
    system_K32L3A60_cm4.h
  doc
  drivers
  osa
  source
  startup
    startup_K32L3A60_cm4.s
  usb
  utilities
  Output
```

```
Workspace
debug
Files
dev_hid_mouse_bm - debug *
  board
  CMSIS
  component
  device
    fsl_device_registers.h
    K32L3A60_cm0plus.h
    K32L3A60_cm0plus_features...
    system_K32L3A60_cm0plus.c
    system_K32L3A60_cm0plus.h
  doc
  drivers
  osa
  source
  startup
    startup_K32L3A60_cm0plus.s
  usb
  utilities
  Output
```

2. Open M4 project in IAR. Right click, in option-->General Options-->target and update the CPU type form M4 to M0p.



Options for node "dev_hid_mouse_bm"

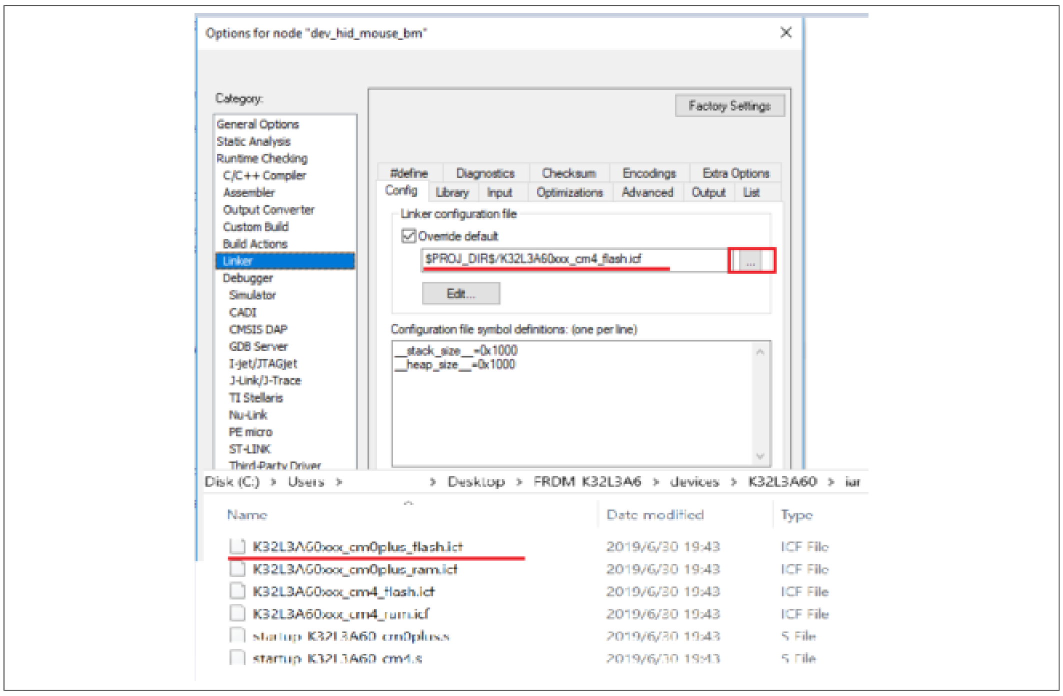
Category: General Options

Target — Processor variant: Device: NXP K32L3A60_M4 (Core: Cortex-M4; CMSIS-Pack: None)

Floating point settings: FPU: VFPv4 single precision; D registers: 16

Processor variant: Device: NXP K32L3A60_M0 (Core: Cortex-M0+; CMSIS-Pack: None)

3. Again, in M4 project in IAR, in option-->linker-->config, change the linker configure file form M4 link file to M0p link file. The linker file path is devices\ K32L3A60 folder.



Linker configuration file

☑ Override default

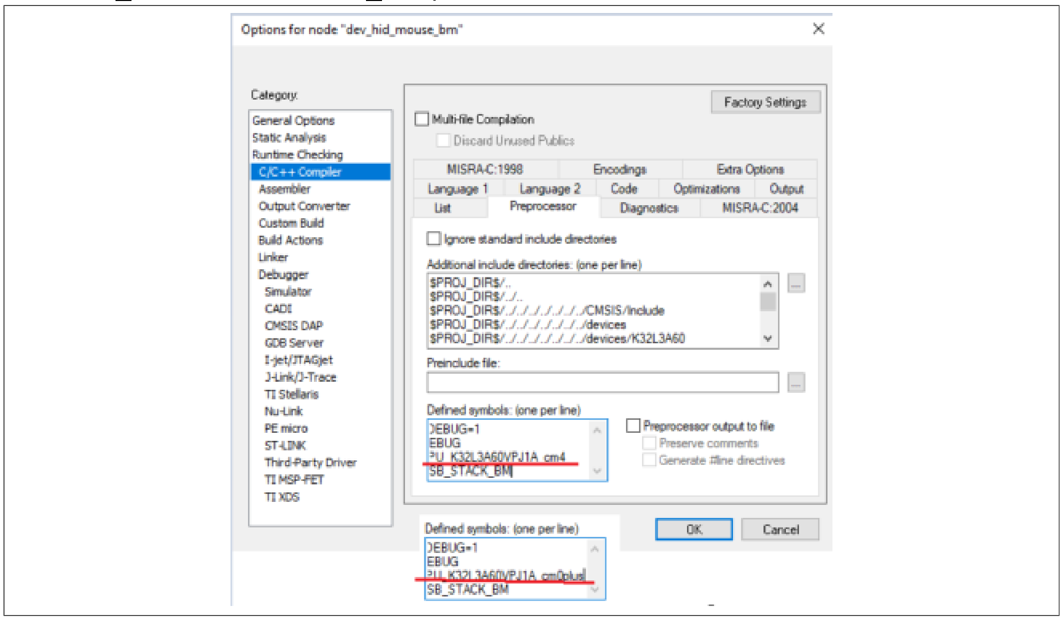
$PROJ_DIR$/K32L3A60xxx_cm4_flash.icf

Configuration file symbol definitions: (one per line)

```
__stack_size__=0x1000
__heap_size__=0x1000
```

Name	Date modified	Type
K32L3A60xxx_cm0plus_flash.icf	2019/6/30 19:43	ICF File
K32L3A60xxx_cm0plus_ram.icf	2019/6/30 19:43	ICF File
K32L3A60xxx_cm4_flash.icf	2019/6/30 19:43	ICF File
K32L3A60xxx_cm4_ram.icf	2019/6/30 19:43	ICF File
startup_K32L3A60_cm0plus.s	2019/6/30 19:43	S File
startup_K32L3A60_cm4.s	2019/6/30 19:43	S File

4. In M4 project in IAR, in option-->c/c++ compiler-->Preprocessor, change the CPU MACRO from "CPU_K32W042S1M2VPJ_cm4" to "CPU_K32W042S1M2VPJ_cm0plus".



☐ Multi-file Compilation

☐ Discard Unused Publics

☐ Ignore standard include directories

Additional include directories: (one per line)

```
$PROJ_DIR$/..
$PROJ_DIR$/../..
$PROJ_DIR$/../../../../../../../CMSIS/Include
$PROJ_DIR$/../../../../../../../devices
$PROJ_DIR$/../../../../../../../devices/K32L3A60
```

Preinclude file:

Defined symbols: (one per line)

```
DEBUG=1
EBUG
PU_K32L3A60VPJ1A_cm4
SB_STACK_BM
```

☐ Preprocessor output to file

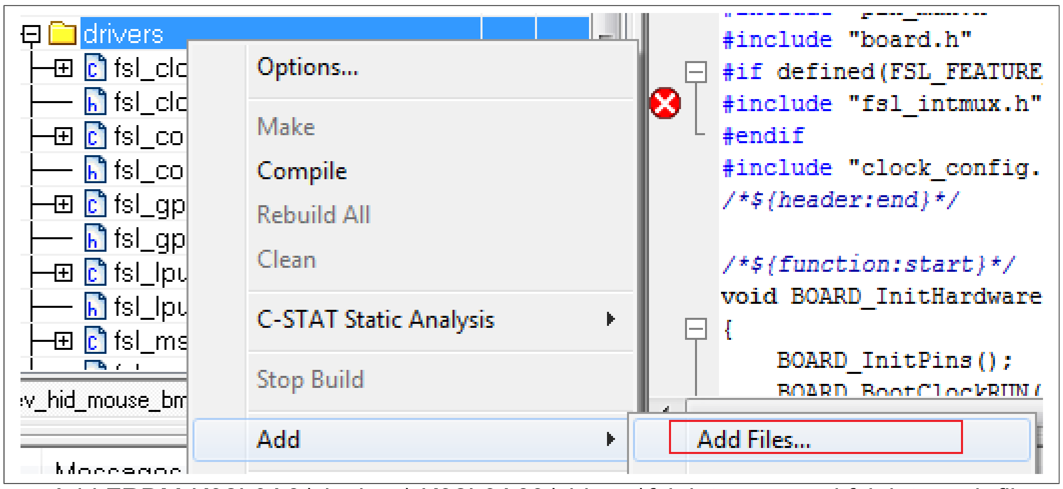
☐ Preserve comments

☐ Generate #line directives

Defined symbols: (one per line)

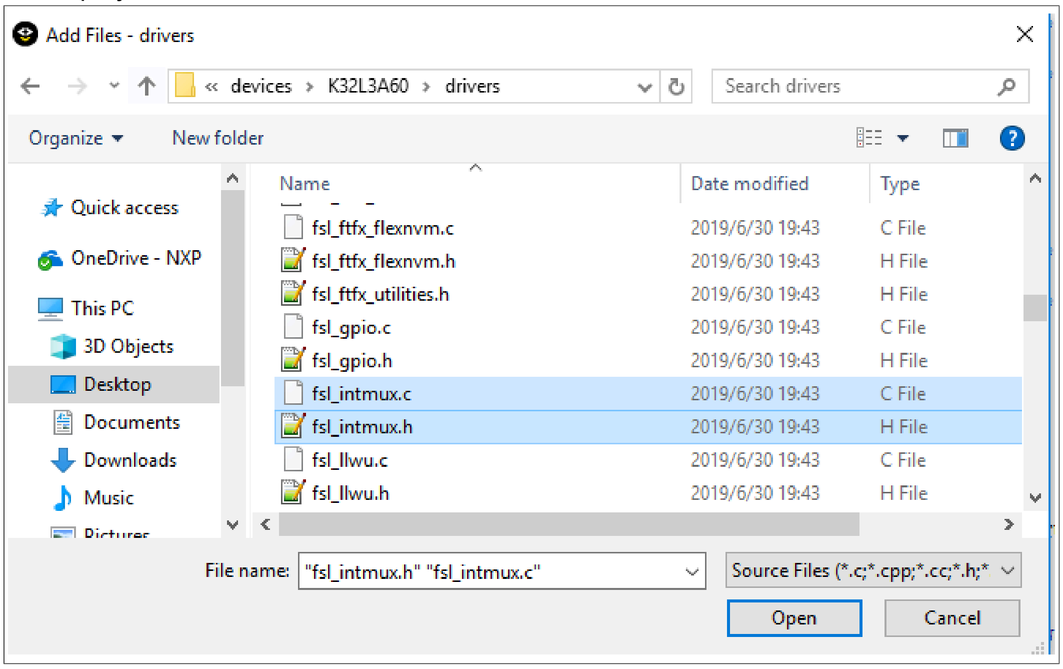
```
DEBUG=1
EBUG
PU_K32L3A60VPJ1A_cm0plus
SB_STACK_BM
```

5. Add int-mux file to M0p project, driver > add > add files, as shown in the figure below



```
#include "board.h"
#if defined(FSL_FEATURE
#include "fsl_intmux.h"
#endif
#include "clock_config.
/*${header:end}*/

/*${function:start}*/
void BOARD_InitHardware
{
    BOARD_InitPins();
```

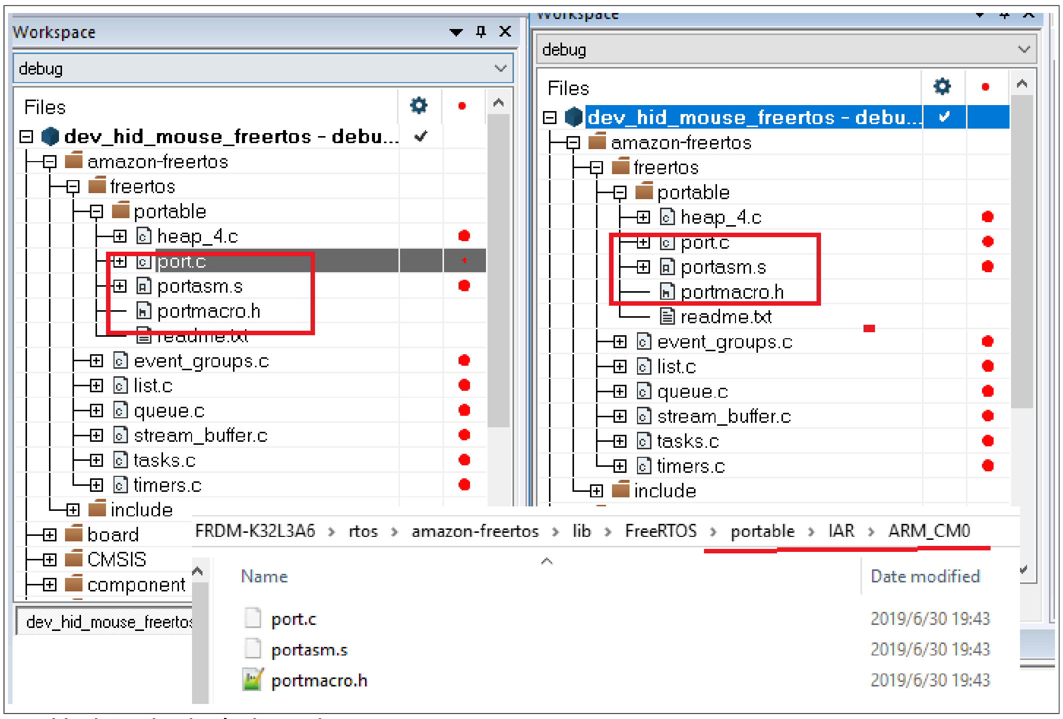
Add FRDM-K32L3A6 \devices\ K32L3A60 \drivers\fsl_intmux.c and fsl_intmux.h files to project.



Name	Date modified	Type
fsl_ftfx_flexnvm.c	2019/6/30 19:43	C File
fsl_ftfx_flexnvm.h	2019/6/30 19:43	H File
fsl_ftfx_utilities.h	2019/6/30 19:43	H File
fsl_gpio.c	2019/6/30 19:43	C File
fsl_gpio.h	2019/6/30 19:43	H File
fsl_intmux.c	2019/6/30 19:43	C File
fsl_intmux.h	2019/6/30 19:43	H File
fsl_llwu.c	2019/6/30 19:43	C File
fsl_llwu.h	2019/6/30 19:43	H File

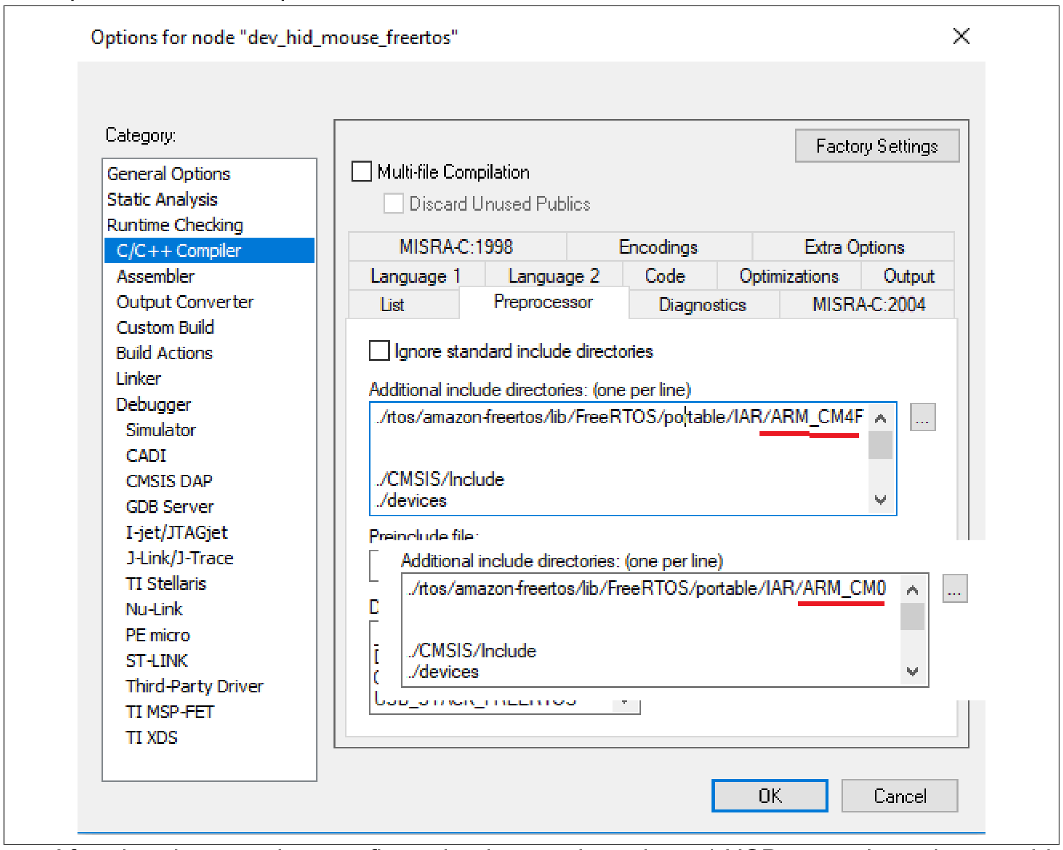
File name: "fsl_intmux.h" "fsl_intmux.c"

6. In M4 project in IAR, in option C/C++ compiler > Preprocessor. The file path “$PROJ_DIR$/../../../../../../devices/ K32L3A60 /drivers” must be in M0p project setting.
7. For freertos example, update the freertos related portable file and include path from M4 to M0.



Name	Date modified
port.c	2019/6/30 19:43
portasm.s	2019/6/30 19:43
portmacro.h	2019/6/30 19:43

Update the include path:



Additional include directories: (one per line)

```
./rtos/amazon-freertos/lib/FreeRTOS/portable/IAR/ARM_CM4F

./CMSIS/Include
./devices
```

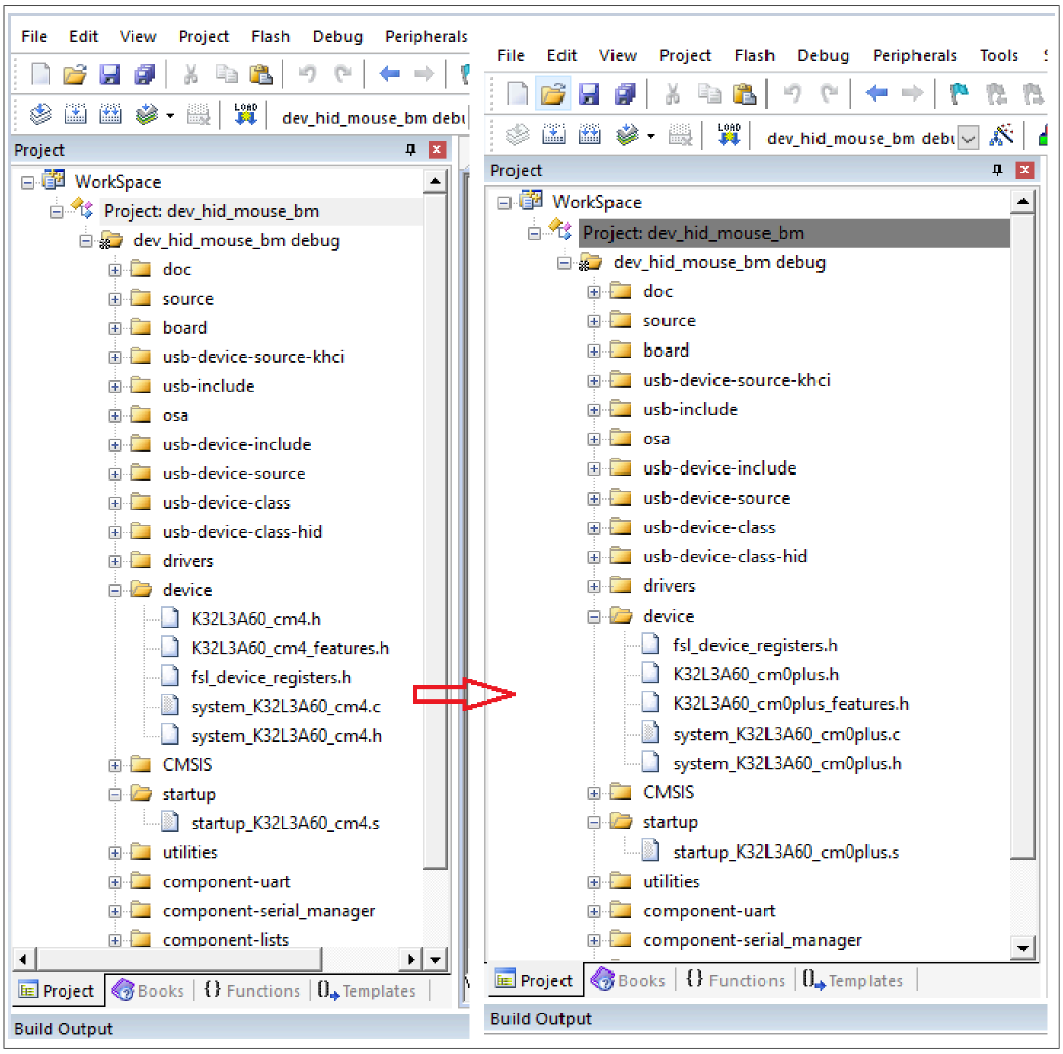
Additional include directories: (one per line)

```
./rtos/amazon-freertos/lib/FreeRTOS/portable/IAR/ARM_CM0

./CMSIS/Include
./devices
```

After the above project configuration is complete, the m4 USB example project would be changed to M0p project. M0p example USB project can now be downloaded and debugged.

3 MDK

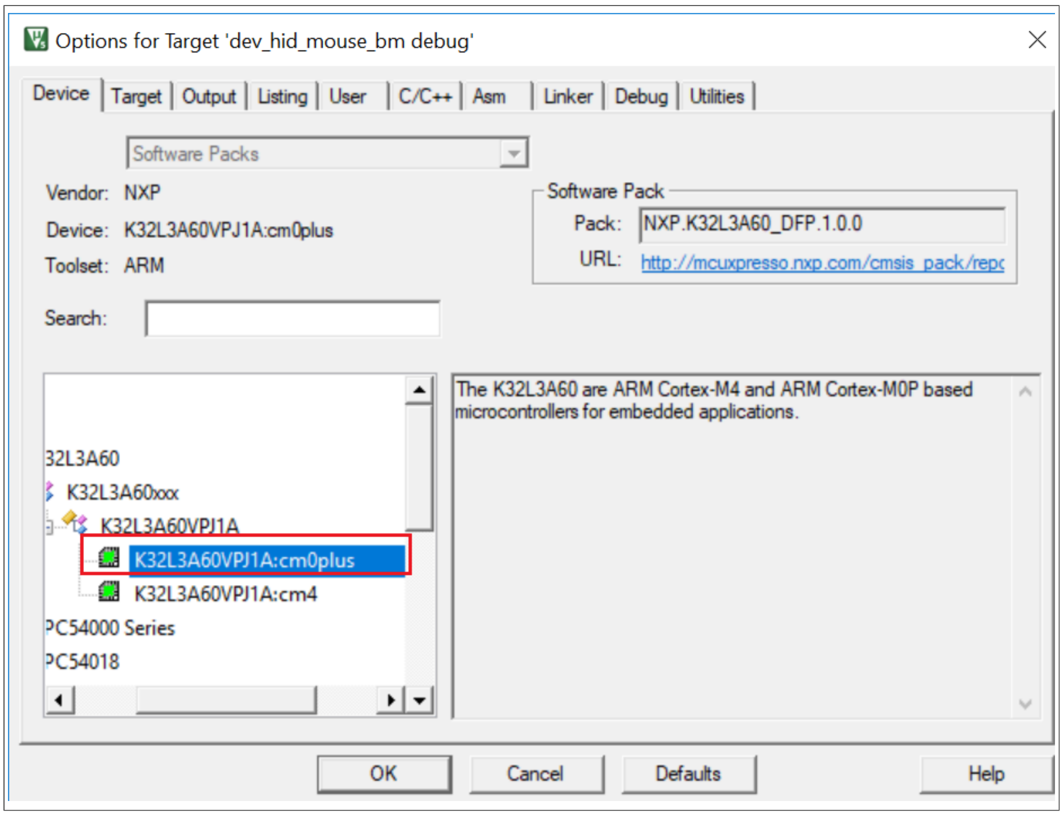
1. Change startup and system file from M4 platform files to M0p platform files. The below picture shows how to change m4 project files to M0p project file.
   The files are available in folder: FRDM-K32L3A6\devices\K32L3A60


File Edit View Project Flash Debug Peripherals
dev_hid_mouse_bm debug
Project
Project Books Functions Templates
Build Output
File Edit View Project Flash Debug Peripherals Tools
dev_hid_mouse_bm debug
Project
Project Books Functions Templates
Build Output

```
WorkSpace
  Project: dev_hid_mouse_bm
    dev_hid_mouse_bm debug
      doc
      source
      board
      usb-device-source-khci
      usb-include
      osa
      usb-device-include
      usb-device-source
      usb-device-class
      usb-device-class-hid
      drivers
      device
        K32L3A60_cm4.h
        K32L3A60_cm4_features.h
        fsl_device_registers.h
        system_K32L3A60_cm4.c
        system_K32L3A60_cm4.h
      CMSIS
      startup
        startup_K32L3A60_cm4.s
      utilities
      component-uart
      component-serial_manager
      component-lists
```

```
WorkSpace
  Project: dev_hid_mouse_bm
    dev_hid_mouse_bm debug
      doc
      source
      board
      usb-device-source-khci
      usb-include
      osa
      usb-device-include
      usb-device-source
      usb-device-class
      usb-device-class-hid
      drivers
      device
        fsl_device_registers.h
        K32L3A60_cm0plus.h
        K32L3A60_cm0plus_features.h
        system_K32L3A60_cm0plus.c
        system_K32L3A60_cm0plus.h
      CMSIS
      startup
        startup_K32L3A60_cm0plus.s
      utilities
      component-uart
      component-serial_manager
```

2. Open M4 project in MDK, in option > Device, change the CPU type form M4 to M0p.



Options for Target 'dev_hid_mouse_bm debug'

Software Packs

Vendor: NXP

Device: K32L3A60VPJ1A:cm0plus

Toolset: ARM

Search:

Software Pack

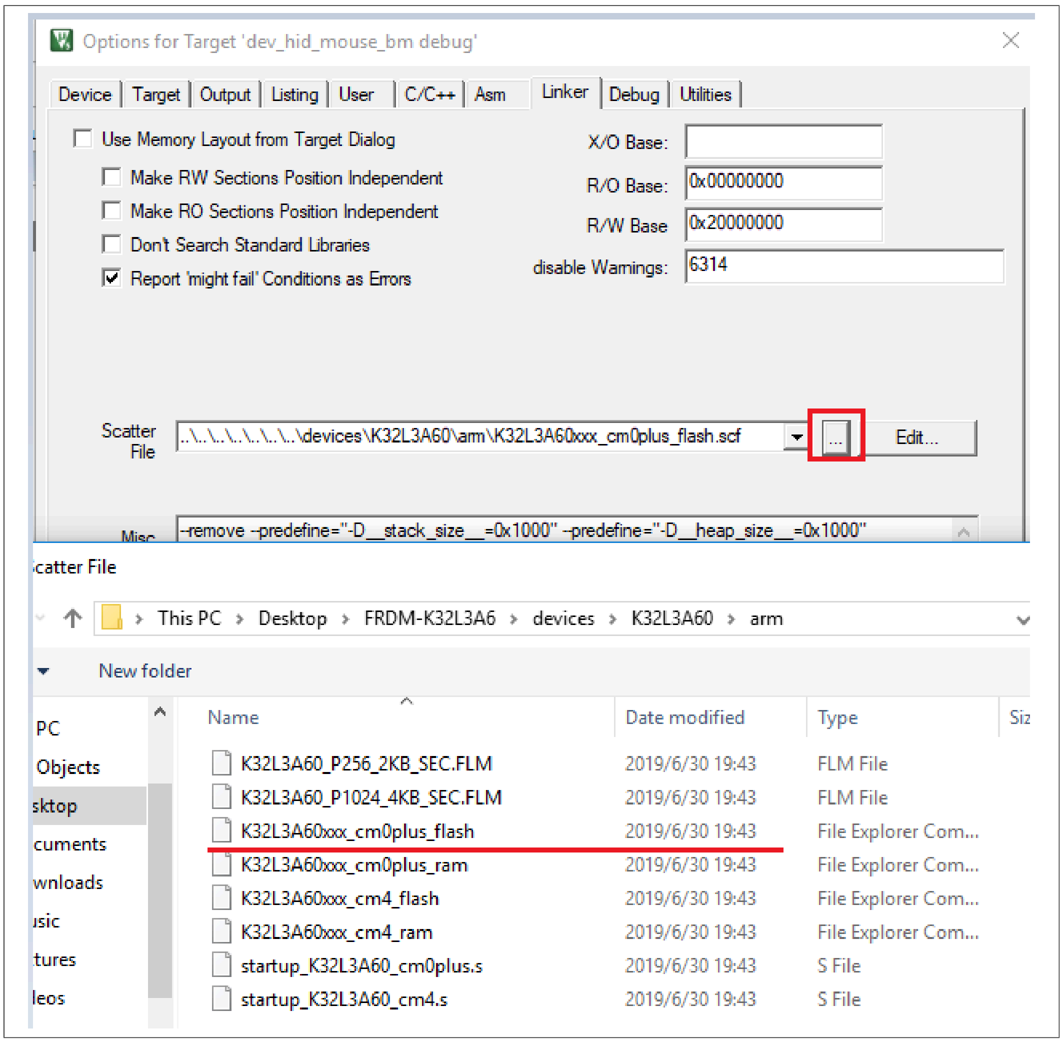
Pack: NXP.K32L3A60_DFP.1.0.0

URL: http://mcuxpresso.nxp.com/cmsis_pack/repc

32L3A60
K32L3A60xxx
K32L3A60VPJ1A
K32L3A60VPJ1A:cm0plus
K32L3A60VPJ1A:cm4
PC54000 Series
PC54018

The K32L3A60 are ARM Cortex-M4 and ARM Cortex-M0P based microcontrollers for embedded applications.

3. Again in M4 project in MDK, in option-->linker, change the linker configure file form M4 link file to M0p link file. The linker file path is devices\K32L3A60 folder.



Options for Target 'dev_hid_mouse_bm debug'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

- [ ] Use Memory Layout from Target Dialog
  - [ ] Make RW Sections Position Independent
  - [ ] Make RO Sections Position Independent
  - [ ] Dont Search Standard Libraries
  - [✓] Report 'might fail' Conditions as Errors

X/O Base:

R/O Base: 0x00000000

R/W Base 0x20000000

disable Warnings: 6314

Scatter File: ..\..\..\..\..\..\..\devices\K32L3A60\arm\K32L3A60xxx_cm0plus_flash.scf

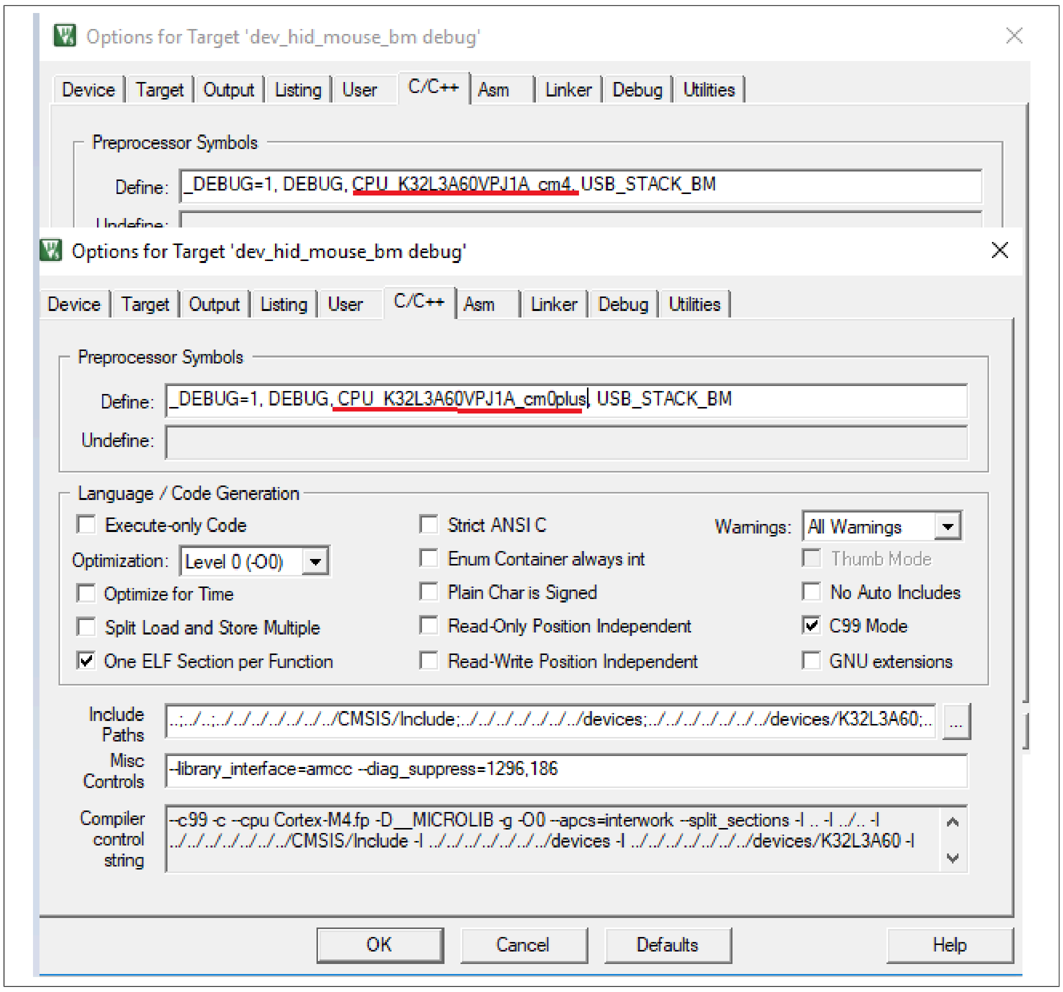
Misc: --remove --predefine="-D__stack_size__=0x1000" --predefine="-D__heap_size__=0x1000"

catter File

This PC › Desktop › FRDM-K32L3A6 › devices › K32L3A60 › arm

Name	Date modified	Type
K32L3A60_P256_2KB_SEC.FLM	2019/6/30 19:43	FLM File
K32L3A60_P1024_4KB_SEC.FLM	2019/6/30 19:43	FLM File
K32L3A60xxx_cm0plus_flash	2019/6/30 19:43	File Explorer Com...
K32L3A60xxx_cm0plus_ram	2019/6/30 19:43	File Explorer Com...
K32L3A60xxx_cm4_flash	2019/6/30 19:43	File Explorer Com...
K32L3A60xxx_cm4_ram	2019/6/30 19:43	File Explorer Com...
startup_K32L3A60_cm0plus.s	2019/6/30 19:43	S File
startup_K32L3A60_cm4.s	2019/6/30 19:43	S File

4. In M4 project in MDK, in option > C/C++ compiler, change the CPU MACRO from " CPU_K32L3A60VPJ1A_cm4" to "CPU_K32L3A60VPJ1A_cm0plus".



Define: _DEBUG=1, DEBUG, CPU_K32L3A60VPJ1A_cm4, USB_STACK_BM

Define: _DEBUG=1, DEBUG, CPU_K32L3A60VPJ1A_cm0plus, USB_STACK_BM

Undefine:

Language / Code Generation

- [ ] Execute-only Code
- Optimization: Level 0 (-O0)
- [ ] Optimize for Time
- [ ] Split Load and Store Multiple
- [x] One ELF Section per Function
- [ ] Strict ANSI C
- [ ] Enum Container always int
- [ ] Plain Char is Signed
- [ ] Read-Only Position Independent
- [ ] Read-Write Position Independent
- Warnings: All Warnings
- [ ] Thumb Mode
- [ ] No Auto Includes
- [x] C99 Mode
- [ ] GNU extensions

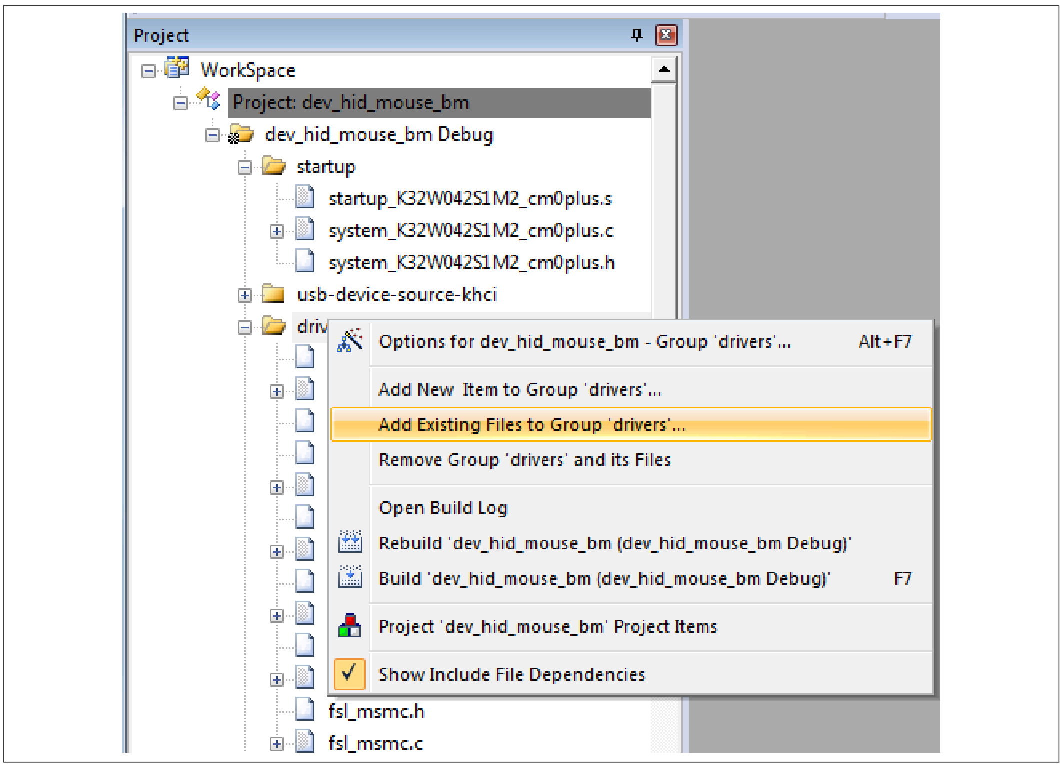
Include Paths: ..;../..;../../../../../../../CMSIS/Include;../../../../../../../devices;../../../../../../../devices/K32L3A60;..

Misc Controls: --library_interface=armcc --diag_suppress=1296,186

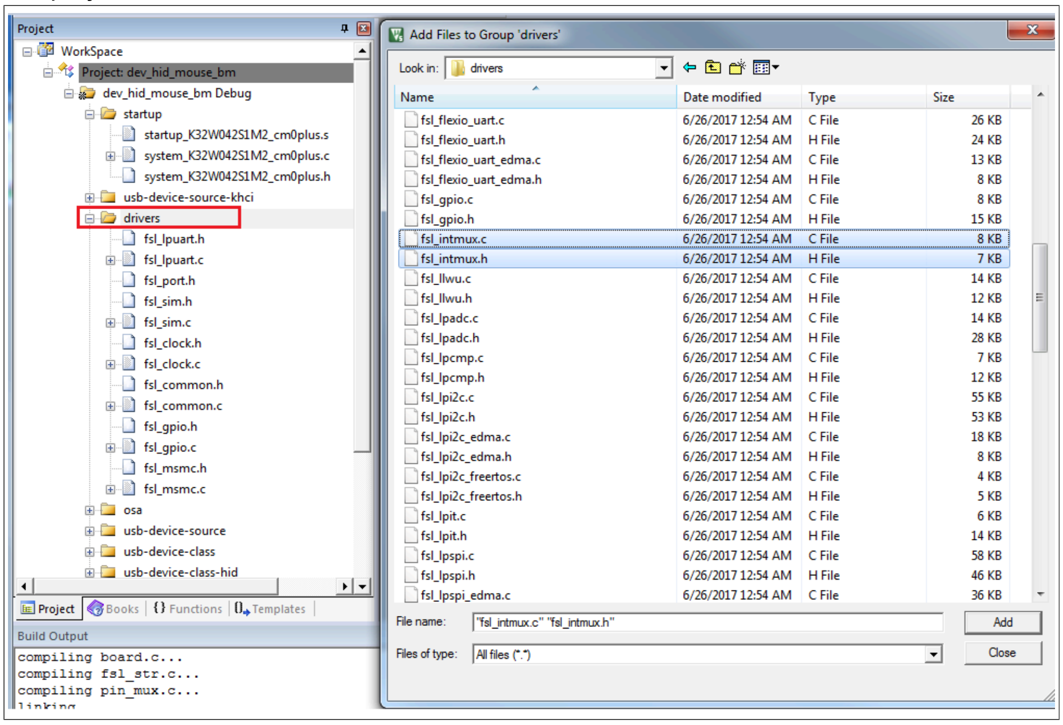
Compiler control string:

```
--c99 -c --cpu Cortex-M4.fp -D__MICROLIB -g -O0 --apcs=interwork --split_sections -I .. -I ../.. -I
../../../../../../../CMSIS/Include -I ../../../../../../../devices -I ../../../../../../../devices/K32L3A60 -I
```

5. Add int-mux file to M0p project, driver -add Exiting Files to Group ‘driver’, as shown in the image bellow


Project
WorkSpace
Project: dev_hid_mouse_bm
dev_hid_mouse_bm Debug
startup
startup_K32W042S1M2_cm0plus.s
system_K32W042S1M2_cm0plus.c
system_K32W042S1M2_cm0plus.h
usb-device-source-khci
driv
Options for dev_hid_mouse_bm - Group 'drivers'... Alt+F7
Add New Item to Group 'drivers'...
Add Existing Files to Group 'drivers'...
Remove Group 'drivers' and its Files
Open Build Log
Rebuild 'dev_hid_mouse_bm (dev_hid_mouse_bm Debug)'
Build 'dev_hid_mouse_bm (dev_hid_mouse_bm Debug)' F7
Project 'dev_hid_mouse_bm' Project Items
Show Include File Dependencies
fsl_msmc.h
fsl_msmc.c

6. Add FRDM-K32L3A6\devices\K32L3A60\drivers\fsl_intmux.c and fsl_intmux.h files to project.



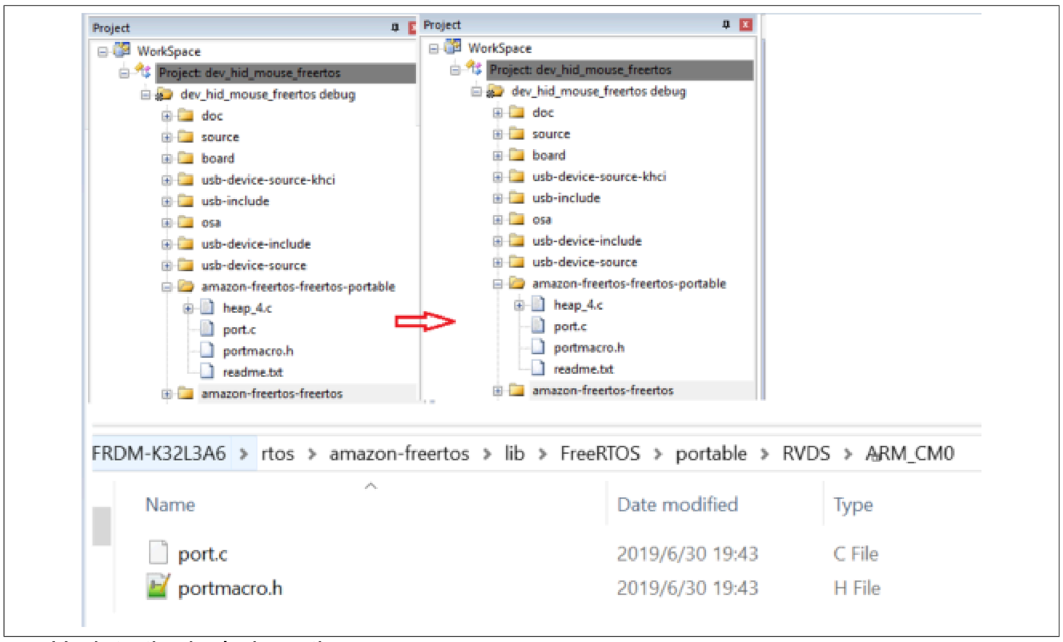
Name	Date modified	Type	Size
fsl_flexio_uart.c	6/26/2017 12:54 AM	C File	26 KB
fsl_flexio_uart.h	6/26/2017 12:54 AM	H File	24 KB
fsl_flexio_uart_edma.c	6/26/2017 12:54 AM	C File	13 KB
fsl_flexio_uart_edma.h	6/26/2017 12:54 AM	H File	8 KB
fsl_gpio.c	6/26/2017 12:54 AM	C File	8 KB
fsl_gpio.h	6/26/2017 12:54 AM	H File	15 KB
fsl_intmux.c	6/26/2017 12:54 AM	C File	8 KB
fsl_intmux.h	6/26/2017 12:54 AM	H File	7 KB
fsl_llwu.c	6/26/2017 12:54 AM	C File	14 KB
fsl_llwu.h	6/26/2017 12:54 AM	H File	12 KB
fsl_lpadc.c	6/26/2017 12:54 AM	C File	14 KB
fsl_lpadc.h	6/26/2017 12:54 AM	H File	28 KB
fsl_lpcmp.c	6/26/2017 12:54 AM	C File	7 KB
fsl_lpcmp.h	6/26/2017 12:54 AM	H File	12 KB
fsl_lpi2c.c	6/26/2017 12:54 AM	C File	55 KB
fsl_lpi2c.h	6/26/2017 12:54 AM	H File	53 KB
fsl_lpi2c_edma.c	6/26/2017 12:54 AM	C File	18 KB
fsl_lpi2c_edma.h	6/26/2017 12:54 AM	H File	8 KB
fsl_lpi2c_freertos.c	6/26/2017 12:54 AM	C File	4 KB
fsl_lpi2c_freertos.h	6/26/2017 12:54 AM	H File	5 KB
fsl_lpit.c	6/26/2017 12:54 AM	C File	6 KB
fsl_lpit.h	6/26/2017 12:54 AM	H File	14 KB
fsl_lpspi.c	6/26/2017 12:54 AM	C File	58 KB
fsl_lpspi.h	6/26/2017 12:54 AM	H File	46 KB
fsl_lpspi_edma.c	6/26/2017 12:54 AM	C File	36 KB

File name: "fsl_intmux.c" "fsl_intmux.h"

Files of type: All files (*.*)

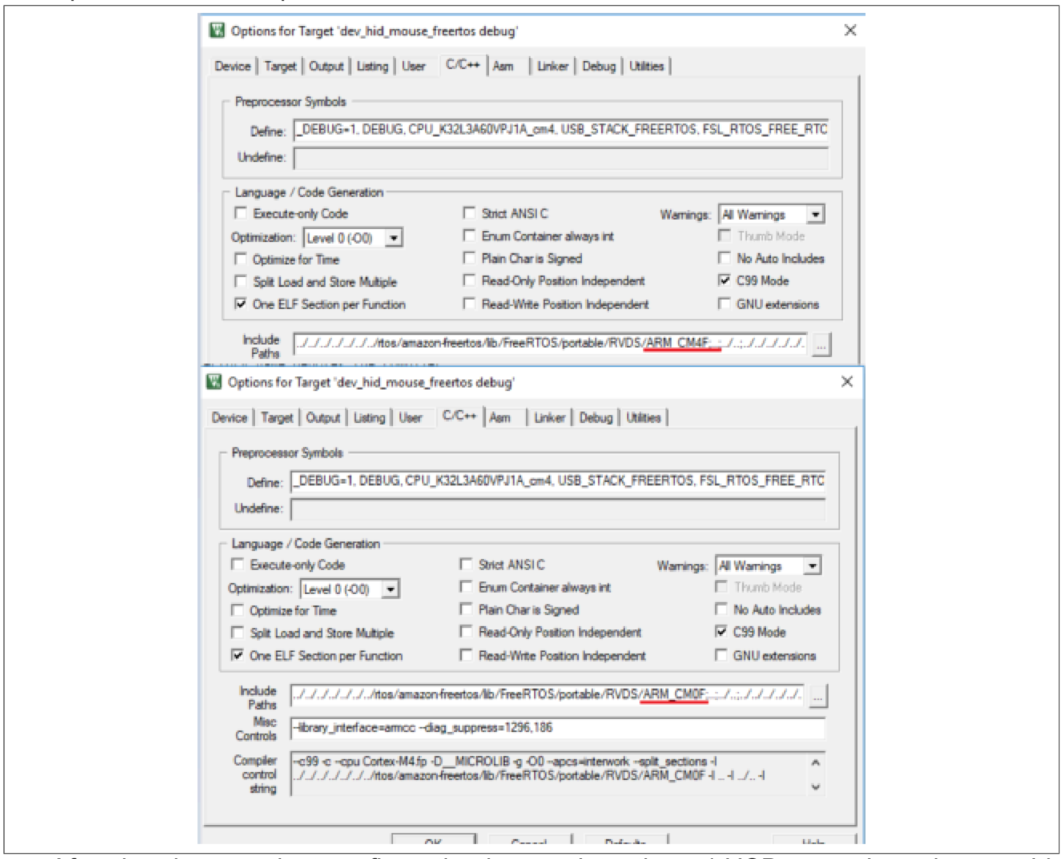
```
compiling board.c...
compiling fsl_str.c...
compiling pin_mux.c...
linking
```

7. For freertos example. Update the freertos related portable file and include path from M4 to M0.


Project
WorkSpace
Project: dev_hid_mouse_freertos
dev_hid_mouse_freertos debug
doc
source
board
usb-device-source-khci
usb-include
osa
usb-device-include
usb-device-source
amazon-freertos-freertos-portable
heap_4.c
port.c
portmacro.h
readme.txt
amazon-freertos-freertos
FRDM-K32L3A6 › rtos › amazon-freertos › lib › FreeRTOS › portable › RVDS › ARM_CM0

Name	Date modified	Type
port.c	2019/6/30 19:43	C File
portmacro.h	2019/6/30 19:43	H File

Update the include path:



Preprocessor Symbols

Define: _DEBUG=1, DEBUG, CPU_K32L3A60VPJ1A_cm4, USB_STACK_FREERTOS, FSL_RTOS_FREE_RTC

Undefine:

Language / Code Generation

- [ ] Execute-only Code
- Optimization: Level 0 (-O0)
- [ ] Optimize for Time
- [ ] Split Load and Store Multiple
- [x] One ELF Section per Function
- [ ] Strict ANSI C
- [ ] Enum Container always int
- [ ] Plain Char is Signed
- [ ] Read-Only Position Independent
- [ ] Read-Write Position Independent
- Warnings: All Warnings
- [ ] Thumb Mode
- [ ] No Auto Includes
- [x] C99 Mode
- [ ] GNU extensions

Include Paths: ../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/RVDS/ARM_CM4F;..;../..;../../../../../.

Include Paths: ../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/RVDS/ARM_CM0F;..;../..;../../../../../.

Misc Controls: --library_interface=armcc --diag_suppress=1296,186

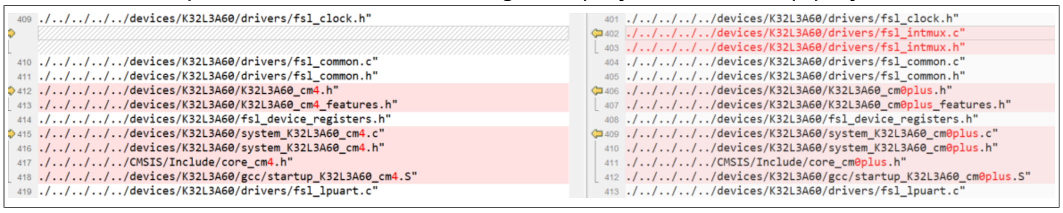
Compiler control string:

```
--c99 -c --cpu Cortex-M4.fp -D__MICROLIB -g -O0 --apcs=interwork --split_sections -I
../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/RVDS/ARM_CM0F -I .. -I ../.. -I
```

After the above project configuration is complete, the m4 USB example project would be changed to M0p project. M0p example USB project can now be downloaded and debugged.

4 ARMGCC

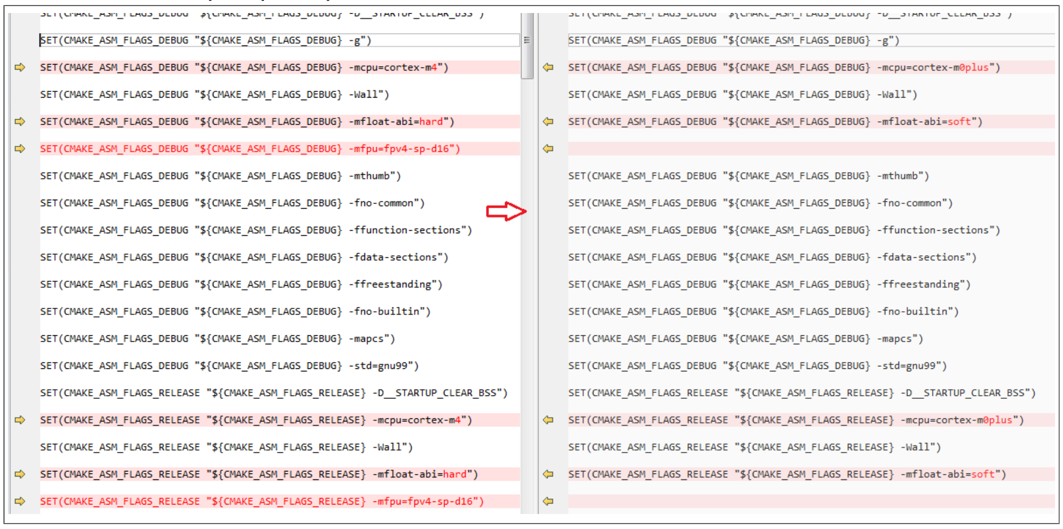
1. Update startup and system file from M4 platform files to M0p platform files.
   Open the CMakeLists.txt of the example, such as FRDM-K32L3A6\boards\frdmk 32l3a6\usb_examples\usb_device_hid_mouse\bm\armgcc\ CMakeLists.txt
   The bellow picture shows how to change m4 project files to M0p project file.



```
./../../../../devices/K32L3A60/drivers/fsl_clock.h"

./../../../../devices/K32L3A60/drivers/fsl_common.c"
./../../../../devices/K32L3A60/drivers/fsl_common.h"
./../../../../devices/K32L3A60/K32L3A60_cm4.h"
./../../../../devices/K32L3A60/K32L3A60_cm4_features.h"
./../../../../devices/K32L3A60/fsl_device_registers.h"
./../../../../devices/K32L3A60/system_K32L3A60_cm4.c"
./../../../../devices/K32L3A60/system_K32L3A60_cm4.h"
./../../../../CMSIS/Include/core_cm4.h"
./../../../../devices/K32L3A60/gcc/startup_K32L3A60_cm4.S"
./../../../../devices/K32L3A60/drivers/fsl_lpuart.c"
```

```
./../../../../devices/K32L3A60/drivers/fsl_clock.h"
./../../../../devices/K32L3A60/drivers/fsl_intmux.c"
./../../../../devices/K32L3A60/drivers/fsl_intmux.h"
./../../../../devices/K32L3A60/drivers/fsl_common.c"
./../../../../devices/K32L3A60/drivers/fsl_common.h"
./../../../../devices/K32L3A60/K32L3A60_cm0plus.h"
./../../../../devices/K32L3A60/K32L3A60_cm0plus_features.h"
./../../../../devices/K32L3A60/fsl_device_registers.h"
./../../../../devices/K32L3A60/system_K32L3A60_cm0plus.c"
./../../../../devices/K32L3A60/system_K32L3A60_cm0plus.h"
./../../../../CMSIS/Include/core_cm0plus.h"
./../../../../devices/K32L3A60/gcc/startup_K32L3A60_cm0plus.S"
./../../../../devices/K32L3A60/drivers/fsl_lpuart.c"
```

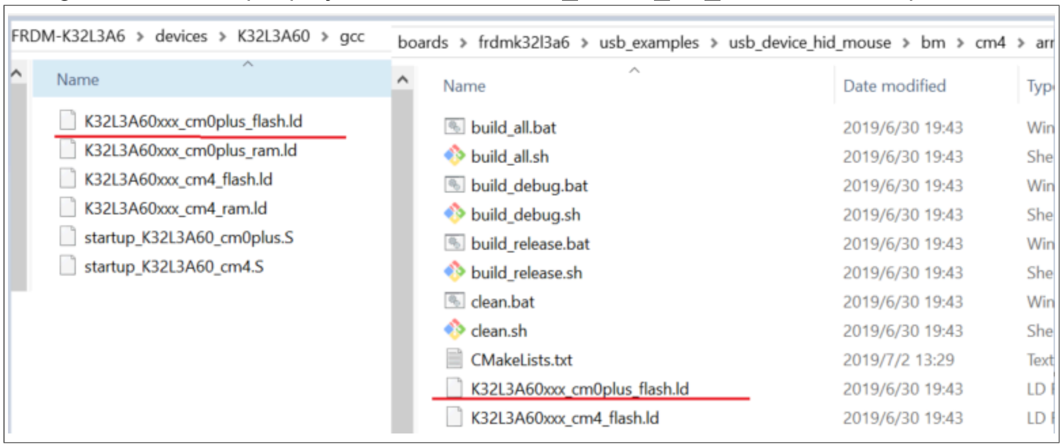
2. Open CMakeLists.txt.
   Replace all the CPU setting from cortex-m4 to cortex-m0plus.
   Replace all the ”-mfloat-abi=hard” to “-mfloat-abi=soft”
   Delete all “mfpu=fpv4-sp-d16”



```
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -g")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mcpu=cortex-m4")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -Wall")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mfloat-abi=hard")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mfpu=fpv4-sp-d16")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mthumb")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -fno-common")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -ffunction-sections")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -fdata-sections")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -ffreestanding")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -fno-builtin")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mapcs")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -std=gnu99")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -D__STARTUP_CLEAR_BSS")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -mcpu=cortex-m4")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -Wall")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -mfloat-abi=hard")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -mfpu=fpv4-sp-d16")
```

```
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -g")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mcpu=cortex-m0plus")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -Wall")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mfloat-abi=soft")

SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mthumb")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -fno-common")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -ffunction-sections")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -fdata-sections")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -ffreestanding")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -fno-builtin")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -mapcs")
SET(CMAKE_ASM_FLAGS_DEBUG "${CMAKE_ASM_FLAGS_DEBUG} -std=gnu99")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -D__STARTUP_CLEAR_BSS")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -mcpu=cortex-m0plus")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -Wall")
SET(CMAKE_ASM_FLAGS_RELEASE "${CMAKE_ASM_FLAGS_RELEASE} -mfloat-abi=soft")
```

3. Change the linker configure file form M4 link file to M0p link file. The linker file path is devices\ K32L3A60 folder.
   Copy K32L3A60xxx_cm0plus_flash.ld from FRDM-K32L3A6 \devices\ K32L3A60 \gcc to the example project folder. Take usb_device_hid_mouse as example.



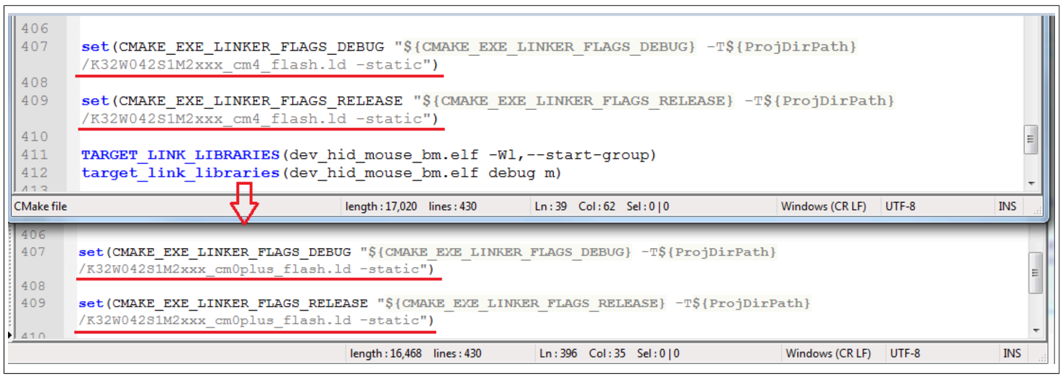
FRDM-K32L3A6 › devices › K32L3A60 › gcc

Name
K32L3A60xxx_cm0plus_flash.ld
K32L3A60xxx_cm0plus_ram.ld
K32L3A60xxx_cm4_flash.ld
K32L3A60xxx_cm4_ram.ld
startup_K32L3A60_cm0plus.S
startup_K32L3A60_cm4.S

boards › frdmk32l3a6 › usb_examples › usb_device_hid_mouse › bm › cm4 › arr

Name	Date modified	Typ
build_all.bat	2019/6/30 19:43	Win
build_all.sh	2019/6/30 19:43	She
build_debug.bat	2019/6/30 19:43	Win
build_debug.sh	2019/6/30 19:43	She
build_release.bat	2019/6/30 19:43	Win
build_release.sh	2019/6/30 19:43	She
clean.bat	2019/6/30 19:43	Win
clean.sh	2019/6/30 19:43	She
CMakeLists.txt	2019/7/2 13:29	Text
K32L3A60xxx_cm0plus_flash.ld	2019/6/30 19:43	LD I
K32L3A60xxx_cm4_flash.ld	2019/6/30 19:43	LD I

4. Modify the CMakeLists.txt.



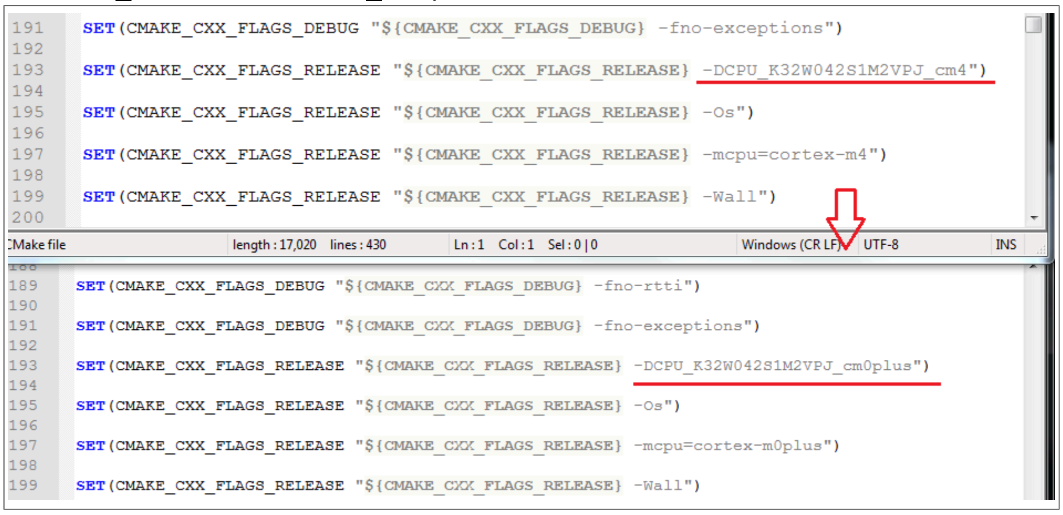
```
set(CMAKE_EXE_LINKER_FLAGS_DEBUG "${CMAKE_EXE_LINKER_FLAGS_DEBUG} -T${ProjDirPath}
/K32W042S1M2xxx_cm4_flash.ld -static")

set(CMAKE_EXE_LINKER_FLAGS_RELEASE "${CMAKE_EXE_LINKER_FLAGS_RELEASE} -T${ProjDirPath}
/K32W042S1M2xxx_cm4_flash.ld -static")

TARGET_LINK_LIBRARIES(dev_hid_mouse_bm.elf -Wl,--start-group)
target_link_libraries(dev_hid_mouse_bm.elf debug m)
```

CMake file   length : 17,020   lines : 430   Ln : 39   Col : 62   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS

```
set(CMAKE_EXE_LINKER_FLAGS_DEBUG "${CMAKE_EXE_LINKER_FLAGS_DEBUG} -T${ProjDirPath}
/K32W042S1M2xxx_cm0plus_flash.ld -static")

set(CMAKE_EXE_LINKER_FLAGS_RELEASE "${CMAKE_EXE_LINKER_FLAGS_RELEASE} -T${ProjDirPath}
/K32W042S1M2xxx_cm0plus_flash.ld -static")
```

length : 16,468   lines : 430   Ln : 396   Col : 35   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS

5. Change all the CPU MACRO from "CPU_K32W042S1M2VPJ_cm4" to "CPU_K32W042S1M2VPJ_cm0plus" in CMakeLists.txt.

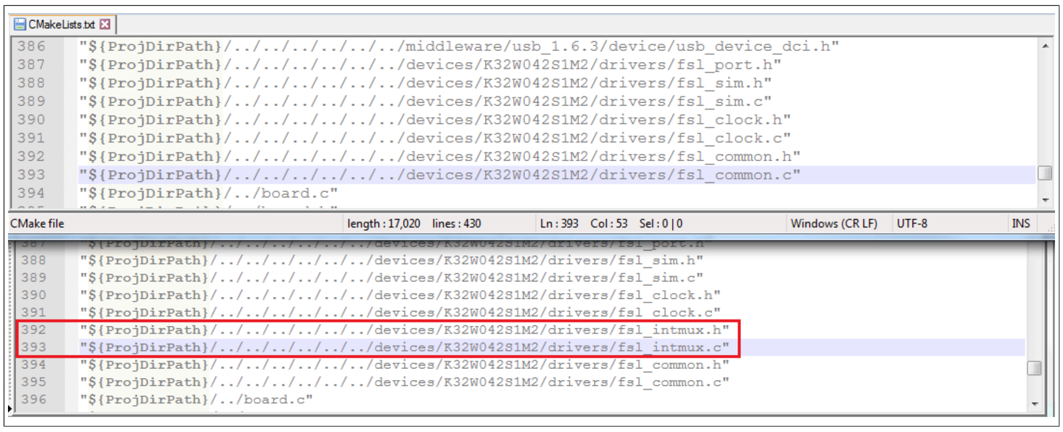


```
SET(CMAKE_CXX_FLAGS_DEBUG "${CMAKE_CXX_FLAGS_DEBUG} -fno-exceptions")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -DCPU_K32W042S1M2VPJ_cm4")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -Os")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -mcpu=cortex-m4")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -Wall")
```

CMake file   length : 17,020   lines : 430   Ln : 1   Col : 1   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS

```
SET(CMAKE_CXX_FLAGS_DEBUG "${CMAKE_CXX_FLAGS_DEBUG} -fno-rtti")

SET(CMAKE_CXX_FLAGS_DEBUG "${CMAKE_CXX_FLAGS_DEBUG} -fno-exceptions")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -DCPU_K32W042S1M2VPJ_cm0plus")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -Os")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -mcpu=cortex-m0plus")

SET(CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE} -Wall")
```

6. Add int-mux file to M0p project, as bellow
   Add FRDM-K32L3A6 \devices\K32W042 K32L3A60 S1M2\drivers\fsl_intmux.c and fsl_intmux.h files to project.
7. Check the fsl_intmux.c/ fsl_intmux.h is in FRDM-K32L3A6 \devices\K32W042 K32L3A60 S1M2\drivers folder.



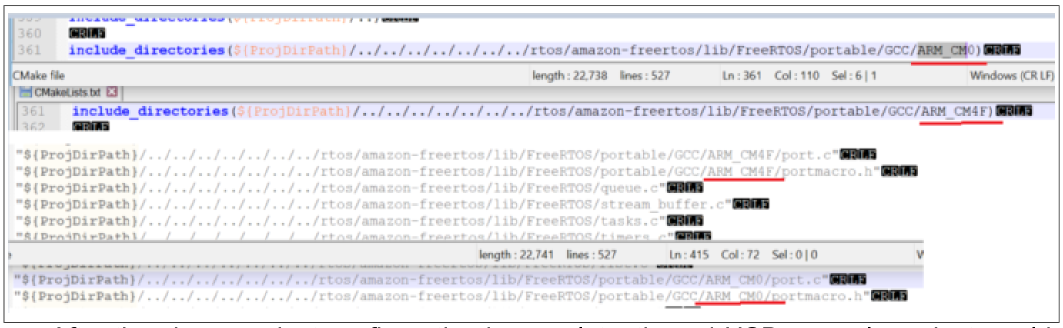
CMakeLists.txt

```
"${ProjDirPath}/../../../../../../middleware/usb_1.6.3/device/usb_device_dci.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_port.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_sim.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_sim.c"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_clock.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_clock.c"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_common.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_common.c"
"${ProjDirPath}/../board.c"
```

CMake file   length : 17,020   lines : 430   Ln : 393   Col : 53   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS

```
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_sim.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_sim.c"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_clock.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_clock.c"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_intmux.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_intmux.c"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_common.h"
"${ProjDirPath}/../../../../../../devices/K32W042S1M2/drivers/fsl_common.c"
"${ProjDirPath}/../board.c"
```

8. For freertos example. Update the freertos related portable file and include path from M4 to M0.
   Change all the source path and include path“Source/portable/GCC/ARM_CM4F” to “Source/portable/GCC/ARM_CM0”,



```
include_directories(${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/GCC/ARM_CM0)CRLF
```

CMake file length : 22,738 lines : 527 Ln : 361 Col : 110 Sel : 6 | 1 Windows (CR LF)

CMakeLists.txt

```
include_directories(${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/GCC/ARM_CM4F)CRLF
CRLF
```

```
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/GCC/ARM_CM4F/port.c"CRLF
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/GCC/ARM_CM4F/portmacro.h"CRLF
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/queue.c"CRLF
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/stream_buffer.c"CRLF
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/tasks.c"CRLF
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/timers.c"CRLF
```

length : 22,741 lines : 527 Ln : 415 Col : 72 Sel : 0 | 0

```
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/GCC/ARM_CM0/port.c"CRLF
"${ProjDirPath}/../../../../../../../rtos/amazon-freertos/lib/FreeRTOS/portable/GCC/ARM_CM0/portmacro.h"CRLF
```

After the above project configuration is complete, the m4 USB example project would be changed to M0p project. M0p example USB project can now be downloaded and debugged.

5 Revision history

This table summarizes revisions to this document.

Table 1. Revision history

Revision number	Date	Substantive changes
0	20 March 2020	Initial release
1	11 July 2022	Editorial and layout updates.

6 Legal information

6.1 Definitions

6.2 Disclaimers

6.3 Trademarks

Contents

Date of release: 11 July 2022
Document identifier: AN12671