

{Sigmastar IPCM UART 使用参考}

{ Version 1 }

摘要

本文主要介绍 IPCM UART 以及其使用，包括简单的原理以及多路串口的配置说明，方便配合客户端使用多路串口。

关键词:

UART ttyS0 ttyS1 ttyS2

REVISION HISTORY

Revision No.	Description	Updated By	Date
{Version 1}	• {Initial release}	{英文名_中文名}	{MM/DD/YYYY}

目录

1	HW 原理图介绍	错误!未定义书签。
1.1	公版电路图说明	错误!未定义书签。
1.2	EVB board uart pad 介绍	错误!未定义书签。
2	IPCM UART SW 配置方法	错误!未定义书签。
3	测试 UART	错误!未定义书签。

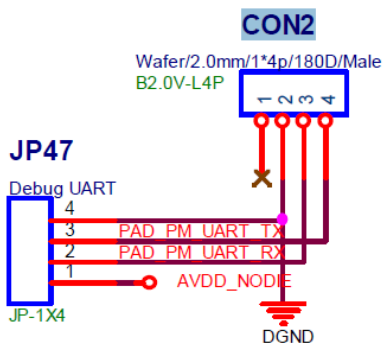
1 HW 原理图介绍

1.1 公版电路图说明

关于 UART 的电路图做简单的说明, BGA package EVB board, 对应 chip 来说我们是有一路 UART port : CON2 for debug UART

如下图:

Debug UART



1.2 EVB board uart pad 介绍

在内部有一路 debug uart, 公版电路图上显示 pad, 对应 EVB 的 board 有一路 pad, 对应 CON2。**Debug UART 仅供系统 logging 与 debug 使用**. 每一个具体的 board 上对应的接口都可以配置具体的 uart(可以通过缓存器将所有的 uart 都通过一个 pad 来测试, 可以通过敲缓存器实现测试)。

2 IPCM UART SW 配置方法

IPCM 的 uart 与具体的 pad 配置流程如下：

1、首先在 linux-4.9\arch\arm\boot\dts\infinity6e.dtsi

```
aliases {
    console = &uart0;
    serial0 = &uart0;
};

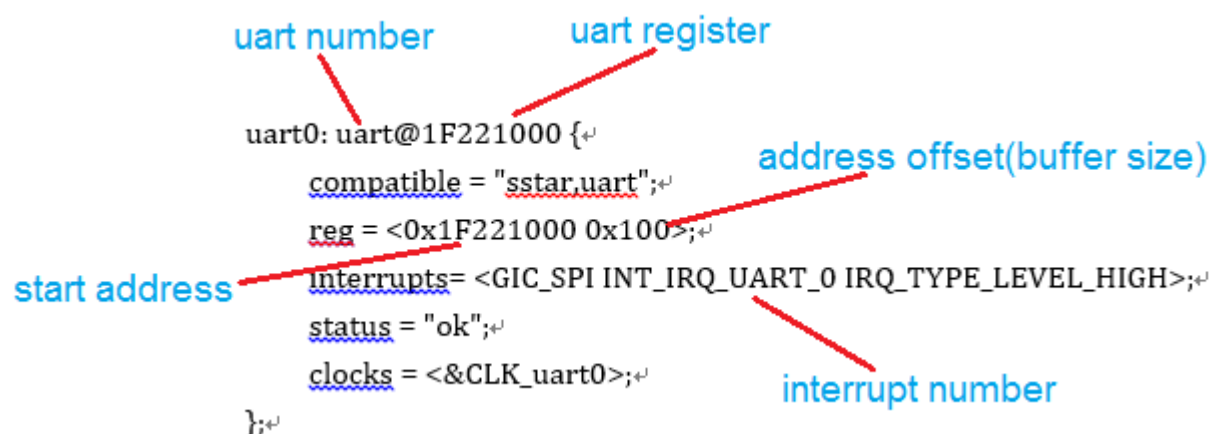
.....

uart0: uart@1F221000 {
    compatible = "sstar,uart";
    reg = <0x1F221000 0x100>;
    interrupts= <GIC_SPI INT_IRQ_UART_0 IRQ_TYPE_LEVEL_HIGH>;
    status = "ok";
    clocks = <&CLK_uart0>;
};
```

该文件是 linux 系统的设备树配置文件，该档中的 uart0 对应在系统中通过命令查看到的 ttyS0 如下：

```
/sys/class/tty # ls -l
total 0
lrwxrwxrwx 1 root root 0 Jan 1 00:00 console -> ../../devices/virtual/tty/console
lrwxrwxrwx 1 root root 0 Jan 1 00:00 ptmx -> ../../devices/virtual/tty/ptmx
lrwxrwxrwx 1 root root 0 Jan 1 00:00 tty -> ../../devices/virtual/tty/tty
lrwxrwxrwx 1 root root 0 Jan 1 00:00 ttyS0 -> ../../devices/soc0/soc/1f221000.uart/tty/ttyS0
```

对应配置文件说明如下：



```
uart0: uart@1F221000 {
    compatible = "sstar,uart";
    reg = <0x1F221000 0x100>;
    interrupts= <GIC_SPI INT_IRQ_UART_0 IRQ_TYPE_LEVEL_HIGH>;
    status = "ok";
    clocks = <&CLK_uart0>;
};
```

Diagram annotations:

- uart number**: points to `uart0`
- uart register**: points to `uart@1F221000`
- address offset(buffer size)**: points to `0x100` in the `reg` property
- start address**: points to `0x1F221000` in the `reg` property
- interrupt number**: points to `INT_IRQ_UART_0` in the `interrupts` property

在板子中查看 uart 设备相关信息可直接 cat ms_uart 文件，如下：

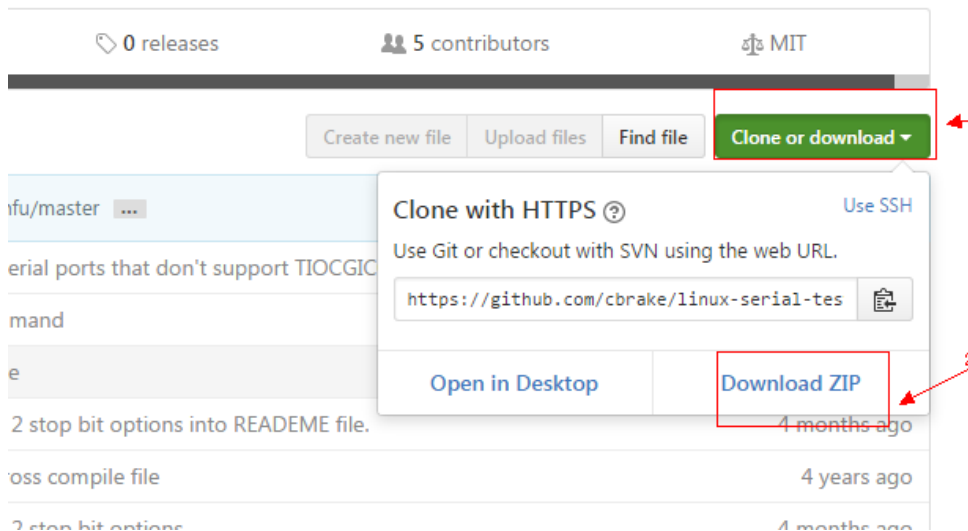
```
// # cd proc/tty/driver/  
/proc/tty/driver # ls  
ms_uart  
/proc/tty/driver # cat ms_uart  
serinfo:1.0 driver revision:  
0: uart:unknown mmio:0x00000000 irq:40 tx:14379 rx:339 RTS|CTS|DTR|DSR|CD  
/proc/tty/driver #
```

也可以通过命令行查看如下：

```
// # dmesg | grep tty  
Kernel command line: console=ttyS0,115200 root=/dev/mtdblock2 rootfstype=squashfs ro init=/linuxrc cma=64M  
console [ttyS0] enabled  
1f221000.uart: ttyS0 at MMIO 0x0 (irq = 40, base_baud = 10800000) is a unknown
```

3 测试 UART

- 1、在网上 dump 一个开源的测试程序，地址如下:<https://github.com/cbrake/linux-serial-test> 可以直接点击进行按照如下方式下载即可：



下载后的文件如下：



linux-serial-test
.c

- 2、编译下载的.c 文件



serial_test.tar.g
z

用上述档，里面包含 makefile 以及 serial_test.c 档，然后操作如下步骤：

```
tar xzvf serial_test.tar.gz
cd serial_test
make
```

然后去 bin 目录下取可执行文件



serial_test

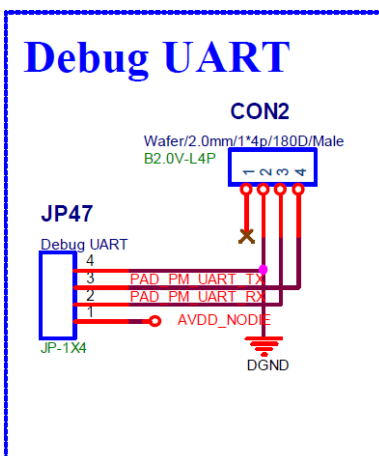
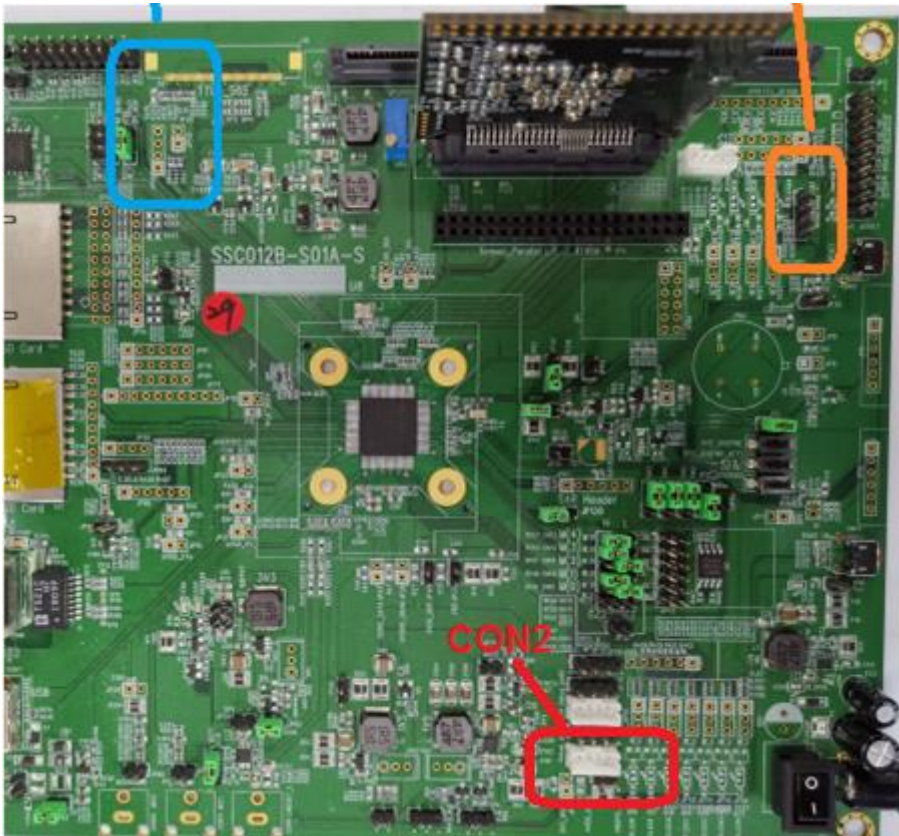
具体操作如下：

```

tar: Exiting with failure status due to previous errors
gavin.ran@szsmc220:~/linux$ tar xzvf serial_test.tar.gz 1
serial_test/
serial_test/Makefile
serial_test/src/
serial_test/src/serial-test.c
gavin.ran@szsmc220:~/linux$ cd serial_test 2
gavin.ran@szsmc220:~/linux/serial_test$ make 3
CC src/serial-test.c
BIN /home/gavin.ran/linux/serial_test/obj/serial_test
cp -f /home/gavin.ran/linux/serial_test/obj/serial_test /home/gavin.ran/linux/serial_test/bin 4
gavin.ran@szsmc220:~/linux/serial_test$

```

2、EVB board 上对应的 JP-Connection 都可以配置具体的 uart, 具体如下：



对应 pc 端的连接

```
Linux serial test/app
Error opening serial port
[/system/vendor/serial_test]## ./serial_test -s -e -p /dev/ttyS1 -b 115200
Linux serial tes[II]CHIP_FUNCTION SET. ID=2, param=1
tapping UART
[II]CHIP_FUNCTION SET. ID=1, param=1
[II]CHIP_FUNCTION SET. ID=2, param=1
[II]CHIP_FUNCTION SET. ID=1, param=1
[II]CHIP_FUNCTION SET. ID=2, param=1
[II]CHIP_FUNCTION SET. ID=1, param=1
/dev/ttyS1: count for this session: rx=0, tx=69391, rx err=0
/dev/ttyS1: TIOCGICOUNT: ret=0, rx=0, tx=65296, frame = 0, overrun = 0, par
/dev/ttyS1: count for this session: rx=0, tx=138511, rx err=0
/dev/ttyS1: TIOCGICOUNT: ret=0, rx=0, tx=134416, frame = 0, overrun = 0, par
/dev/ttyS1: count for this session: rx=0, tx=207631, rx err=0
/dev/ttyS1: TIOCGICOUNT: ret=0, rx=0, tx=203536, frame = 0, overrun = 0, par
/dev/ttyS1: count for this session: rx=0, tx=276751, rx err=0
/dev/ttyS1: TIOCGICOUNT: ret=0, rx=0, tx=272656, frame = 0, overrun = 0, par
/dev/ttyS1: count for this session: rx=0, tx=349711, rx err=0
/dev/ttyS1: TIOCGICOUNT: ret=0, rx=0, tx=345616, frame = 0, overrun = 0, par
/dev/ttyS1: count for this session: rx=0, tx=418831, rx err=0
```

4、将服务器 mount 的一个目录 mount 到板子上，如我讲如下目录 mount 到 board 上：

/home/gavin.ran/gavin_demo/serial_test

用在 board 上执行：mount -t nfs -o nolock 172.21.28.81:/home/gavin.ran/gavin_demo /system/vendor

然后进入：cd /system/vendor/serial_test 测试。