HCRTOS SDK uart userguide

1. 文档履历

版本号	日期	制/修订人	制/修订记录
1.0	2023.03.25	邱浩佳	新增文档说明

2. 概述

2.1 编写目的

介绍uart的功能和指导开发

2.2 读者对象

软件开发工程师和技术支持工程师

3. 模块介绍

注意一代1512系列硬件uart只支持uart0和uart1；uart2和uart3只支持gpio的tx；二代并无此限制；

hcrtos的串口支持五种使用情况：

1. 串口使用硬件tx和rx；
2. 串口使用硬件rx和gpio引脚模拟tx；
3. 仅有gpio引脚模拟tx；
4. 串口只使用硬件tx；
5. 串口只使用硬件rx；

注意：串口有涉及使用gpio引脚模拟tx时，波特率位9600；其它为115200或者其它常用波特率，可通过ioctl进行配置；

3.1 设备树配置

使用串口作为标准输出的配置如下，将其修改为所使用的引脚和uart@x，并添加到对应的设备树文件中。

```
        uart@2 {
                // case 1, hw rx/tx，硬件rx和tx
                pinmux-active = <PINPAD_B20 3 PINPAD_B19 3>;

/*
                // case 2, hw rx, gpio tx，硬件rx和gpio tx
                pinmux-active = <PINPAD_B20 3 PINPAD_B19 0>;
                tx-gpios = <PINPAD_B19>;

                // case 3, gpio tx only，串口只使用gpio tx
                pinmux-active = <PINPAD_B19 0>;
                tx-gpios = <PINPAD_B19>;

                // case 4, hw tx only，串口只使用硬件tx
                pinmux-active = <PINPAD_B19 3>;

                // case 5, hw rx only，串口使用硬件rx
                pinmux-active = <PINPAD_B20 3>;
*/
                device_type = "hichip,hcrtos-setup-setbit";
                reg_bit = <0xb8800094 18 1>;
                devpath = "/dev/uart2";
                status = "okay";
        };

        /* 系统阶段的标准输出 */
        stdio {
```

```
                serial0 = "/hcrtos/uart@2";
        };

        /* bootloader阶段的标准输出 */
        boot-stdio {
                serial0 = "/hcrtos/uart@2";
        };
```

其中：pinmux-active = <PINPAD_B20 3 PINPAD_B19 3>；PINPAD_B20和PINPAND_B19为所使用的串口引脚，**3是将其配置为rx和tx功能的值，其它引脚的值不一定是3。** 具体要根据原理图所使用的引脚进行替换和进行配置。不同引脚复用为串口所配置的值不一定相同，io引脚的复用功能值选择可以查看components/kernel/source/include/uapi/hcuapi/pinmux/hc16xx_pinmux.h。**

4. 模块接口说明

uart驱动除了通过设备树经行配置，还可以通过ioctl命令进行配置和操作。

```
ioctl(fd, SCIIOC_SET_HIGH_SPEED, 0);


ioctl(fd, SCIIOC_SET_NORMAL_SPEED, 0);


ioctl(fd, SCIIOC_SET_BAUD_RATE_115200, 0);


ioctl(fd, SCIIOC_SET_BAUD_RATE_57600, 0);


ioctl(fd, SCIIOC_SET_BAUD_RATE_19200, 0);


ioctl(fd, SCIIOC_SET_BAUD_RATE_9600, 0);


struct sci_setting sci_setting_para = { 0 };    //sci_setting具体参数请参考Sample Code
ioctl(fd, SCIIOC_SET_SETTING, &sci_setting_para);
```

5. 模块测试用例与Sample Code

5.1 uart配置Sample Code

```c
#include <stdint.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <getopt.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/ioctl.h>
#include <kernel/delay.h>
#include <kernel/lib/console.h>

#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <freertos/semphr.h>
#include <freertos/queue.h>
#include <hcuapi/sci.h>

static void print_help(void) {
    printf("\tfor example : uart_test -n1 -m1 -b1 -p2                \n");
    printf("\t      'n' 1 means open /dev/uart1              \n");
    printf("\t      'm' 0 means mode SCIIOC_SET_HIGH_SPEED:3.375MHz \n");
    printf("\t          1 means mode SCIIOC_SET_NORMAL_SPEED:115200 \n");
    printf("\t          2 means mode SCIIOC_SET_BAUD_RATE_115200    \n");
    printf("\t          3 means mode SCIIOC_SET_BAUD_RATE_57600     \n");
    printf("\t          4 means mode SCIIOC_SET_BAUD_RATE_19200     \n");
    printf("\t          5 means mode SCIIOC_SET_BAUD_RATE_9600      \n");
    printf("\t          6 means mode SCIIOC_SET_SETTING             \n");
    printf("\t      'b' 1 means bits_mode(range:0~5):           \
        \n\t                1 stop bit and 8 data bits      \
        \n\t                1.5 stop bits and 5 data bits       \
        \n\t                2 stop bits and 6 data bits         \
        \n\t                2 stop bits and 7 data bits         \
        \n\t                2 stop bits and 8 data bits         \n");
    printf("\t      'p' 1 means parity_mode(range:0~2):         \
        \n\t                PARITY_EVEN             \
        \n\t                PARITY_ODD              \
        \n\t                PARITY_NONE             \n");
}

int uart_test(int argc, char *argv[])
{
    int fd;
    char dev_path[32];
    char ch;
    opterr = 0;
    optind = 0;
    uint32_t uart   = 0;
    uint32_t mode   = 0;
    uint32_t bits   = 0;
    uint32_t parity = 0;
    struct sci_setting sci_setting_para = { 0 };

    if (argc < 2) {
```

```
        print_help();
        return -1;
    }

    while ((ch = getopt(argc, argv, "hn:m:b:p:")) != EOF) {
        switch (ch) {
            case 'h':
                print_help();
                return 0;
            case 'n':
                uart = atoi(optarg);
                break;
            case 'm':
                mode = atoi(optarg);
                break;
            case 'b':
                bits = atoi(optarg);
                break;
            case 'p':
                parity = atoi(optarg);
                break;
            default:
                printf("Invalid parameter %c\r\n", ch);
                print_help();
                return -1;
        }
    }

    sprintf(dev_path, "/dev/uart%d", (int)uart);
    fd = open(dev_path, O_RDWR);
    if (fd < 0) {
        printf("can't open %s\n",dev_path);
        return -EINVAL;
    }

    if (bits > bits_mode4 || parity > PARITY_NONE) {
        printf("bits mode / parity mode: Invalid parameter\n");
        return -1;
    }

    sci_setting_para.bits_mode = bits;
    sci_setting_para.parity_mode = parity;

    switch (mode) {
    case 0:
        ioctl(fd, SCIIOC_SET_HIGH_SPEED, 0);
        break;
    case 1:
        ioctl(fd, SCIIOC_SET_NORMAL_SPEED, 0);
        break;
    case 2:
        ioctl(fd, SCIIOC_SET_BAUD_RATE_115200, 0);
        break;
    case 3:
        ioctl(fd, SCIIOC_SET_BAUD_RATE_57600, 0);
        break;
    case 4:
        ioctl(fd, SCIIOC_SET_BAUD_RATE_19200, 0);
```

```
        break;
    case 5:
        ioctl(fd, SCIIOC_SET_BAUD_RATE_9600, 0);
        break;
    case 6:
        ioctl(fd, SCIIOC_SET_SETTING, &sci_setting_para);
        break;
    }

    close(fd);

    return 0;
}

CONSOLE_CMD(uart_test, NULL, uart_test, CONSOLE_CMD_MODE_SELF, "uart test function app")

```

5.2 uart读写Sample Code

用标准C函数open、read、write函数既可以读写串口，对应的路径为/dev/uart1，这里以串口1为示例。

```
uint8_t read_byte[10] = {0};
int fd = 0;

fd = open("dev/uart1",O_RDWR);
if (fd < 0) {
   printf("uart1 open error.....\n");
   return -1;
}

write(fd, "1234567890", 10);

read(fd, &read_byte, 10);
```

6. 模块调试方法

在SDK根目录下：输入make menuconfig，根据路径打开测试命令

```
There is no help available for this option.
Symbol: CONFIG_CMD_UART [=y]
Type  : bool
Prompt: uart test operations
  Location:
    -> Components
      -> Cmds (BR2_PACKAGE_CMDS [=y])
  Defined at source:85
  Depends on: BR2_PACKAGE_CMDS [=y]
```

选中后执行：make cmds-rebuild all，烧录完毕后，在串口输入以下内容使用调试命令，SCIIOC_SET_SETTING参数需要和bits_mode和parity_mode一起使用，配置完需要重新配置串口工具。

```
hc1600a@dbc3000v10# uart_test -h
        for example : uart_test -n1 -m1 -b1 -p2
                        'n'     1 means open /dev/uart1
                        'm'     0 means mode SCIIOC_SET_HIGH_SPEED:3.375MHz
                                1 means mode SCIIOC_SET_NORMAL_SPEED:115200
                                2 means mode SCIIOC_SET_BAUD_RATE_115200
                                3 means mode SCIIOC_SET_BAUD_RATE_57600
                                4 means mode SCIIOC_SET_BAUD_RATE_19200
                                5 means mode SCIIOC_SET_BAUD_RATE_9600
                                6 means mode SCIIOC_SET_SETTING
                        'b'     1 means bits_mode(range:0~5):
                                        1 stop bit and 8 data bits
                                        1.5 stop bits and 5 data bits
                                        2 stop bits and 6 data bits
                                        2 stop bits and 7 data bits
                                        2 stop bits and 8 data bits
                        'p'     1 means parity_mode(range:0~2):
                                        PARITY_EVEN
                                        PARITY_ODD
                                        PARITY_NONE
```

7. 常见问题

Q：使用gpio tx时，发现串口打印助手没有输出或者输出乱码。

A：涉及gpio tx模式时，串口波特率只有9600，请修改串口助手波特率。