AN13970

Running Zephyr RTOS on Cadence Tensilica HiFi 4 DSP

Rev. 2 — 28 November 2023 Application note

Document information

Information	Content
Keywords	AN13970, i.MX 8M Plus, HiFi 4 DSP, Zephyr, IPC, OpenAMP, remoteproc
Abstract	This document explains how to harness the power processing of HiFi 4 DSP available in the NXP i.MX 8M Plus processor by running Zephyr RTOS on the DSP.

1 Introduction

Running Zephyr on Arm Cortex-A or Cortex-M cores is widely discussed and there are many examples on how to implement it. However, many Cortex-based microcontroller units (MCUs) and microprocessor units (MPUs) have on-chip digital signal processors (DSPs) incorporated to offload compute-intensive tasks.

The Cadence Tensilica HiFi 4 DSP is one such example of a high-performance embedded DSP optimized for audio, voice, or neural network processing. This application note explains how to harness the power processing of the HiFi 4 DSP available in the NXP i.MX 8M Plus processor, by running Zephyr real-time operating system (RTOS) on the DSP; while Linux operating system (OS) runs on the main Cortex-A core.

Using example applications, this document explains:

- How to launch the applications on the HiFi 4 DSP
- How the HiFi 4 DSP and the main processor core communicate to each other
- How to get the output of the applications

In this document, all the examples are explained using existing drivers and/or frameworks from Linux OS and Zephyr RTOS.

2 Hardware platform

The i.MX 8M Plus EVK board is based on the NXP i.MX 8M Plus applications processor, which is composed of:

- 4x Arm Cortex-A53 up to 1.8 GHz
- 1x Arm Cortex-M7 up to 800 MHz
- Cadence Tensilica HiFi 4 DSP up to 800 MHz

Figure 1 shows the top view of the i.MX 8M Plus EVK board.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

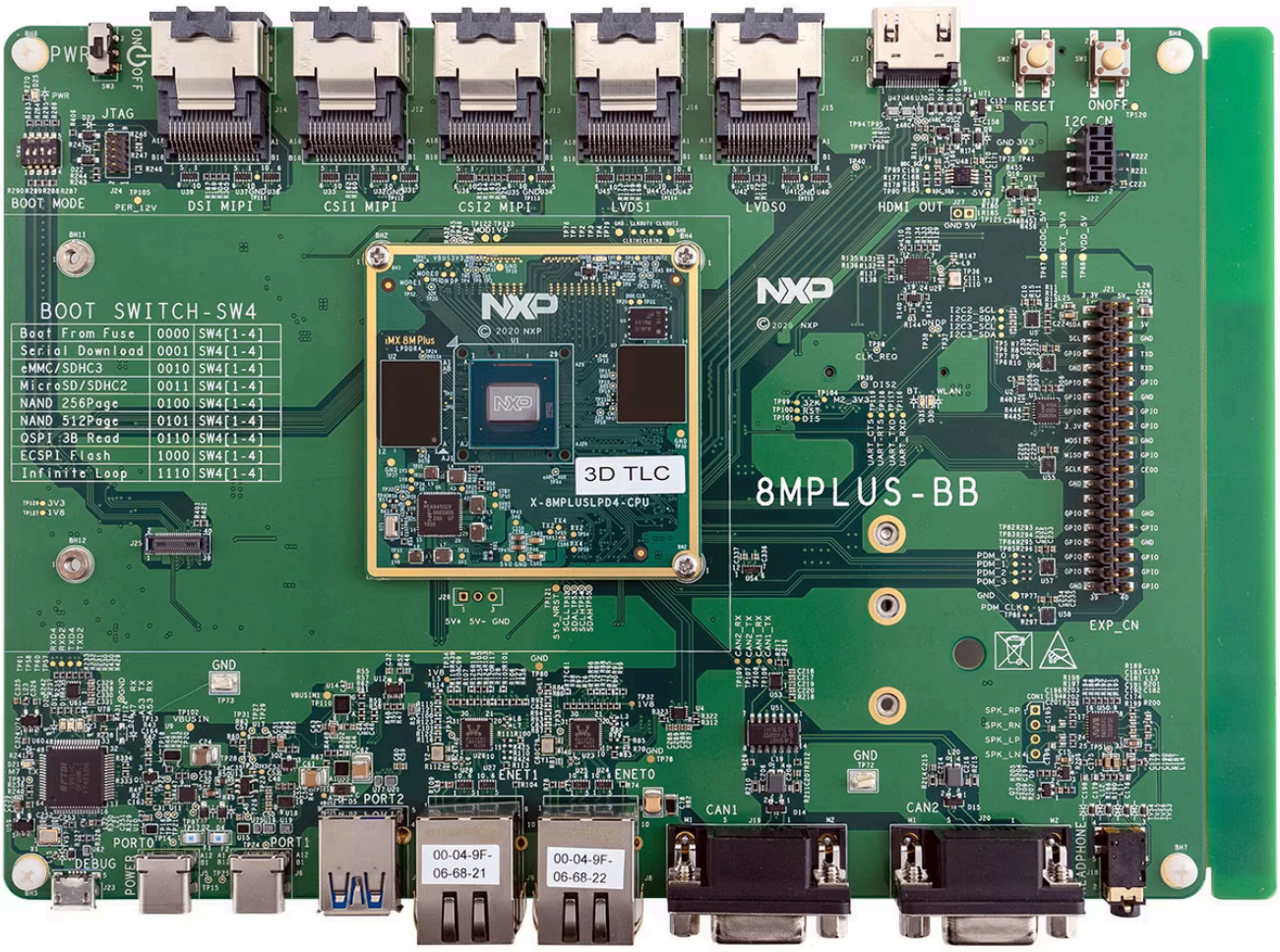

Figure 1. i.MX 8M Plus EVK board top view

For more details on the i.MX 8M Plus EVK board, see i.MX 8M Plus EVK.

3 Zephyr OS

The Zephyr Project is a scalable real-time operating system (RTOS) supporting multiple hardware architectures, optimized for resource-constrained devices, and built with security in mind. It is based on a small-footprint kernel designed for use on resource-constrained systems.

NXP offers various evaluation and prototyping platforms that the Zephyr OS can support. Developers are able to tailor a solution easily to meet their needs using a true open source project with hardware, developer tools, and sensor and device drivers. Security enhancements with Zephyr OS enable easy implementation of device management, connectivity stacks, and file systems.

For more details on the Zephyr RTOS, visit www.zephyrproject.org/.

4 HiFi 4 audio DSP

The HiFi 4 Audio Engine is a highly optimized audio processor geared for efficient execution of audio and voice codecs and pre- and post-processing modules.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

In Zephyr, the board that supports the audio DSP from i.MX 8M Plus is nxp_adsp_imx8m.

4.1 Supported features

The Zephyr nxp_adsp_imx8m board configuration supports the hardware features shown in Table 1.

Table 1. Supported hardware features

Interface	Controller	Driver/component
SYSTICK	On-chip	systick
CLOCK	On-chip	clock_control
PINMUX	On-chip	pinmux
UART	On-chip	serial port-polling

Note: *The port does not support other hardware features currently.*

The default configuration can be found in the defconfig file, boards/xtensa/nxp_adsp_imx8m/nxp_adsp_imx8m_defconfig.

4.2 Connections and I/Os

The i.MX 8M Plus EVK board is tested with the pinmux controller configuration shown in Table 2.

Table 2. Connections

Board name	SoC name	Usage
UART4 RXD	UART4_TXD	UART console
UART4 TXD	UART4_RXD	UART console

4.3 System clock

The HiFi 4 DSP core is configured to run at 800 MHz clock speed.

4.4 Serial port

The i.MX 8M Plus SoC has four Universal Asynchronous Receiver/Transmitters (UARTs). Only UART_4 is configured for the DSP console and the remaining UARTs are not used/tested.

5 Building and running Zephyr samples on HiFi 4 DSP

This section describes how to build and run Zephyr samples on HiFi 4 DSP using the following two applications:

- Section 5.1
- Section 5.2

5.1 hello_world application

The Zephyr's hello_world application is a simple sample that can be used with one of the Supported boards and prints "Hello World" to the console.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

5.1.1 Load hello_world application on DSP

To load the hello_world application on the DSP, use the Linux remoteproc driver.

In Linux, a generic i.MX remoteproc driver and a DSP-specific driver (imx_dsp_rproc) are already available.

Because the application is running on the DSP, use the imx_dsp_rproc driver. To use the driver, enable `CONFIG_IMX_DSP_REMOTEPROC` in the Linux kernel.

5.1.2 Compile hello_world application

Compile the hello_world application in Zephyr for the i.MX 8M Plus DSP, it means, compile the application for the nxp_adsp_imx8m board.

Go to the `zephyr/` folder from `zephyrproject` and run:

```
~/zephyrproject/zephyr$ west build -p always -b nxp_adsp_imx8m samples/
hello_world/
~/zephyrproject/zephyr$
~/zephyrproject/zephyr$ ls -la build/zephyr
total 4288
drwxr-xr-x 14 user nxp   4096 Oct 17 17:05 .
drwxr-xr-x  7 user nxp   4096 Oct 17 17:05 ..
drwxr-xr-x  5 user nxp   4096 Oct 17 17:05 arch
drwxr-xr-x  3 user nxp   4096 Oct 17 17:05 boards
drwxr-xr-x  5 user nxp   4096 Oct 17 17:05 cmake
-rw-r--r--  1 user nxp     64 Oct 17 17:05 .cmake.dotconfig.checksum
drwxr-xr-x  6 user nxp   4096 Oct 17 17:05 CMakeFiles
-rw-r--r--  1 user nxp  12355 Oct 17 17:05 cmake_install.cmake
-rw-r--r--  1 user nxp  39648 Oct 17 17:05 .config
...
-rw-r--r--  1 user nxp   2275 Oct 17 17:05 zephyr.dts
-rw-r--r--  1 user nxp    619 Oct 17 17:05 zephyr.dts.d
-rw-r--r--  1 user nxp 124460 Oct 17 17:05 zephyr.dts.pre
-rwxr-xr-x  1 user nxp 715896 Oct 17 17:05 zephyr.elf
-rw-r--r--  1 user nxp 408374 Oct 17 17:05 zephyr_final.map
-rw-r--r--  1 user nxp 408374 Oct 17 17:05 zephyr.map
-rwxr-xr-x  1 user nxp 717052 Oct 17 17:05 zephyr_pre0.elf
-rw-r--r--  1 user nxp 408886 Oct 17 17:05 zephyr_pre0.map
-rw-r--r--  1 user nxp   7273 Oct 17 17:05 zephyr.stat
```

The `zephyr.elf` file is used as the firmware to be loaded on the DSP.

5.1.3 Run hello_world application on DSP

The subsections that follow explain how to run the hello_world application on HiFi 4 DSP from i.MX 8M Plus.

5.1.3.1 Start i.MX 8M Plus EVK board

Start the i.MX 8M Plus EVK board with a specific device tree source (DTS).

Use `imx8mp-evk-dsp.dtb`, and after inserting the `imx_dsp_rproc.ko` kernel module, you get:

```
root@imx8mpevk:~# ls -la /sys/class/remoteproc/
total 0
drwxr-xr-x  2 root root 0 Mar  3 09:49 .
drwxr-xr-x 90 root root 0 Mar  3 09:49 ..
lrwxrwxrwx  1 root root 0 Mar  3 09:54 remoteproc0 -> ../../devices/
platform/3b6e8000.dsp/remoteproc/remoteproc0
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
root@imx8mpevk:~# cat /sys/class/remoteproc/remoteproc0/firmware
imx/dsp/hifi4.bin
root@imx8mpevk:~#
```

Here, `remoteproc0`, which is for DSP, is used.

5.1.3.2 Check firmware on board

Check the firmware image on the board:

```
root@imx8mpevk:~# ls -la /lib/firmware/imx/zephyr/
total 1256
drwxr-xr-x  2 root root   4096 Mar  3 11:03 .
drwxr-xr-x 12 root root   4096 Mar  9  2018 ..
-rwxr-xr-x  1 root root  41520 Mar  9  2018 imx8-hello_world-zephyr.elf
-rwxr-xr-x  1 root root  57100 Mar  9  2018 imx8m-hello_world-zephyr.elf
-rwxr-x---  1 root root 996276 Mar  3 10:38 imx8m-openamp_rsc_table-zephyr.elf
-rwxr-xr-x  1 root root  87876 Mar  9  2018 imx8m-philosophers-zephyr.elf
-rwxr-xr-x  1 root root  58124 Mar  9  2018 imx8m-synchronization-zephyr.elf
-rwxr-xr-x  1 root root  41520 Mar  9  2018 imx8x-hello_world-zephyr.elf
root@imx8mpevk:~#
```

The firmware must be present in `/lib/firmware` before the remoteproc driver is probed; however, it can also be given with an absolute path.

5.1.3.3 Insert `imx_dsp_rproc.ko` kernel module

By default, the i.MX DSP remoteproc protocol waits for a READY reply from the remote processor. Because not all Zephyr sample applications (especially simple applications that do not use the mailbox) send a READY reply, you must use the remoteproc module `imx_dsp_rproc.ko` without waiting for a reply. The `imx_dsp_rproc.ko` module is implemented using the kernel module parameter `no_maiboxes`, as shown below:

```
root@imx8mpevk:~# modinfo imx_dsp_rproc
filename:       /lib/modules/6.1.55-02981-g63bd8fa873a2/kernel/drivers/
remoteproc/imx_dsp_rproc.ko
author:         Shengjiu Wang <shengjiu.wang@nxp.com>
description:    i.MX HiFi Core Remote Processor Control Driver
license:        GPL v2
...
depends:
intree:         Y
name:           imx_dsp_rproc
parm:           no_mailboxes:There is no mailbox between cores, so ignore remote
 proc reply after start, default is 0 (off). (int)
root@imx8mpevk:~#
```

By default, the `no_maiboxes` parameter is off — do not ignore the reply from the remote processor.

Therefore, first check the `imx_dsp_rproc` parameter. If it is off, remove the module and insert it with the right parameter.

```
root@imx8mpevk:~# grep -H '' /sys/module/imx_dsp_rproc/parameters/* /
*no_mailboxes param is off */
/sys/module/imx_dsp_rproc/parameters/no_mailboxes:0
root@imx8mpevk:~#
root@imx8mpevk:~# rmmod imx_dsp_rproc /* remove kernel module */
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
[ 797.922929] remoteproc remoteproc0: releasing imx-dsp-rproc
root@imx8mpevk:~#
root@imx8mpevk:~# modprobe imx_dsp_rproc no_mailboxes=1 /* insert kernel module
 with the right parameter */
[ 819.930792] remoteproc remoteproc0: imx-dsp-rproc is available
root@imx8mpevk:~#
root@imx8mpevk:~# ls -la /sys/class/remoteproc/ /* now, we have remoteproc0, for
 DSP */
total 0
drwxr-xr-x  2 root root 0 Mar  3 09:49 .
drwxr-xr-x 90 root root 0 Mar  3 09:49 ..
lrwxrwxrwx  1 root root 0 Mar  3 10:20 remoteproc0 -> ../../devices/
platform/3b6e8000.dsp/remoteproc/remoteproc0
root@imx8mpevk:~#
```

5.1.3.4 Load firmware on DSP and run it

To load the firmware on DSP and run it, execute the following commands:

```
root@imx8mpevk:~# echo -n /lib/firmware/imx/zephyr/imx8m-hello-world-zephyr.elf
 > /sys/class/remoteproc/remoteproc0/firmware
root@imx8mpevk:~# echo start > /sys/class/remoteproc/remoteproc0/state
[  107.320099] remoteproc remoteproc0: powering up imx-dsp-rproc
[  107.326031] remoteproc remoteproc0: Direct firmware load for /lib/firmware/
imx/zephyr/imx8m-hello-world-zephyr.elf failed with error -2
[  107.336696] remoteproc remoteproc0: Falling back to sysfs fallback for: /lib/
firmware/imx/zephyr/imx8m-hello-world-zephyr.elf
[  107.348365] remoteproc remoteproc0: Booting fw image /lib/firmware/imx/
zephyr/imx8m-hello-world-zephyr.elf, size 715896
[  107.360096] remoteproc remoteproc0: no resource table found for this firmware
[  107.367735] remoteproc remoteproc0: remote processor imx-dsp-rproc is now up
root@imx8mpevk:~#
```

5.1.3.5 Stop firmware

To stop the firmware, use the following command:

```
root@imx8mpevk:~# echo stop > /sys/class/remoteproc/remoteproc0/state
[  206.148281] remoteproc remoteproc0: stopped remote processor imx-dsp-rproc
root@imx8mpevk:~#
```

5.1.4 Get hello_world application output

To get the hello_world application output, follow these steps:

1. Get console through UART.
2. Open a serial terminal on the fourth serial port:

   ```
   user@developerpc:~# minicom -D /dev/ttyUSB3
   ```

   You see the following message in the terminal (also shown in Figure 2):

   ```
   Hello World! nxp_adsp_imx8m
   *** Booting Zephyr OS build zephyr-v3.5.0-1510-gaa71ed4a1f55 ***
   ```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

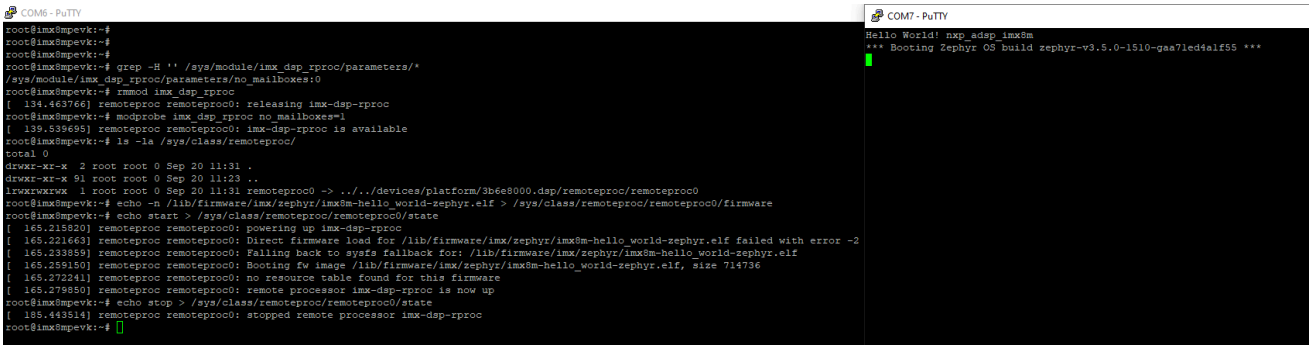

Figure 2. hello_world application output

You can use the above steps to build and test any other sample, such as synchronization or philosophers.

You can also run more complex samples that demonstrate how Linux and Zephyr can work in unison. The next section exemplifies the openamp_rsc_table application.

5.2 openamp_rsc_table application

The Zephyr's openamp_rsc_table application demonstrates how to use Open Asymmetric Multi-Processing (OpenAMP) with Zephyr based on a resource table. It is designed to respond to the:

- Linux rpmsg client sample
- Linux rpmsg tty driver

This sample implementation is compatible with platforms that embed a Linux kernel OS on the main processor and a Zephyr application on the coprocessor.

5.2.1 Load openamp_rsc_table application on DSP

As mentioned in Section 5.1.1, to load the openamp_rsc_table application on the DSP, use the imx_dsp_rproc driver, after enabling `CONFIG_IMX_DSP_REMOTEPROC` in the Linux kernel.

5.2.2 Compile openamp_rsc_table application in Zephyr

Compile the openamp_rsc_table application in Zephyr for the i.MX 8M Plus DSP, it means, compile the application for the nxp_adsp_imx8m board.

Go to the `zephyr/` folder from `zephyrproject` and run:

```
~/zephyrproject/zephyr$ west build -p always -b nxp_adsp_imx8m samples/subsys/
ipc/openamp_rsc_table
~/zephyrproject/zephyr$
~/zephyrproject/zephyr$ ls -la build/zephyr total 5284
drwxr-xr-x 14 nxa06898 nxp   4096 Sep 27 17:42 .
drwxr-xr-x  7 nxa06898 nxp   4096 Sep 27 17:42 ..
drwxr-xr-x  5 nxa06898 nxp   4096 Sep 27 17:42 arch
drwxr-xr-x  3 nxa06898 nxp   4096 Sep 27 17:42 boards
drwxr-xr-x  5 nxa06898 nxp   4096 Sep 27 17:42 cmake
-rw-r--r--  1 nxa06898 nxp     96 Sep 27 17:42 .cmake.dotconfig.checksum
drwxr-xr-x  6 nxa06898 nxp   4096 Sep 27 17:42 CMakeFiles
-rw-r--r--  1 nxa06898 nxp  13684 Sep 27 17:42 cmake_install.cmake
-rw-r--r--  1 nxa06898 nxp  41787 Sep 27 17:42 .config
drwxr-xr-x 16 nxa06898 nxp   4096 Sep 27 17:42 drivers
...
drwxr-xr-x  4 nxa06898 nxp   4096 Sep 27 17:42 soc
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
drwxr-xr-x 23 nxa06898 nxp   4096 Sep 27 17:42 subsys
-rw-r--r--  1 nxa06898 nxp   2421 Sep 27 17:42 zephyr.dts
-rw-r--r--  1 nxa06898 nxp    730 Sep 27 17:42 zephyr.dts.d
-rw-r--r--  1 nxa06898 nxp 124812 Sep 27 17:42 zephyr.dts.pre
-rw-r--r--  1 nxa06898 nxp 550701 Sep 27 17:42 zephyr_final.map
-rwxr-xr-x  1 nxa06898 nxp 998304 Sep 27 17:42 zephyr_openamp_rsc_table.elf
-rw-r--r--  1 nxa06898 nxp 550701 Sep 27 17:42 zephyr_openamp_rsc_table.map
-rw-r--r--  1 nxa06898 nxp   7463 Sep 27 17:42 zephyr_openamp_rsc_table.stat
-rwxr-xr-x  1 nxa06898 nxp 998476 Sep 27 17:42 zephyr_pre0.elf
-rw-r--r--  1 nxa06898 nxp 551293 Sep 27 17:42 zephyr_pre0.map
...
```

The `zephyr_openamp_rsc_table.elf` file is used as the firmware to be loaded on the DSP.

5.2.3 Run openamp_rsc_table application on DSP in Linux

The subsections that follow explain how to run the openamp_rsc_table application on HiFi 4 DSP from i.MX 8M Plus in Linux.

The `rpmsg_client_sample.ko` and `rpmsg_tty.ko` modules are used to communicate with the openamp_rsc_table application that runs on the DSP. These are sample modules that run on the main processor (Cortex A core).

To build the `rpmsg_client_sample.ko` and `rpmsg_tty.ko` modules, enable the `CONFIG_SAMPLE_RPMSG_CLIENT` and `CONFIG_RPMSG_TTY` configurations, respectively, in the Linux kernel.

5.2.3.1 Start i.MX 8M Plus EVK board

Start the i.MX 8M Plus EVK board with a specific DTS.

Use `imx8mp-evk-dsp.dtb`, and after inserting the `imx_dsp_rproc.ko` kernel module, you get:

```
root@imx8mpevk:~# insmod imx_dsp_rproc.ko
[  115.172960] remoteproc remoteproc0: imx-dsp-rproc is available
root@imx8mpevk:~# ls -la /sys/class/remoteproc/
total 0
drwxr-xr-x  2 root root 0 Mar  3 09:49 .
drwxr-xr-x 90 root root 0 Mar  3 09:49 ..
lrwxrwxrwx  1 root root 0 Mar  3 20:09 remoteproc0 -> ../../devices/
platform/3b6e8000.dsp/remoteproc/remoteproc0
```

Here, `remoteproc0`, which is for DSP, is used.

5.2.3.2 Check firmware on board

Check the firmware image on the board:

```
root@imx8mpevk:~# ls -la /lib/firmware/imx/zephyr/
total 148
drwxr-xr-x  2 root root   4096 Mar  9  2018 .
drwxr-xr-x 11 root root   4096 Mar  9  2018 ..
-rwxr-xr-x  1 root root  41524 Mar  9  2018 imx8-hello-world-zephyr.elf
-rwxr-xr-x  1 root root  57100 Mar  9  2018 imx8m-hello-world-zephyr.elf
-rwxr-xr-x  1 root root  41524 Mar  9  2018 imx8x-hello-world-zephyr.elf
-rwxr-xr-x  1 root root 998304 Mar  9  2018 imx8m-openamp_rsc_table-zephyr.elf
```

The firmware must be present in `/lib/firmware` before the remoteproc driver is probed; however, it can also be given with an absolute path.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

5.2.3.3 Insert `imx_dsp_rproc.ko` kernel module

Insert the `imx_dsp_rproc.ko` kernel module as shown below:

```
root@imx8mpevk:~# modprobe imx_dsp_rproc
[  115.172960] remoteproc remoteproc0: imx-dsp-rproc is available
root@imx8mpevk:~# ls -la /sys/class/remoteproc/
total 0
drwxr-xr-x  2 root root 0 Mar  3 09:49 .
drwxr-xr-x 90 root root 0 Mar  3 09:49 ..
lrwxrwxrwx  1 root root 0 Mar  3 20:09 remoteproc0 -> ../../devices/
platform/3b6e8000.dsp/remoteproc/remoteproc0
```

5.2.3.4 Insert rpmsg Linux client samples

Insert rpmsg Linux client samples as follows:

```
root@imx8mpevk:~# modprobe rpmsg_client_sample.ko     /* rpmsg client sample
 driver used to communicate with remote processor over the rpmsg bus */
root@imx8mpevk:~# modprobe rpmsg_tty.ko               /* export rpmsg endpoints
 as tty devices, usually found as /dev/ttyRPMSGx */
```

5.2.3.5 Load firmware on DSP and run it

5.2.3.5.1 rpmsg client sample

From Linux, send 100 messages with "hello world" to Zephyr.

Linux console:

```
root@imx8mpevk:~# echo -n zephyr_openamp_rsc_table.elf > /sys/class/remoteproc/
remoteproc0/firmware
root@imx8mpevk:~# echo start > /sys/class/remoteproc/remoteproc0/state
[  200.630824] remoteproc remoteproc0: powering up imx-dsp-rproc
[  200.637393] remoteproc remoteproc0: Booting fw image
 zephyr_openamp_rsc_table.elf, size 999412
[  200.649895] rproc-virtio rproc-virtio.2.auto: assigned reserved memory node
 vdev0buffer@94300000
[  200.662289] virtio_rpmsg_bus virtio0: rpmsg host is online
[  200.667889] rproc-virtio rproc-virtio.2.auto: registered virtio0 (type 7)
[  200.674715] remoteproc remoteproc0: remote processor imx-dsp-rproc is now up
[  200.681908] virtio_rpmsg_bus virtio0: creating channel rpmsg-client-sample
 addr 0x400
[  200.689959] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: new
 channel: 0x400 -> 0x400!
[  200.700409] virtio_rpmsg_bus virtio0: creating channel rpmsg-tty addr 0x401
[  200.707894] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 1 (src: 0x400)
[  200.717703] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 2 (src: 0x400)
[  200.726580] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 3 (src: 0x400)
[  200.735433] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 4 (src: 0x400)
[  200.744289] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 5 (src: 0x400)
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
[  200.753158] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 6 (src: 0x400)
[  200.761988] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 7 (src: 0x400)
[  200.770827] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 8 (src: 0x400)
[  200.779680] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 9 (src: 0x400)
[  200.788511] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 10 (src: 0x400)
...
[  201.529273] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 93 (src: 0x400)
[  201.538195] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 94 (src: 0x400)
[  201.547120] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 95 (src: 0x400)
[  201.556048] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 96 (src: 0x400)
[  201.564975] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 97 (src: 0x400)
[  201.573901] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 98 (src: 0x400)
[  201.582816] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 99 (src: 0x400)
[  201.591742] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: incoming
 msg 100 (src: 0x400)
[  201.600716] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: goodbye!
[  201.607877] virtio_rpmsg_bus virtio0: destroying channel rpmsg-client-sample
 addr 0x400
[  201.615953] rpmsg_client_sample virtio0.rpmsg-client-sample.-1.1024: rpmsg
 sample client driver is removed
```

Zephyr console:

```
*** Booting Zephyr OS build zephyr-v3.4.0-4490-gd885048637d6 ***
Starting application threads!

OpenAMP[remote]  linux responder demo started

OpenAMP[remote] Linux sample client responder started

[00:00:00.015,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0

[00:00:00.020,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.024,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.053,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0

[00:00:00.053,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.070,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
[00:00:00.070,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.079,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0

[00:00:00.079,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.088,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0

[00:00:00.088,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.097,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0

...
[00:00:00.935,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0

[00:00:00.935,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.944,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0

[00:00:00.944,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received

[00:00:00.944,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received
```

5.2.3.5.2 rpmsg TTY demo

On the Linux console, send a message to Zephyr, which replies with the "TTY <add>" prefix. <addr> corresponds to the Zephyr rpmsg-tty endpoint address.

Linux console:

```
root@imx8mpevk:~# cat /dev/ttyRPMSG0 &
[1] 1540
root@imx8mpevk:~# echo "Hello Zephyr" >/dev/ttyRPMSG0
TTY 0x0401: Hello Zephyr
root@imx8mpevk:~#
```

Zephyr console:

```
*** Booting Zephyr OS build zephyr-v3.4.0-4490-gd885048637d6 ***
Starting application threads!

OpenAMP[remote]  linux responder demo started

OpenAMP[remote] Linux tty responder started

[00:06:02.049,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
[00:06:02.049,000] <dbg> openamp_rsc_table: mailbox_notify: mailbox_notify: msg
 received
```

5.2.3.6 Stop firmware

To stop the firmware, use the following command:

```
root@imx8mpevk:~# echo stop > /sys/class/remoteproc/remoteproc0/state
[  495.366531] remoteproc remoteproc0: stopped remote processor imx-dsp-rproc
```

5.2.4 Get openamp_rsc_table application output

To get the openamp_rsc_table application output, follow these steps:

1. Get console through UART.
2. Open a serial terminal on the fourth serial port:

```
user@developerpc:~# minicom -D /dev/ttyUSB3
```

You see the following messages in the terminal (also shown in Figure 3):

```
*** Booting Zephyr OS build zephyr-v3.4.0-4490-gd885048637d6 ***
Starting application threads!

OpenAMP[remote]  linux responder demo started

OpenAMP[remote] Linux sample client responder started

OpenAMP[remote] Linux tty responder started
[00:00:00.015,000] <dbg> openamp_rsc_table: platform_ipm_callback:
 platform_ipm_callback: msg received from mb 0
```

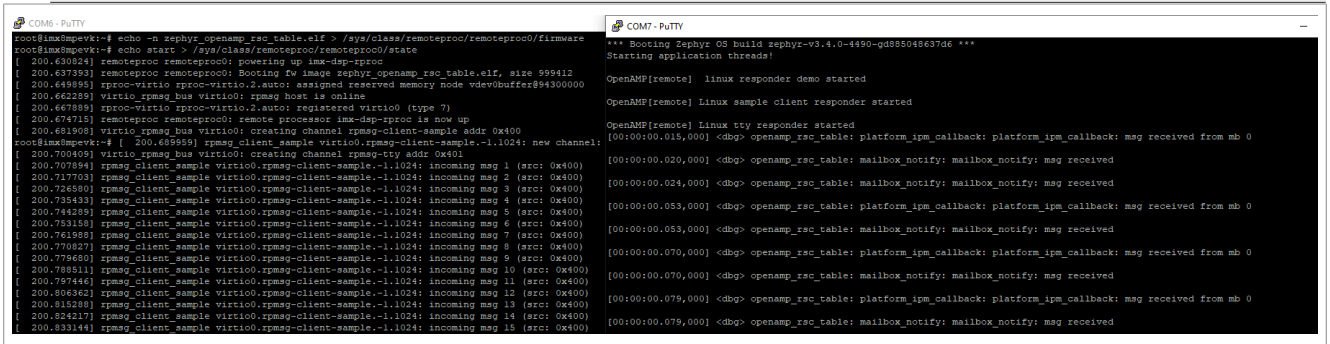

Figure 3. openamp_rsc_table application output

6 Acronyms

Table 3 lists the acronyms used in this document.

Table 3. Acronyms

Acronym	Description
DSP	Digital signal processor
DTS	Device tree source
IPC	Inter-process communication

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Table 3. Acronyms...*continued*

Acronym	Description
MCU	Microcontroller unit
MPU	Microprocessor unit
OpenAMP	Open Asymmetric Multi-Processing
OS	Operating system
rproc	Remote processor
rsc_table	Resource table
RTOS	Real-time operating system
UART	Universal Asynchronous Receiver/Transmitter

7 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2023 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

8 Revision history

Table 4 summarizes the revisions to this document.

Table 4. Revision history

Document ID	Release date	Description
AN13970 v.2	28 November 2023	Updated these sections: • Section 5.1.2 • Section 5.1.3.1 • Section 5.1.3.2 • Section 5.1.3.3 • Section 5.1.3.4 • Section 5.1.3.5

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Table 4. Revision history...*continued*

Document ID	Release date	Description
		• Section 5.1.4
		Added a new section: Section 5.2
AN13970 v.1	1 June 2023	Initial public release

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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 B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

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 and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) 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.

Cadence — the Cadence logo, and the other Cadence marks found at www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

i.MX — is a trademark of NXP B.V.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2023 NXP B.V. **All rights reserved.**

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

Date of release: 28 November 2023
Document identifier: AN13970