

FreeRTOS™ BSP v.1.0.0 for i.MX 7Dual Release Notes

1 Overview

These are the release notes for the FreeRTOS™ BSP 1.0.0 for the i.MX 7Dual ARM® Cortex®-M4 core. The core of the BSP is a set of peripheral drivers.

The peripheral driver wraps the hardware register accesses into a set of stateless functional primitives that provide the building blocks for high-level applications.

The FreeRTOS BSP includes a set of example applications demonstrating the use of the peripheral drivers and some demonstrations of integrated FreeRTOS operating system (OS) v8.0.0. The FreeRTOS BSP also integrates RPMsg middleware to provide an easy-to-use multi-core communication API for i.MX product families.

Contents

1	Overview.....	1
2	What Is New	2
3	Development Tools.....	3
4	Supported Development Systems	4
5	Release Contents	5
6	FreeRTOS BSP Release Overview	6
6.1	i.MX 7Dual Cortex-M4 platform support.....	6
6.2	Demo applications	7
6.3	Driver examples	7
6.4	Other integrated software solutions	7
7	Known Issues	9
8	Revision History.....	10

2 What Is New

These are the new features for FreeRTOS BSP 1.0.0 i.MX 7Dual:

- Added Peripheral support:
 - i.MX 7Dual ADC
 - i.MX 7Dual CCM
 - i.MX GPIO
 - i.MX I2C
 - i.MX MU
 - i.MX UART
 - i.MX WDOG
 - ECSPI
 - FlexCAN
 - GPT
 - Resource Domain Control (RDC)
 - SEMA4
- Added Multi-core communication support:
 - RMPmsg

3 Development Tools

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

- IAR Embedded Workbench for ARM[®] version 7.30.4
- Makefiles support with GCC revision 4.8-2014-q3-update from ARM Embedded
- ARM Development Studio 5 (DS-5[™]) version 5.20.1 (32 bit)
- Lauterbach TRACE32 PowerView for ARM version R.2014.09.000058270

4 Supported Development Systems

This release was tested with boards and devices listed in this table.

Table 2 Supported MCU devices and development boards

Development boards	i.MX devices
i.MX 7Dual SABRE board MCIMX7x-SDB (Rev.A)	PCIMX7D7DVM10SA
i.MX 7Dual Validation board XM7DLPDDR312-SS (Rev.A)	PCIMX7D7DVK10SA

5 Release Contents

This table describes the release contents.

Table 3 Release contents

Deliverable	Location
Examples	<install_dir>/examples/...
Demo applications	<install_dir>/examples/<board_name>/demo_apps/...
Driver examples	<install_dir>/examples/<board_name>/driver_examples/...
Documentation	<install_dir>/doc/...
Middleware	<install_dir>/middleware/...
RPMMsg stacks	<install_dir>/middleware/multicore/open-amp/...
Driver library, startup code and utilities	<install_dir>/platform/...
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source	<install_dir>/platform/CMSIS/...
Linker control files for each supported tool chain	<install_dir>/platform/devices/<device_name>/linker/...
SoC header files, extension header files, and feature header files	<install_dir>/platform/devices/<device_name>/include
CMSIS-compliant startup code	<install_dir>/platform/ devices/<device_name>/startup/...
Peripheral Drivers	<install_dir>/platform/drivers/...
Utilities such as debug console	<install_dir>/platform/utilities/...
RTOS Kernel folder	<install_dir>/rtos/...
FreeRTOS Kernel Code	<install_dir>/rtos/FreeRTOS/...
External tools for build and debug	<install_dir>/tools/...

6 FreeRTOS BSP Release Overview

The FreeRTOS BSP is intended for use with Freescale's i.MX product family on the ARM Cortex-M4 core. The release consists of:

- i.MX 7Dual ARM Cortex-M4 platform support (ARM Cortex-A7 is not supported)
- Demo applications/Driver examples
- Multi-core communication stack (RPMsg)
- FreeRTOS kernel v8.0.0
- Documentation (*FreeRTOS BSP i.MX 7Dual API Reference Manual* and various user's guides)

6.1 i.MX 7Dual Cortex-M4 platform support

The FreeRTOS BSP platform directory contains the startup code, driver libraries for peripherals, header files, linker files, and utilities such as the debug console implementation.

6.1.1 Startup code

The FreeRTOS BSP includes simple CMSIS-compliant startup code for i.MX 7Dual which efficiently delivers the code execution to the `main()` function. An application can include the startup code directly in the project build environment for a cleaner project build environment.

6.1.2 Driver library

The FreeRTOS BSP provides a set of drivers for the i.MX family on-chip peripherals. The drivers are designed and implemented around the peripheral hardware blocks rather than for a specific i.MX device, and work with or without FreeRTOS OS.

The drivers are designed to wrap hardware register accesses into functional accesses. They are stateless and are intended to cover the most frequently used hardware functionality. They are written in C language, which increases the reuse of code for other products as they are designed to be initialized at runtime based on the custom configuration depending on the product. For more details, see chapter "Architectural Overview" in the *FreeRTOS BSP i.MX 7Dual API Reference Manual* (FRTOS7DAPIRM).

6.1.3 Header files

The FreeRTOS BSP devices directory contains device-specific header files that provide direct access to the i.MX peripheral registers. Each supported i.MX device in the FreeRTOS BSP has an overall System-on-Chip (SoC) memory-mapped header file. Along with the SoC header files, the FreeRTOS™ BSP CMSIS directory includes common CMSIS header files for the ARM Cortex-M core and DSP library from the ARM CMSIS version 4.2 release.

6.1.4 Linker files

The FreeRTOS BSP devices directory contains linker control files (or simply linker files) for each supported tool chain and i.MX device.

6.1.5 Utilities

The utilities directory contains useful software utilities such as a debug console.

6.1.6 RDC settings

i.MX 7Dual is a multi-core device and Resource Domain Control (RDC) could be used to grant peripheral and/or memory access permissions to each core.

The examples and demo applications in the FreeRTOS BSP make use of RDC to allocate peripheral access permission. When running the ARM Cortex-A7 application with FreeRTOS BSP example/demo, it is important to respect the reserved peripheral. FreeRTOS BSP application has reserved peripherals that are used only by ARM Cortex-M4, and any access from ARM Cortex-A7 on those peripherals may cause the program to hang.

The default RDC setting is:

- ARM Cortex-M4 core is assigned to RDC domain 1, and ARM Cortex-A7 and other bus masters use the default assignment (RDC domain 0)
- Every example/demo has its specific RDC setting in its hardware_init.c. Most of them are set to exclusive access.

The user of this package can remove or change the RDC settings in the example/demo or in his application.

6.2 Demo applications

The demo applications demonstrate the usage of the drivers and other integrated software solutions along with FreeRTOS OS on i.MX 7Dual. For details, see the *FreeRTOS BSP i.MX 7Dual Demo Applications User's Guide* (FRTOS7DDAUG).

6.3 Driver examples

The driver examples demonstrate the usage of drivers on bare metal environment. For details, see the *FreeRTOS BSP i.MX 7Dual Demo Applications User's Guide* (FRTOS7DDAUG).

6.4 Other integrated software solutions

The FreeRTOS BSP is designed for easy integration with other software solutions such as FreeRTOS OS and RPMsg multi-core communication.

6.4.1 RTOS

The FreeRTOS BSP is pre-integrated with the FreeRTOS OS (www.freertos.org) v8.0.0.

6.4.2 RPMSG

A multi-core communication stack RPMsg is integrated with FreeRTOS BSP and can be used to communicate with the i.MX ARM Cortex-A7 core. The stack is from the github project OpenAMP (github.com/OpenAMP/open-amp) and has been ported to this FreeRTOS BSP release. Besides the RPMsg API, no other feature from OpenAMP is included in this release. For more information, see the OpenAMP manuals under <install_dir>/middleware/multicore/open-amp/docs.

7 Known Issues

Table 4 Known issues

Issue	Workaround
When running the FreeRTOS sensor demo image, the Linux boot fails with the imx7d-sdb-m4.dtb.	Disable the sensor support in the Linux imx7d-sdb-m4.dts file. Because all the sensors are connected by I2C2, the simple way is to disable the I2C2 in the Linux imx7d-sdb-m4.dts.

8 Revision History

This table summarizes revisions to this document.

Revision number	Date	Substantial changes
0	08/2015	Initial release.

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

www.freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Tower is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM, ARM powered logo, Keil, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2015 Freescale Semiconductor, Inc. All rights reserved.

