

# FreeRTOS for i.MX6SX

Biyong SUN  
11, DEC 2015



EXTERNAL USE



SECURE CONNECTIONS  
FOR A SMARTER WORLD

# FreeRTOS ARM GCC Compile Environment VS. MQX

## Windows

	CMAKE	ARM GCC	MinGW
freeRTOS	Y	Y	Y
MQX	N	Y	Y

## Linux

	CMAKE	ARM GCC	MinGW
freeRTOS	Y	Y	N
MQX	N	Y	N



# FreeRTOS Various Memories VS. MQX

## Windows/Linux

	TCM	OCRAM	DDR	QSPI
freeRTOS	Y	Y	Y	Y
MQX	N	Y	Y	Y

# FreeRTOS Development Tools

The requirements for development tools are listed in the release notes.

The FreeRTOS BSP 1.0.0 was compiled and tested with these development tools:

- IAR Embedded Workbench for ARM® version 7.40.3
- Makefiles support with GCC revision 4.9-2015-q1-update from ARM Embedded
- CMake 3.0.x
- ARM Development Studio 5 (DS-5™) version 5.20.1 (32 bit)
- Lauterbach TRACE32 PowerView for ARM version R.2014.09.000058270



# FreeRTOS ARM GCC Compile VS. MQX

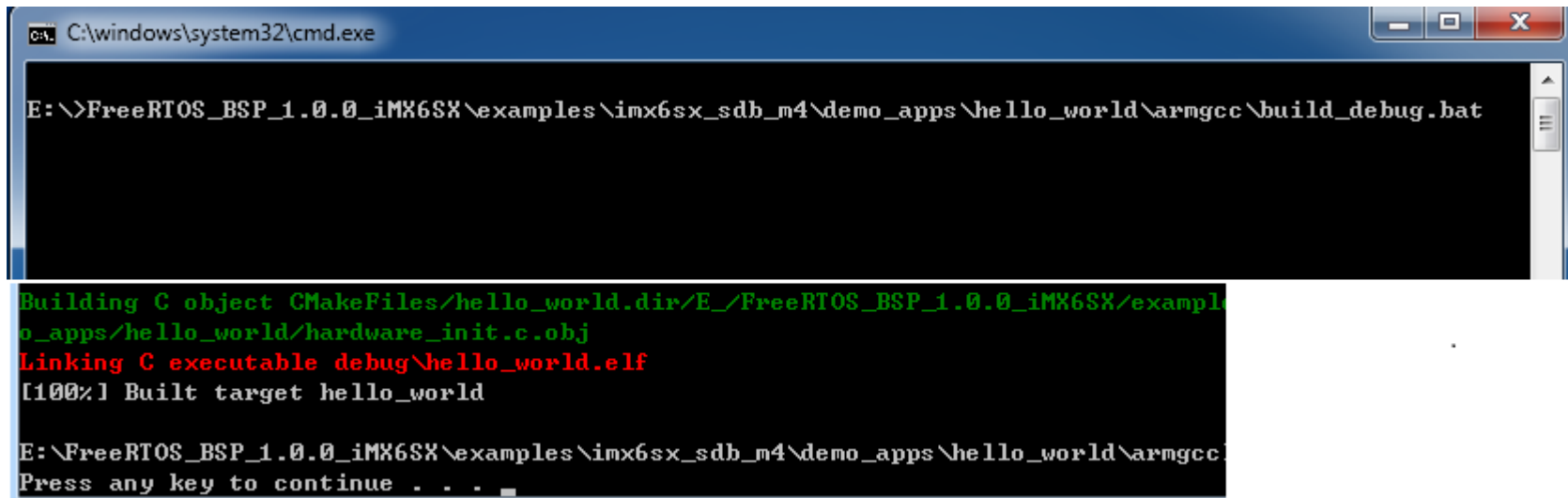
FreeRTOS --- Compile the application directly. And output elf and binary files.

MQX --- Need to compile library before compile application.

And output elf file only. Need to manually objcopy to binary file.

# Compile FreeRTOS application with ARM GCC windows

- Follow “Getting Started with FreeRTOS™ BSP for i.MX 6SoloX” ---  
5 Building a Demo Using ARM® GCC” to install the compile environment and freeRTOS.  
Compare to MQX, just one more step to install cmake.  
**Note: Please don't forget to add a new system environment variable for ARMGCC\_DIR**
- Enter the application directory run the batch file or double click the batch file.



```
C:\windows\system32\cmd.exe
E:\>FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc\build_debug.bat

Building C object CMakeFiles/hello_world.dir/E:/FreeRTOS_BSP_1.0.0_iMX6SX/examples\demo_apps\hello_world\hardware_init.c.obj
Linking C executable debug\hello_world.elf
[100%] Built target hello_world

E:\FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc
Press any key to continue . . .
```

# Compile FreeRTOS application with ARM GCC Linux

- Not instruction in documents for compiling with Linux.  
But make sure cmake is installed in the host.
- Enter the application directory run the shell script file with given ARMGCC\_DIR

```
ARMGCC_DIR=~/gcc-arm-none-eabi-4_9-2015q1 ./build_debug.sh
```

```
[ 86%] [ 91%] Building C object CMakeFiles/hello_world.dir/home/bysun/tmp/FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/pin_mux.c.obj  
Building C object CMakeFiles/hello_world.dir/home/bysun/tmp/FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/board.c.obj  
[ 95%] Building C object CMakeFiles/hello_world.dir/home/bysun/tmp/FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/clock_freq.c.obj  
[100%] Building C object  
CMakeFiles/hello_world.dir/home/bysun/tmp/FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/demo_apps/hello_world/hardware_init.c.obj  
Linking C executable debug/hello_world.elf  
[100%] Built target hello_world
```

```
ls debug/  
hello_world.bin hello_world.elf hello_world.hex hello_world.map
```

# Run FreeRTOS demo

- Check the Link script in CMakeLists.txt

```
hello_world
|-- armgcc
  |-- build_all.bat
  |-- build_all.sh
  |-- build_debug.bat
  |-- build_debug.sh
  |-- build_release.bat
  |-- build_release.sh
  |-- clean.bat
  |-- clean.sh
  `-- CMakeLists.txt
```

```
24 # DEBUG LINK FILE
25 set(CMAKE_EXE_LINKER_FLAGS_DEBUG "${CMAKE_EXE_LINKER_FLAGS_DEBUG} -
T${ProjDirPath}/../../../../platform/devices/MCIMX6X/linker/gcc/MCIMX6X_M4_tcm.ld -static")
26
27 # RELEASE LINK FILE
28 set(CMAKE_EXE_LINKER_FLAGS_RELEASE "${CMAKE_EXE_LINKER_FLAGS_RELEASE} -
T${ProjDirPath}/../../../../platform/devices/MCIMX6X/linker/gcc/MCIMX6X_M4_tcm.ld -static ")
```





# Run FreeRTOS demo (Cont.)

- Address Table for TCM, OCRAM, DDR, QSPI demo

## Windows/Linux

	Link Script	Linux Address	M4 Boot Command
TCM	MCIMX6X_M4_tcm.ld	0x007F8000	bootaux 0x007F8000
OCRAM	MCIMX6X_M4_ocram.ld	0x00910000	bootaux 0x00910000
DDR	MCIMX6X_M4_ddr.ld	0x09ff00000	bootaux 0x09ff00000
QSPI AI	MCIMX6X_M4_qspi1b.ld	0x68000000	bootaux 0x68000000
QSPI SDB	MCIMX6X_M4_qspi2b.ld	0x78000000	bootaux 0x78000000

# Run FreeRTOS demo (Cont.)

- TCM Run

## **Limitation:**

The U-Boot on the Cortex-A9 core cannot load a Cortex-M4 image from SD Card to Cortex-M4 core's Tightly Coupled Memory (TCM) directly. This is a hardware limitation.

## **Workaround:**

Load the Cortex-M4 image from SD Card to OCRAM or DDR first and then copy the image from OCRAM or DDR to the Cortex-M4 core's TCM.

- a. `fatload mmc n:1 0x80000000 hello_world.bin`: Load the application image from the SD card to DDR RAM.  
(n=2 on SABRE-SD board and n=0 on SABRE-AI board)
- b. `dcache flush`: Flush cached content to DDR RAM.
- c. `cp.b 0x80000000 0x7F8000 0x8000`: Copy M4 image from DDR RAM to TCM.
- d. `dcache flush`: Flush cached content to TCM.
- e. `bootaux 0x7F8000`: Start the M4 core from the TCM.

# Compile Environment Troubleshooting

- ARMGCC DIR not set (windows)

```
C:\windows\system32\cmd.exe

C:\Freescall\FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc>cmake -DCMAKE_TOOLCHAIN_FILE="../../../tools/cmake_toolchain_files/armgcc.cmake" -G "MinGW Makefiles" -DCMAKE_BUILD_TYPE=Debug .
CMake Error at C:/Freescall/FreeRTOS_BSP_1.0.0_iMX6SX/tools/cmake_toolchain_files/armgcc.cmake:23 (MESSAGE):
***Please set ARMGCC_DIR in environment variables***
Call Stack (most recent call first):
  C:/Program Files (x86)/CMake/share/cmake-3.4/Modules/CMakeDetermineSystem.cmake:95 (include)
  CMakeLists.txt

CMake Error: CMAKE_C_COMPILER not set, after EnableLanguage
CMake Error: CMAKE_CXX_COMPILER not set, after EnableLanguage
-- Configuring incomplete, errors occurred!

C:\Freescall\FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc>mingw32-make -j4
mingw32-make: *** No targets specified and no makefile found.  Stop.

C:\Freescall\FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc>pause
Press any key to continue . . . _
```



# Compile Environment Troubleshooting(Cont.)

- ARMGCC\_DIR not set (linux)

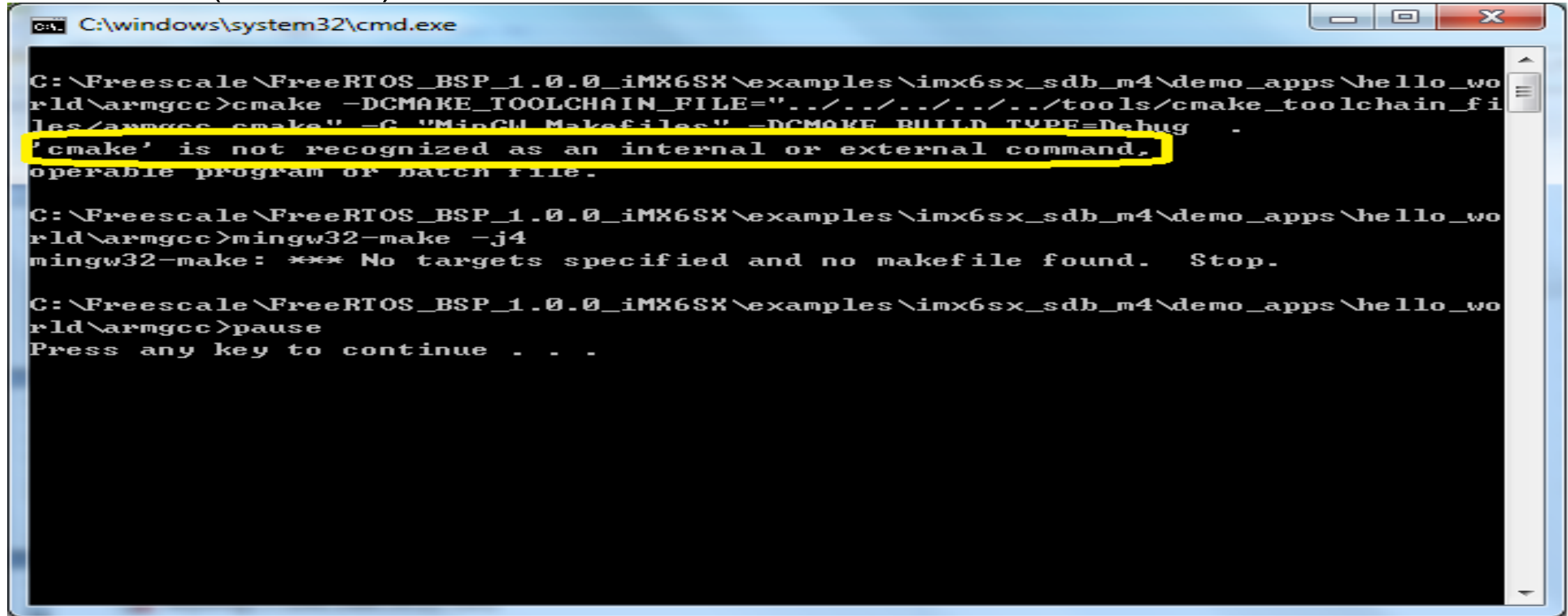
```
^C CMake Error at /home/...mp/FreeRTOS_BSP_1_0_0_iMX6SX/tools/cmake_toolchain_files/armgcc.cma
^E ***Please set ARMGCC_DIR in envionment variables***
^C Call Stack (most recent call first):
^E /usr/share/cmake-2.8/Modules/CMakeDetermineSystem.cmake:86 (INCLUDE)

^E

CMake Error: Error required internal CMake variable not set, cmake may be not be built correctl
Missing variable is:
^C CMAKE_C_COMPILER_ENV_VAR
CMake Error: Error required internal CMake variable not set, cmake may be not be built correctl
Missing variable is:
^C CMAKE_C_COMPILER
```

# Compile Environment Troubleshooting(Cont.)

- No cmake(windows)



```
C:\windows\system32\cmd.exe

C:\Freescale\FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc>cmake -DCMAKE_TOOLCHAIN_FILE="../../../../../../../../tools/cmake_toolchain_files/armgcc_cmake" -G "MinGW Makefiles" -DCMAKE_BUILD_TYPE=Debug .
'cmake' is not recognized as an internal or external command,
operable program or batch file.

C:\Freescale\FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc>mingw32-make -j4
mingw32-make: *** No targets specified and no makefile found.  Stop.

C:\Freescale\FreeRTOS_BSP_1.0.0_iMX6SX\examples\imx6sx_sdb_m4\demo_apps\hello_world\armgcc>pause
Press any key to continue . . .
```

# Compile Environment Troubleshooting(Cont.)

- No cmake(linux)

```
make: *** No targets specified and no makefile found.  Stop.  
/build debug.sh: 2: cmake: not found  
make: *** No targets specified and no makefile found.  Stop.
```

# Compile Environment Troubleshooting(Cont.)

- cmake version too low. But appears link issue.

```
-- Configuring done
-- Generating done
-- Build files have been written to: /home/.../FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/demo_apps/hello_world/armgcc
[ 4%] [ 8%] [ 13%] Building C object CMakeFiles/hello_world.dir/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/rtos/FreeRTOS/Source/portable/GCC/ARM_CM4F/port.c.obj
[ 17%] Building ASM object CMakeFiles/hello_world.dir/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/platform/devices/MCIMX6X/startup/gcc/startup_MCIMX6X_M4.S.obj
Building C object CMakeFiles/hello_world.dir/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/rtos/FreeRTOS/Source/portable/MemMang/heap_2.c.obj
Building C object CMakeFiles/hello_world.dir/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/demo_apps/hello_world/main.c.obj
/home/.../gcc-arm-none-eabi-4_9-2015q1/bin/.../lib/gcc/arm-none-eabi/4.9.3/../../../../arm-none-eabi/lib/armv7e-m/fpu/libg.a(lib_a-exit.o): In function `exit':
exit.c:(.text.exit+0x16): undefined reference to `_exit'
/home/.../gcc-arm-none-eabi-4_9-2015q1/bin/.../lib/gcc/arm-none-eabi/4.9.3/../../../../arm-none-eabi/lib/armv7e-m/fpu/crt0.o: In function `_start':
(.text+0x4a): undefined reference to `main'
/tmp/cc4akkN8.o: In function `Reset_Handler':
/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/platform/devices/MCIMX6X/startup/gcc/startup_MCIMX6X_M4.S:209: undefined reference to `SystemInit'
/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/platform/devices/MCIMX6X/startup/gcc/startup_MCIMX6X_M4.S:219: undefined reference to `__data_start__'
/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/platform/devices/MCIMX6X/startup/gcc/startup_MCIMX6X_M4.S:220: undefined reference to `__data_end__'
/tmp/cc4akkN8.o: In function `__isr_vector':
(.isr_vector+0x0): undefined reference to `__StackTop'
collect2: error: ld returned 1 exit status
make[2]: *** [CMakeFiles/hello_world.dir/home/.../FreeRTOS_BSP_1.0.0_iMX6SX/platform/devices/MCIMX6X/startup/gcc/startup_MCIMX6X_M4.S.obj] Error 1
make[2]: *** Waiting for unfinished jobs....
make[1]: *** [CMakeFiles/hello_world.dir/all] Error 2
make: *** [all] Error 2







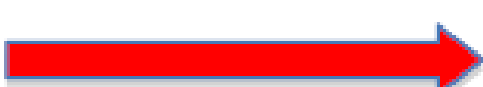

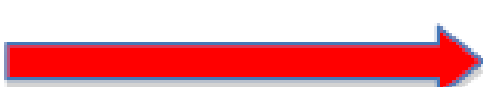



ig@debian6:~/FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/demo_apps/hello_world/armgcc$ cmake /V
cmake version 2.8.2
ig@debian6:~/FreeRTOS_BSP_1.0.0_iMX6SX/examples/imx6sx_sdb_m4/demo_apps/hello_world/armgcc$
```

**cmake version 2.8.2**

**Need cmake version 3.0.x**

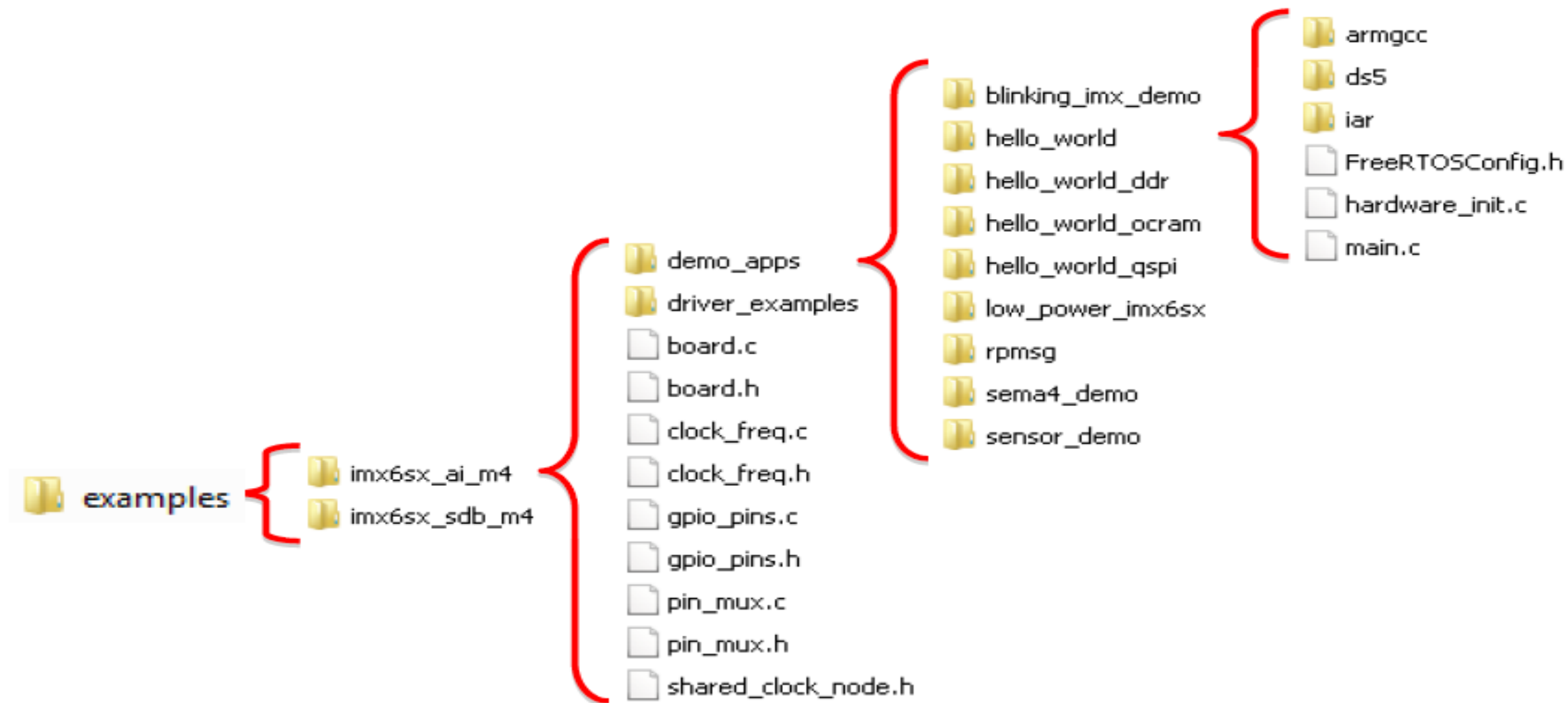


# Demo source files

-  doc
-  examples  Demo (example) source code and configuration files.
-  middleware  Shared middleware modules used by demos/examples.
-  platform  Shared, SoC-specific linker files, startup code and source for peripheral drivers, and useful services.
-  rtos  The source code of FreeRTOS kernel and its porting layer.
-  tools
-  LA\_OPT\_FSL\_OPEN\_3RD\_PARTY\_IP.htm
-  SW-Content-Register-FreeRTOS-BSP-1.0.0-i.MX65X.txt



# Demo source files(Cont.)



# With `_bm` and Without `_bm`

`_bm` --- `bm` for **b**are **m**etal

Example:

FreeRTOS\_BSP\_1.0.0\_iMX6SX/examples/imx6sx\_sdb\_m4/demo\_apps/rpmsg

common pingpong\_**bm** pingpong\_freertos str\_echo\_**bm** str\_echo\_freertos

# Remote Processor Messaging (RPMmsg)

## Kernel Configurations

### Location:

- > Device Drivers
- > Rpmmsg drivers

-\*- RPMSG bus driver

<M> IMX RPMSG pingpong driver -- loadable modules only

<M> IMX RPMSG tty driver -- loadable modules only

Note: Currently the driver is built as module ONLY.

# Remote Processor Messaging (RPMmsg) (Cont.)

Virtual TTY demo:

```
/unit_tests/mxc_mcc_tty_test.out /dev/ttyRPMMSG 115200 R 100 1000 &  
insmod imx_rpmsg_tty.ko  
imx_rpmsg_tty rpmsg0: new channel: 0x400 -> 0x1!  
Install rpmsg tty driver!  
echo deadbeaf > /dev/ttyRPMMSG  
imx_rpmsg_tty rpmsg0: msg(<- src 0x1) deadbeaf len 8
```

# Remote Processor Messaging (RPMsg) (Cont.)

```
COM13:115200baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help
insmod /rpmmsg/imx_rpmmsg_tty.ko
/unit_tests/mxc_mcc_tty_test.out /dev/ttyRPMMSG 115200 R 100 1000
&
echo 'Hello M4 !!!' > /dev/ttyRPMMSG

root@imx6sx_all:~# insmod /rpmmsg/imx_rpmmsg_tty.ko
imx_rpmmsg_tty rpmmsg0: new channel: 0x400 -> 0x0!
Install rpmmsg tty driver!
root@imx6sx_all:~# /unit_tests/mxc_mcc_tty_test.out /dev/ttyRPMMSG
000 &200 R 100 1
[1] 896
root@imx6sx_all:~# Serial port /dev/ttyRPMMSG opened
Speed set to 115200

root@imx6sx_all:~# echo 'Hello M4 !!!' > /dev/ttyRPMMSG
root@imx6sx_all:~# [1] READ finished :13 bytes

Hello M4 !!!

root@imx6sx_all:~#
```

```
COM14:115200baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help
RPMMSG String Echo FreeRTOS RTOS API Demo...
RPMMSG Init as Remote
Same service handshake is done, M4 has setup a rpmmsg ch
annel [0 ---> 1024]
Get Message From Master Side : "Hello M4 !!!" [len : 13
```



# Reference Documents

1. FreeRTOS\_BSP\_1.0.0\_i.MX\_6SoloX\_Release\_Notes
2. Getting\_Started\_with\_FreeRTOS\_BSP\_for\_i.MX\_6SoloX
3. FreeRTOS\_BSP\_i.MX\_6SoloX\_Demo\_User's\_Guide
4. Getting\_Started\_with\_Multicore\_Programming\_for\_i.MX\_6SoloX.pdf  
<https://community.freescale.com/docs/DOC-329540>





SECURE CONNECTIONS  
FOR A SMARTER WORLD