



MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

Version: 1.0
Release date: 2020-11-28

Use of this document and any information contained therein is subject to the terms and conditions set forth in [Exhibit 1](#). This document is subject to change without notice.

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

Version History

Version	Date	Description
1.0	2020-11-28	Official release

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

Table of Contents

Version History	2
Table of Contents.....	3
1 Getting Started	4
1.1 Overview	4
1.2 Code Layout.....	4
1.3 Build Image.....	5
1.4 Memory Layout	7
2 How-Tos	8
2.1 How to Modify FreeRTOS Config File	8
2.2 How to Modify Project Config File	8
2.3 How to Add Driver.....	8
2.4 How to Add CLI Command	8
2.5 How to Specify or Modify LSP	8
2.6 How to Check Code text/bss/data Size	9
2.7 How to Allocate Memory	9
2.8 How to Specify Function, Data, File to Normal DRAM Section	10
2.9 How to Specify lib*.a to the DRAM Section	10
2.10 How to Build One Static Library (*.a).....	10
2.11 How to Run or Stop HiFi 4 DSP	11
Exhibit 1 Terms and Conditions.....	12

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

1 Getting Started

This chapter introduces the MT7933 DSP FreeRTOS project integrated with the MT7933 IoT project and gives you an idea of what you need to prepare to get started.

1.1 Overview

The MT7933 is embedded with a single Cadence HiFi 4 DSP processor based on the Xtensa LX7 architecture.

1.2 Code Layout

In the MT7933 FreeRTOS source code, HIFI4_A means dsp core0 only. The code layout is as below.

```
<src_root>.
@./tinysys/adsp
+---- HIFI4
|
|   +---- build          --> Build system, including hifi4's environment variables, build
|   configurations, etc.
|   |   +---- Android.mk
|   |   +---- clear_do_vars.mk
|   |   +---- clear_feature_vars.mk
|   |   +---- clear_vars.mk
|   |   +---- config_common.mk
|   |   +---- definitions.mk
|   |   +---- do_binary.mk
|   |   +---- do_instance.mk
|   |   +---- feature.mk
|   |   +---- loader.mk
|   |   +---- main.mk
|   |   +---- register_do.mk
|   |   +---- register_feature.mk
|   |   +---- Makefile    --> Main Makefile
|   |   +---- build.sh    --> Build script
|
|   +---- drivers --> Common driver for hifi4dsp
|   |   +---- audio
|   |   +---- cli
|   |   +---- exception
|   |   +---- include
|   |   +---- ipi
|   |   +---- mem_mgt
|   |   +---- mpu
|   |   +---- printf
|   |   +---- wdt
|
|   +---- kernel
|   |   +---- FreeRTOS
|   |   |   +---- Source          --> FreeRTOS kernel code
|   |   |   |   +---- portable
|   |   |   |   |   +---- MemMang          --> Memory Management
|   |   |   |   |   |   +---- XCC
|   |   |   |   |   |   +---- Xtensa          --> Xtensa architecture code
|   |   |   +---- FreeRTOS-Plus  --> FreeRTOS plugins
|   |   |   +---- Source
```

1.3 Build Image

1. The MT7933 FreeRTOS project needs Cadence XCC toolchain. The Xtensa C and C++ Compiler (XCC) is an advanced optimizing compiler for all Xtensa processors. But the XCC compiler must be authorized by Cadence.

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

2. After installing the XCC toolchain, you can modify the toolchain configuration in the files listed below:

<src_root>/tinysys/adsp/HIFI4/project/mt7933/platform/build/env_hifi4.mk

<src_root>/tinysys/adsp/HIFI4/project/mt7933/platform/hwcfg/RI-2019.1-

linux/hifi4_MockingBird_PROD/config/hifi4_MockingBird_PROD-params

2.1 <src_root>/tinysys/adsp/HIFI4/project/mt7933/platform/build/env_hifi4.mk

```
#####
## Toolchain
#####
ifeq (,$(ADSP_CROSS_COMPILE))
  XT_TOOLS_VERSION ?= RI-2019.1-linux
  XPLODER_VER := Xplorer-8.0.10
  XTENSA_CORE := hifi4_MockingBird_PROD
  # Configurable XT related paths
  XT_TOOLS := /mtkeda/xtensa/$(XPLODER_VER)/XtDevTools/install/tools/$(XT_TOOLS_VERSION)
  XT_TOOLS_DIR := $(XT_TOOLS)/XtensaTools
  XT_HWREG_DIR ?= $(PLATFORM_DIR)/hwcfg/$(XT_TOOLS_VERSION)/$(XTENSA_CORE)
  XT_LIB_DIR := $(XT_HWREG_DIR)/xtensa-elf/lib
  XTENSA_SYSTEM := $(XT_HWREG_DIR)/config
  XT_TOOLS_BIN_DIR := $(XT_TOOLS_DIR)/bin
  ADSP_CROSS_COMPILE := $(XT_TOOLS_BIN_DIR)/
endif
```

Note: The figure just shows you the lines you need to modify. For example, modify the white line “XT_TOOLS := ...” to where your XCC toolchain is located.

2.2 <src_root>/tinysys/adsp/HIFI4/project/mt7933/platform/hwcfg/RI-2019.1-

linux/hifi4_MockingBird_PROD/config/hifi4_MockingBird_PROD-params

```
# File locations
# Note: Relative paths are relative to the location of the parameter file.
install-prefix = /mtkeda/xtensa/Xplorer-8.0.10/XtDevTools/install/tools/RI-2019.1-linux/XtensaTools
config-prefix = ../
xtensa-tools = /mtkeda/xtensa/Xplorer-8.0.10/XtDevTools/install/tools/RI-2019.1-linux/XtensaTools/Tools
tc-tools = /mtkeda/xtensa/Xplorer-8.0.10/XtDevTools/install/tools/RI-2019.1-linuxXtensaTools/TIE
isa-base-dlls = [
  libisa-core-hw.so
  libisa-core.so
]
```

Note: The figure just shows you the lines you need to modify. For example, modify the white line “XT_TOOLS := ...” to where your XCC toolchain is located.

3. Configure LICENSE file

Create a ~/.flexlmrc file in your home directory and add “XTENSAD_LICENSE_FILE” variable to specify the toolchain license server, for example:

XTENSAD_LICENSE_FILE=7400@mtklc17

4. Build command

4.1 Build dsp binary

4.1.1 Enter <src_root>/tinysys/adsp/HIFI4/

4.1.2 Run “sh build/build.sh iot7933bga-hadron” to build

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

```
input_f: ../out/iot7933bga-hadron/hifi4dsp_load.bin
output_f: ../../../../driver/chip/mt7933/src/hifi4dsp/hifi4dsp_load/inc/hifi4dsp_load.h
```

This command updates the prebuilt binary, hifi4dsp_load.bin, under prebuilt/driver/chip/mt7933

1.4 Memory Layout

The memory map of MT7933DSP includes two sections: SRAM and DRAM, which are defined at LSP

<src_root>/tinysys/adsp/HIFI4/project/mt7933/platform/build/lnk-hifi4/memmap.xmm.

For example

SRAM:

	AP view	(DSP view)
-----	0x4004_0000	(0XA000_0000)
256K + 48KB (text/vector/data/bss/stack/heap)		
(borrow 48KB from AFE SRAM)		
-----		(0xA004_B7FF)
PSRAM(dsp_sysram): 4MB/8MB (non-UHSPSRAM), 4MB/8MB (UHSPSRAM)		
-----	0xA005_8000	(0xA005_0000)
1,152KB (text/data/bss)		
-----		(0xA017_0000)
848KB (Cached heap)		
-----		(0xA024_4000)
112KB (Non-Cached heap)		
-----		(0xA026_0000)
2MB (reserved)		
shared buffer with arm		
depend on PSRAM size		
-----		(0xA046_0000)

2 How-Tos

2.1 How to Modify FreeRTOS Config File

`<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/drivers/HIFI4_*/main/inc/FreeRTOSConfig.h` is mainly to configure FreeRTOS kernel's related configurations.

2.2 How to Modify Project Config File

1. `<src_root>/tinysys/adsp/HIFI4/project/MT7933/config/<project>/HIFI4_A/ProjectConfig.mk` is mainly to configure this project's related configurations.
2. `<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/build/platform.mk` is mainly to configure all projects' common configurations.

2.3 How to Add Driver

1. If the new driver is for cross-SoC uses, add it under `<src_root>/HIFI4/drivers` folder
2. If the new driver is MT7933DSP's own driver, add it under `<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/drivers/HIFI4_A` folder
3. Append new driver's *.c to `C_FILES` at `<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/build/platform.mk`
4. Append new driver header file path to `INCLUDES` at `<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/build/platform.mk` to let others invoke.
5. Add feature config at `<src_root>/tinysys/adsp/HIFI4/project/MT7933/config/<project>/HIFI4_A/ProjectConfig.mk` to control if needed.

2.4 How to Add CLI Command

1. Define `CLI_Command_Definition_t` static variable in your own driver
2. Call `FreeRTOS_CLIRRegisterCommand()` to register CLI command in your own driver init function

2.5 How to Specify or Modify LSP

1. If there is no `lnk-hifi4` under `<src_root>/tinysys/adsp/HIFI4/project/MT7933/config/<project>/HIFI4_A/`, use `<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/build/lnk-hifi4`.

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

2. Modify memmap.xmm under LSP's location, and then use the **xt-genldscripts** command to generate new LSP.

2.6 How to Check Code text/bss/data Size

Please refer to `<src_root>/tinysys/adsp/HIFI4/out/<project>/HIFI4_A/hifi4_*.bin` size or use the "xt-size" command to check.

2.7 How to Allocate Memory

1. MT7933DSP FreeRTOS adopts heap_4 memory management; therefore, the `pvPortMalloc()/vPortFree()` functions provided by FreeRTOS kernel adopts global heap array `ucHeap[configTOTAL_HEAP_SIZE]` as memory allocation heap.
2. xlibc library (c library) provides standard `malloc()/free()` functions, which defines the heap at LSP `memmap.xmm`

```
sram4 : F : 0x400205bc - 0x400205db : .DebugExceptionVector.text .KernelExceptionVector.literal;
sram5 : F : 0x400205dc - 0x400205fb : .KernelExceptionVector.text .UserExceptionVector.literal;
sram6 : F : 0x400205fc - 0x4002061b : .UserExceptionVector.text .DoubleExceptionVector.literal;
sram7 : F : 0x4002061c - 0x4005f7ff : STACK : HEAP : .DoubleExceptionVector.text .sram.rodata .sram.
.rodata .rtos.percpu.data .data .rtos.bss .rtos.percpu.bss .sram.bss .bss;
END sram
```

The HEAP overlaps the STACK completely. The heap generally grows upwards while the stack always grows downwards. This way, they start at opposite ends of the segment and meet in the middle.

3. MT7933DSP FreeRTOS also provides another 2 kinds of heaps to allocate memory. Use following functions to allocate or free memory.

Must set `CFG_MTK_HEAP_SUPPORT = yes` firstly in `ProjectConfig.mk`.

```
#include "mtk_heap.h"

void *MTK_pvPortMalloc(size_t xWantedSize, MTK_eMemoryType eMemoryType);
void MTK_vPortFree(void *pv);

typedef enum {
    MTK_eMemDefault = 0,
    MTK_eMemDramNormal,
    MTK_eMemDramNormalNC,
    MTK_eMemInvalid
} MTK_eMemoryType;
```

The following table shows allocation heap's description:

Memory Type	Description
MTK_eMemDefault	Same as <code>pvPortMalloc()</code> provided by FreeRTOS kernel
MTK_eMemDramNormal	<ol style="list-style-type: none"> 1. On the DRAM memory 2. Use in normal power mode

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

Memory Type	Description
	3. May not be accessed in sleep mode 4. Cacheable
MTK_eMemDramNormalNC	1. On the DRAM memory 2. Use in normal power mode 3. May not be accessed in sleep mode 4. Non-cacheable

2.8 How to Specify Function, Data, File to Normal DRAM Section

1. Add **NORMAL_SECTION_FUNC** keyword to function definition or declare; this function will be compiled into (".dram.text") and (".dram.rodata") sections.
2. Add **NORMAL_SECTION_DATA** keyword to variable definition that has a initial value; this variable will be compiled into (".dram.data") section.
3. Add **NORMAL_SECTION_BSS** keyword to variable definition that has no initial value; this variable will be compiled into (".dram.bss") section.
4. Add **NORMAL_SECTION_RODATA** keyword to variable definition that is const; this variable will be compiled into (".dram.rodata") section.
5. To compile a source code file into normal section(.dram.* section), please append the source code file to **NORMAL_SECTION_C_FILES** in

`<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/build/platform.mk`

For example, **NORMAL_SECTION_C_FILES** += \$(DRIVERS_COMMON_DIR)/mpu/mpu.c

2.9 How to Specify lib*.a to the DRAM Section

To link a 3rd-party library into the normal section(.dram.* section), please append the library to **NORMAL_SECTION_LIBS** in `<src_root>/tinysys/adsp/HIFI4/project/MT7933/platform/build/platform.mk`

For example,

`INCLUDES += $(PLATFORM_DIR)/middleware/lib/inc`

NORMAL_SECTION_LIBS += \$(PLATFORM_DIR)/middleware/lib/example/libexample.a

2.10 How to Build One Static Library (*.a)

Use the following Makefile sample to build, command: `make lib=libxxx.a` (**Recommended**)

This **Makefile** is located at `<source_dir>/tinysys/adsp/HIFI4/project/MT7933/platform/tools/hifi4tools/`; you can copy this file to the directory in parallel with the library source folder.

The following example uses the source files under the folder **imbSwVad_20190214** to build a libimb.a.

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

```
[mtk05764@mhfsdclx026 hifi4tools]$ ls -lR imbSwVad_20190214
imbSwVad_20190214:
total 16
drwxr-xr-x 1 mtk05764 mtk05764 8 May 29 14:24 subdir
-rw-r--r-- 1 mtk05764 mtk05764 801 Feb 14 13:19 swVAD.c
-rw-r--r-- 1 mtk05764 mtk05764 536 Feb 14 13:19 swVAD.h
-rw-r--r-- 1 mtk05764 mtk05764 3765 Feb 14 15:15 swVAD_impl.c
-rw-r--r-- 1 mtk05764 mtk05764 1085 Feb 14 13:18 swVAD_impl.h

imbSwVad_20190214/subdir:
total 4
-rw-r--r-- 1 mtk05764 mtk05764 23 May 29 11:29 kk.c
[mtk05764@mhfsdclx026 hifi4tools]$
```

You can add the following configuration items to Makefile, and then *make lib=libimb.a*

```
#####
-C_FILES := test.c
+C_FILES := swVAD.c \
+          swVAD_impl.c \
+          kk.c
S_FILES :=
-INCLUDES :=
+INCLUDES := imbSwVad_20190214
C_DEF :=
-SRC_DIRS :=
+SRC_DIRS := imbSwVad_20190214 : imbSwVad_20190214/subdir/
```

2.11 How to Run or Stop HiFi 4 DSP

Boot to Cortex-M33 shell and init psram, and then use *adsp cli* to start or stop HiFi 4 DSP

```
$ adsp adsp_poweron
```

```
$ adsp adsp_shutdown
```

MT793X IoT SDK for HiFi 4 DSP FreeRTOS User Guide

Exhibit 1 Terms and Conditions

Your access to and use of this document and the information contained herein (collectively this “Document”) is subject to your (including the corporation or other legal entity you represent, collectively “You”) acceptance of the terms and conditions set forth below (“T&C”). By using, accessing or downloading this Document, You are accepting the T&C and agree to be bound by the T&C. If You don’t agree to the T&C, You may not use this Document and shall immediately destroy any copy thereof.

This Document contains information that is confidential and proprietary to MediaTek Inc. and/or its affiliates (collectively “MediaTek”) or its licensors and is provided solely for Your internal use with MediaTek’s chipset(s) described in this Document and shall not be used for any other purposes (including but not limited to identifying or providing evidence to support any potential patent infringement claim against MediaTek or any of MediaTek’s suppliers and/or direct or indirect customers). Unauthorized use or disclosure of the information contained herein is prohibited. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for Your unauthorized use or disclosure of this Document, in whole or in part.

MediaTek and its licensors retain titles and all ownership rights in and to this Document and no license (express or implied, by estoppels or otherwise) to any intellectual propriety rights is granted hereunder. This Document is subject to change without further notification. MediaTek does not assume any responsibility arising out of or in connection with any use of, or reliance on, this Document, and specifically disclaims any and all liability, including, without limitation, consequential or incidental damages.

THIS DOCUMENT AND ANY OTHER MATERIALS OR TECHNICAL SUPPORT PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT, IF ANY, ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, COMPLETENESS OR ACCURACY AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MEDIATEK SHALL NOT BE RESPONSIBLE FOR ANY MEDIATEK DELIVERABLES MADE TO MEET YOUR SPECIFICATIONS OR TO CONFORM TO A PARTICULAR STANDARD OR OPEN FORUM.

Without limiting the generality of the foregoing, MediaTek makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does MediaTek assume any liability arising out of the application or use of any product, circuit or software. You agree that You are solely responsible for the designing, validating and testing Your product incorporating MediaTek’s product and ensure such product meets applicable standards and any safety, security or other requirements.

The above T&C and all acts in connection with the T&C or this Document shall be governed, construed and interpreted in accordance with the laws of Taiwan, without giving effect to the principles of conflicts of law.