

TCC892X UART USER GUIDE

TCC89xx_UART_USER_GUIDE

Rev. 0.10

Mar . 2. 2012

TeleChips

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.

This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.

Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Copyright in the material provided by Telechips, Inc. is owned by Telechips unless otherwise noted.

For reproduction or use of Telechips' copyright material, permission should be sought from Telechips. That permission, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute or commercialize any contents contained in the material in any manner without the written permission of Telechips. Trade marks used in Telechips' copyright material are the property of Telechips.

Important Notice

This product may include technology owned by Microsoft Corporation and in this case it cannot be used or distributed without a license from Microsoft Licensing, GP.

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:

"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial. Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit <http://mp3licensing.com>".

For customers who use other firmware of mp3:

"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit <http://mp3licensing.com>".

For customers who use Digital Wave DRA solution:

"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

For customers who use DTS technology:

"Supply of this implementation of DTS technology does not convey a license, exhaust DTS' rights in the implementation, or imply a right under any patent, or any other industrial or intellectual property right of DTS to use, offer for sale, sell, or import such implementation in any finished end-user or ready-to-use final product. Notice is hereby provided that a license from DTS is required prior to such use."

"This product made under license to U.S. Patents 5,451,942; 5,956,674; 5,974,380; 5,978,762; 6,487,535; 6,226,616 and/or foreign counterparts."

"© 1996 – 2010 DTS, Inc."

Revision History

| Date | Version | Description |
|----------|---------|--|
| 2012-3-2 | 0.10 | This document is a guide to theUART. Initial release |
| | | |
| | | |
| | | |
| | | |

TABLE OF CONTENTS

Contents

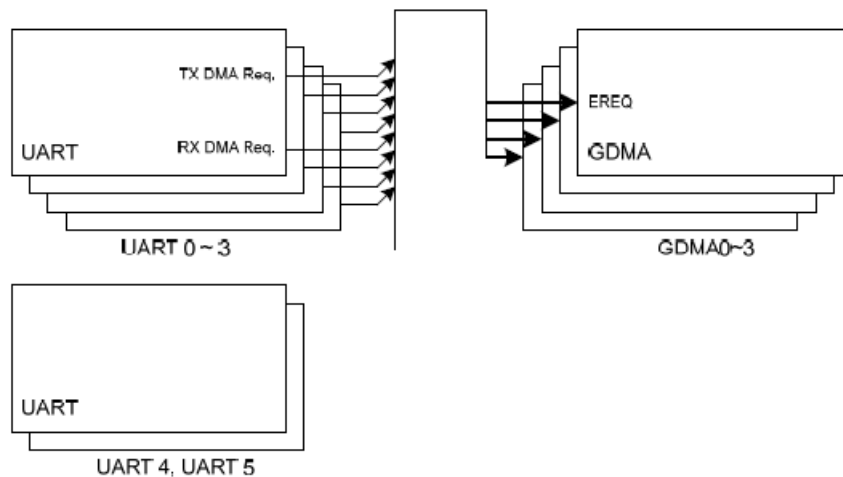
| | |
|--|-----|
| 1 Introduction | 1-1 |
| 2 UART | 2-1 |
| 2.1 TCC892X UART | 2-1 |
| 2.2 UART channel mux..... | 2-1 |
| 2.2.1 How to change port..... | 2-2 |
| 2.3 Default UART on TCC892X | 2-3 |
| 3 Setting the configuration of DMA | 3-4 |
| 3.1 How to set DMA of UART | 3-4 |
| 3.2 Setting platform data for DMA..... | 3-4 |

1 Introduction

This document is to describe method which make user to use UART for TCC892x.

2 UART

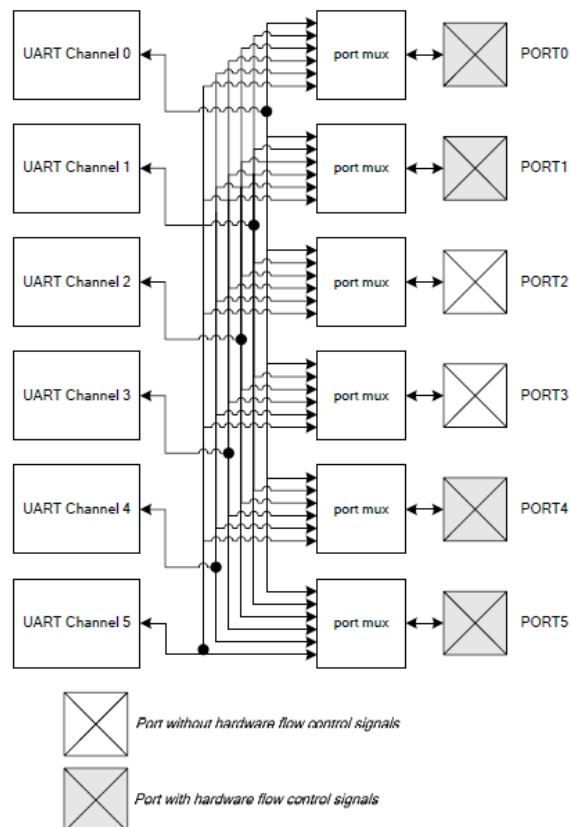
2.1 TCC892X UART



In case of tcc892x platform, it has six uart channel and 0,1,2,3 channels can be operated with DMA. Uart 4,5,6,7 can't use DMA.

2.2 UART channel mux

Each uart channel has port mux. So, it is possible to change from itself to other port of uart



2.2.1 How to change port

You can change port of uart. For example,
if you want to use uart channel 2 by port 4, you can change the port of the uart2 from port 2 to port 4.
You can use port 4 by uart 2.

```
#elif defined (TARGET_TCC8920_EVM)
    PUARTPORTCFG pUARTPORTCFG = (PUARTPORTCFG)HwUART_PORTCFG_BASE;

    //Bruce, should be initialized to not used port.
    pUARTPORTCFG->PCFG0.nREG = 0x00000000;
    pUARTPORTCFG->PCFG1.nREG = 0x00000000;

    #if (HW_REV == 0x1005 || HW_REV == 0x1007)
        // UART0 : Debug UART : GPIO_F[25,26] - UT[16]
        BITCSET(pUARTPORTCFG->PCFG0.nREG, 0xFF, 16);
        gpio_config(TCC_GPF(25), GPIO_FN9);
        gpio_config(TCC_GPF(26), GPIO_FN9);

        // UART1 : BT UART : GPIO_G[12-15] - UT[20]
        BITCSET(pUARTPORTCFG->PCFG0.nREG, 0xFF<<8, 20<<8);

        // UART3 : GPS UART :GPIO_G[4,5] - UT[18]
        BITCSET(pUARTPORTCFG->PCFG0.nREG, 0xFF<<24, 18<<24);
```

If you open uart.c file in lk bootloader(bootable/bootloader/lk/platform/tcc_shared/), you can find code(BITCSET(pUARTPORTCFG->PCFG0.nREG, 0xFF, 16)). This is channel selection. If you want to change port, you should fix this code.

20.3.2 Port Configuration Register

Port Configuration Register 0(PCFG0)

UART_PORT_CFG + 0x00

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|
| UART3 | | | | | | | | UART2 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UART1 | | | | | | | | UART0 | | | | | | | |

| Field | Name | RW | Reset | Description |
|-------|-------|-----|-------|-------------------------------|
| 31-24 | UART3 | R/W | 0x3 | UART #3 port mapping register |
| 23-16 | UART2 | R/W | 0x2 | UART #2 port mapping register |
| 15-8 | UART1 | R/W | 0x1 | UART #1 port mapping register |
| 7-0 | UART0 | R/W | 0x0 | UART #0 port mapping register |

Port Configuration Register 0(PCFG0)

UART_PORT_CFG + 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|
| UART7 | | | | | | | | UART6 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UART5 | | | | | | | | UART4 | | | | | | | |

| Field | Name | RW | Reset | Description |
|-------|-------|-----|-------|-------------------------------|
| 31-24 | UART7 | R/W | 0x3 | UART #7 port mapping register |
| 23-16 | UART6 | R/W | 0x2 | UART #6 port mapping register |
| 15-8 | UART5 | R/W | 0x1 | UART #5 port mapping register |
| 7-0 | UART4 | R/W | 0x0 | UART #4 port mapping register |

2.3 Default UART on TCC892X

On TCC88XX, there are default setting about uart.

- uart0 -> console
- uart1 -> Bluetooth
- uart3 -> GPS

If you want to change these, you can. But we recommend these setting.

3 Setting the configuration of DMA

3.1 How to set DMA of UART

Uart 0,1,2,3 can use DMA. But uart 0 is default console and uart 1 is for Bluetooth. Uart1 for Bluetooth uses DMA basically. If you want to use DMA of uart 2, 3, you should check option of DMA.

You can find option in kernel menuconfig.

- (in kernel folder) make menuconfig --> device drivers --> Character devices --> Serial drivers →

```
<*> Telechips SoC serial support
[*]   Support for DMA mode
[ ]   UART2 - Support for DMA mode
[ ]   UART3 - Support for DMA mode
[*]   Console on TCC serial port
[ ]   Telechips TCC88xx Smartcard driver support
< > MAX3100 support
< > Support for timberdale UART
< > Altera JTAG UART support
< > Altera UART support
```

UART2 – Support for DMA mode , UART3 – Support for DMA mode are DMA option of UART 2,3

3.2 Setting platform data for DMA

```
#if CONFIG_TCC_UART2_DMA
static struct tcc_uart_platform_data uart2_data = {
    .tx_dma_use      = 0,
    .tx_dma_buf_size= SERIAL_TX_DMA_BUF_SIZE,
    .tx_dma_base     = HwGDMA2_BASE,
    .tx_dma_ch       = SERIAL_TX_DMA_CH_NUM,
    .tx_dma_intr     = INT_DMA2_CH0,
    .tx_dma_mode     = SERIAL_TX_DMA_MODE,

    .rx_dma_use      = 1,
    .rx_dma_buf_size= SERIAL_RX_DMA_BUF_SIZE,
    .rx_dma_base     = HwGDMA2_BASE,
    .rx_dma_ch       = SERIAL_RX_DMA_CH_NUM-2,
    .rx_dma_intr     = 0,
    .rx_dma_mode     = SERIAL_RX_DMA_MODE,
};
#endif

#if CONFIG_TCC_UART3_DMA
static struct tcc_uart_platform_data uart3_data = {
    .tx_dma_use      = 0,
    .tx_dma_buf_size= SERIAL_TX_DMA_BUF_SIZE,
    .tx_dma_base     = HwGDMA2_BASE,
    .tx_dma_ch       = SERIAL_TX_DMA_CH_NUM+1,
    .tx_dma_intr     = INT_DMA2_CH1,
    .tx_dma_mode     = SERIAL_TX_DMA_MODE,

    .rx_dma_use      = 1,
    .rx_dma_buf_size= SERIAL_RX_DMA_BUF_SIZE,
    .rx_dma_base     = HwGDMA2_BASE,
    .rx_dma_ch       = SERIAL_RX_DMA_CH_NUM-1,
    .rx_dma_intr     = 0,
    .rx_dma_mode     = SERIAL_RX_DMA_MODE,
};
#endif
```

If you open board-(platform ex,tcc8920).c file(in arch/arm/mach-tcc89xx/), you can find above codes. These are platform data for DMA about uart2,3. If you can't these codes in that file, that platform doesn't be set yet. In case this, you can add these codes in that file(ex, board-tcc9300.c in arch/arm/tcc93xx)

You should also add following codes.

```
static void __init tcc8800_init_machine(void)
{
    __cpu_early_init();

    tcc8800_init_gpio();

#ifdef CONFIG_SPI_TCCXXX_MASTER || defined(CONFIG_SPI_TCCXXX_MASTER_MODULE)
    spi_register_board_info(tcc8800_spi0_board_info, ARRAY_SIZE(tcc8800_spi0_board_info));
    //spi_register_board_info(tcc8800_spi1_board_info, ARRAY_SIZE(tcc8800_spi1_board_info)); //jhlim
#endif

#ifdef CONFIG_TCC_OUTPUT_STARTER
    i2c_register_board_info(5, i2c_devices5, ARRAY_SIZE(i2c_devices5));
#endif

#ifdef CONFIG_SENSORS_AK8975
    /* Input mode */
    tcc_gpio_config(TCC_GPF(26), GPIO_FN(0) | GPIO_PULL_DISABLE); // GPIOF[26]: input mode, disable pull-up/down
    tcc_gpio_direction_input(TCC_GPF(26));
    tcc_gpio_config_intr(EXINT_EI1, EXTINT_GPIOF_26);
#endif

    i2c_register_board_info(0, i2c_devices1, ARRAY_SIZE(i2c_devices1));
#ifdef CONFIG_TOUCHSCREEN_TCC_AK4183
    i2c_register_board_info(1, i2c_devices2, ARRAY_SIZE(i2c_devices2));
#endif

#ifdef CONFIG_TCC_BT_DEV
    /* BT: use UART1 and TX DMA */
    platform_device_add_data(&tcc8800_uart1_device, &uart1_data_bt, sizeof(struct tcc_uart_platform_data));
#endif

#ifdef CONFIG_TCC_UART2_DMA
    platform_device_add_data(&tcc8800_uart2_device, &uart2_data, sizeof(struct tcc_uart_platform_data));
#endif

#ifdef CONFIG_TCC_UART3_DMA
    platform_device_add_data(&tcc8800_uart3_device, &uart3_data, sizeof(struct tcc_uart_platform_data));
#endif
```

Codes in red square are needed for setting DMA. These code also are in that file. And if you can't these codes, you should add these.