# MT793X IoT SDK for FreeRTOS ADSP IPI User Guide

| | |
|---|---|
| Version: | 1.1 |
| Release date: | 2021-07-09 |

## Version History

| Version | Date | Description |
|---|---|---|
| 1.0 | 2020-11-20 | Official release |
| 1.1 | 2021-07-09 | Change file location |

# Table of Contents

# 1 Getting Started

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

## 1.1 Overview

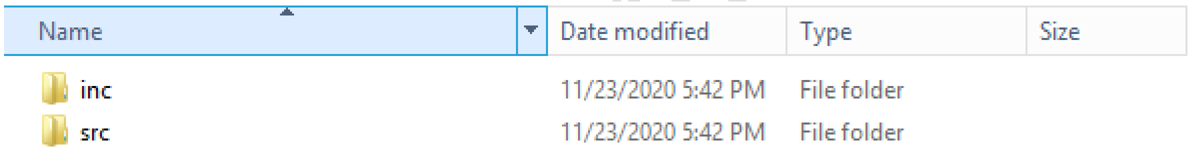
1. ADSP IPI is designed for the communication between AP and ADSP
2. ADSP IPI provides the same APIs for AP and ADSP to support two-way communication.
3. The data transferred by ADSP IPI is unformatted.

## 1.2 Code Layout

The current ADSP IPI driver is divided into two parts, AP and BSP. This section introduces the location of the AP and BSP code.

1. **AP: 7933 freertos**

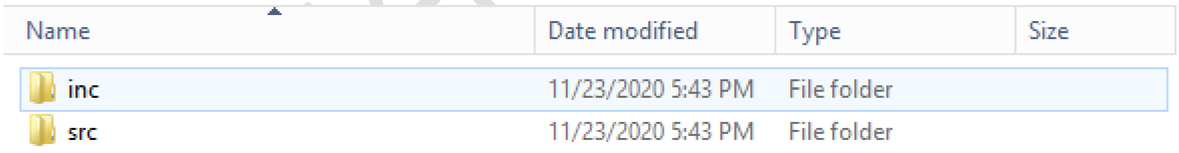
   \middleware\MTK\hifi4dsp\hifi4dsp_ipi\



| Name | Date modified | Type | Size |
|---|---|---|---|
| inc | 11/23/2020 5:42 PM | File folder | |
| src | 11/23/2020 5:42 PM | File folder | |

2. **BSP: freertos**

   \tinysys\adsp\HIFI4\project\mt7933\platform\drivers\HIFI4_A\ipi



| Name | Date modified | Type | Size |
|---|---|---|---|
| inc | 11/23/2020 5:43 PM | File folder | |
| src | 11/23/2020 5:43 PM | File folder | |

## 1.3 ADSP IPI APIs

ADSP IPI provides the same APIs for AP and ADSP to support two-way communication.

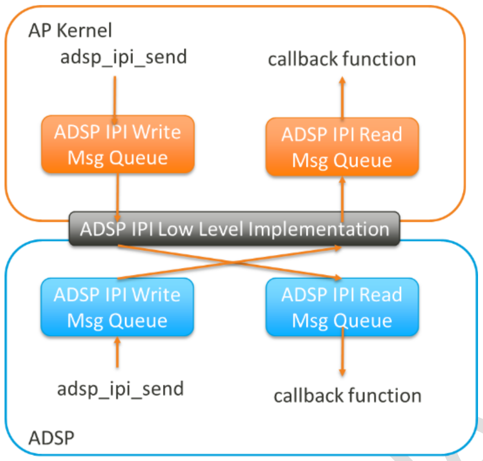
### adsp_ipi_send

- adsp_ipi_status adsp_ipi_send(uint32_t id, void *buf, uint32_t len, uint32_t wait, enum ipi_dir dir);
- Transfer specific command ID and buffer to the opposite end

### adsp_ipi_registration

- adsp_ipi_status adsp_ipi_registration(uint32_t id, ipi_handler_t handler, const char* name);

- Register callback function for specific command ID. The callback function is called after the corresponding command ID is received.



## Notice

- The definition of command ID must be consistent between AP and ADSP.
- Callback function is called in message queue one-by-one; please keep callback function as simple and short as possible.

# 2 ADSP IPI Usage Sample

**Add a new IPI command, IPI_NEW_CMD, which is sent by AP and handled by ADSP**

- Add IPI_NEW_CMD in "enum ipi_id" in adsp_ipi.h[AP side & DSP side] (IPI command must be defined both in AP side and ADSP side and remain consistent in both sides)
- Define a ipi_handler_t function "new_cmd_handler" in ADSP side, "typedef int (*ipi_handler_t)(int id, void* data, unsigned int len)"
- Register IPI command handler in the ADSP module init function, adsp_ipi_registration(IPI_NEW_CMD, new_cmd_handler, "new cmd handler");
- When AP wants to send IPI message to ADSP, call adsp_ipi_send(IPI_NEW_CMD, buffer, len, 1, 0) in AP side.
- The ADSP IPI module transfers the message from AP to ADSP, and calls back an IPI_NEW_CMD handler registered in ADSP side, new_cmd_handler

  -- buffer length is limited to 272 bytes

  -- when new_cmd_handler is called back, the content of parameter "data/len" is the same as the content in AP side

**Add a new IPI command, IPI_NEW_CMD, which is sent by ADSP and handled by AP**

- Add IPI_NEW_CMD in "enum ipi_id" in adsp_ipi.h[AP side & DSP side] (IPI command must be defined both in AP side and ADSP side and remain consistent in both sides)
- Define a ipi_handler_t function "new_cmd_handler" in AP side, "typedef int (*ipi_handler_t)(int id, void* data, unsigned int len)"
- Register IPI command handler in the AP module init function, adsp_ipi_registration(IPI_NEW_CMD, new_cmd_handler, "new cmd handler");
- When ADSP wants to send IPI message to AP, call adsp_ipi_send(IPI_NEW_CMD, buffer, len, 1, 0) in ADSP side.
- The ADSP IPI module transfers the message from ADSP to AP, and calls back an IPI_NEW_CMD handler registered in AP side, new_cmd_handler

  -- buffer length is limited to 272 bytes

-- when new_cmd_handler is called back, the content of parameter "data/len" is the same as the content in ADSP side

## 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.