

项目一	Wi-Fi 串口第 1 部分.....	3
1.1	硬件原理.....	3
1.2	OpenWrt 支持串口 2.....	4
1.3	串口 2 的简单测试.....	6

www.f403tech.cc

项目一 Wi-Fi 串口第 1 部分

本项目目标

- 掌握 RT5350 串口 2 的使用方法
- 实现 Wi-Fi 串口

1.1 硬件原理

从 RT5350 的芯片手册上可以得知, RT5350 一共有两个串口, 分别为 UART Lite、UART Full, 详细定义如下:

UART Lite interface : 2 pins				
P3	RXD2	I, IPD	4mA	UART Lite RXD
N2	TXD2	O, IPD	4mA	UART Lite TXD
UART Full interface : 8 pins				
N3	RXD	I, IPD	4mA	UART RXD.
M3	RIN	I, IPD	4mA	UART RIN.
K4	CTS_N	I, IPD	4mA	UART CTS_N.
L3	DSR_N	I, IPD	4mA	UART DSR_N.
J4	DCD_N	I, IPD	4mA	UART DCD_N.
K2	TXD	O, IPD	4mA	UART TXD.
N4	DTR_N	O, IPD	4mA	UART DTR.
P2	RTS_N	O, IPD	4mA	UART RTS.

UART Lite 就是我们惯称为的串口 1, 作为系统调试串口, 通过这个串口, 我们可以看到 OpenWrt (Linux) 系统的启动信息, 并且可以通过该串口来控制开发板。

UART Full 就是我们惯称为的串口 2, 我们通常就是用它来做 Wi-Fi 串口, 然后用来控制其他串口设备。

UARTL	RXD2	GPIO#16
	TXD2	GPIO#15
UARTF	RIN	GPIO#14
	DSR_N	GPIO#13
	DCD_N	GPIO#12
	DTR_N	GPIO#11
	RXD	GPIO#10
	CTS_N	GPIO#9
	TXD	GPIO#8
	RTS_N	GPIO#7

通过上表, 我们发现串口 1 是和 GPIO15、16 复用的, 而串口 2 是和 GPIO7~14 复用的。如果我们再详细的看芯片手册, 我们就会发现, 串口 2 其实也是和 I2S 复用的, 具体如下。

UARTF pin share scheme

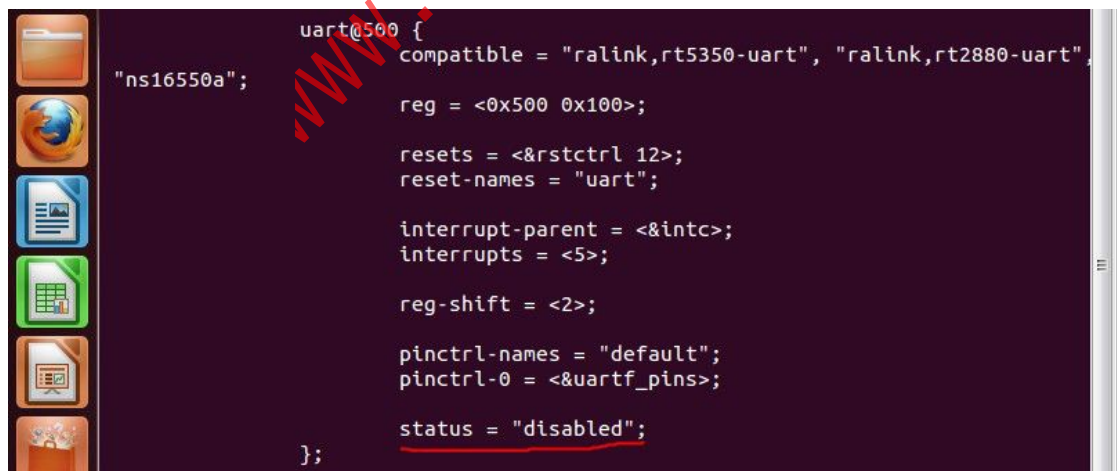
Pin Name	3'b000 UARTF	3'b001 PCM, UARTF	3'b010 PCM, I2S	3'b011 I2S UARTF	3'b100 PCM, GPIO	3'b101 GPIO, UARTF	3'b110 GPIO I2S	3'b111 GPIO
RIN	RIN	PCMDTX	PCMDTX	RXD	PCMDTX	GPIO#14	GPIO#14	GPIO#14
DSR_N	DSR_N	PCMDRX	PCMDRX	CTS_N	PCMDRX	GPIO#13	GPIO#13	GPIO#13
DCD_N	DCD_N	PCMCLK	PCMCLK	TXD	PCMCLK	GPIO#12	GPIO#12	GPIO#12
DTR_N	DTR_N	PCMFS	PCMFS	RTS_N	PCMFS	GPIO#11	GPIO#11	GPIO#11
RXD	RXD	RXD	I2SSDI	I2SSDI	GPIO#10	RXD	I2SSDI	GPIO#10
CTS_N	CTS_N	CTS_N	I2SSDO	I2SSDO	GPIO#9	CTS_N	I2SSDO	GPIO#9
TXD	TXD	TXD	I2SWS	I2SWS	GPIO#8	TXD	I2SWS	GPIO#8
RTS_N	RTS_N	RTS_N	I2SCLK	I2SCLK	GPIO#7	RTS_N	I2SCLK	GPIO#7

在答疑过程中，经常有人问 RT5350 的串口 2 是哪些管脚，那么通过这张表格，我们就能找出串口 2 对应的管脚了。对于串口，我们其实也只用关系发送(TXD)、接受(RXD)和地(GND)三个引脚。因此，对于串口 2，也就是 I2SSDI、I2SWS、GND 三个引脚了。

1.2 OpenWrt 支持串口 2

为了使用开发板的串口 2，必须首先把串口使能。由于我们是使用的 MPRA2 这款路由的配置，在它的配置上串口被配置成了 GPIO，用于控制 LEDs，所以这里需要把它改为串口模式。修改非常简单，只要修改 dts 配置文件就可以，这里就不啰嗦了，直接讲述如何修改串口的配置。

首先修改 openwrt/trunk/target/linux/ramips/dts 目录下的 rt5350.dtsi 配置文件。默认的 rt5350.dtsi 配置文件，是将那个扩展串口状态给禁用了，所以我们看不到扩展串口。这里需要把那行去掉或者将状态改为 okay，详细如下。



默认的 rt5350.dtsi 配置文件

```

uart@500 {
    compatible = "ralink,rt5350-uart", "ralink,rt2880-uart",
    "ns16550a";

    reg = <0x500 0x100>;

    resets = <&rstctrl 12>;
    reset-names = "uart";

    interrupt-parent = <&intc>;
    interrupts = <5>;

    reg-shift = <2>;

    pinctrl-names = "default";
    pinctrl-0 = <&uartf_pins>;

    status = "okay";
};

```

修改后的 rt5350.dtsi 配置文件

rt5350.dtsi 配置文件中除了修改扩展串口状态外，还需要修改设备节点。

因为我们把扩展串口状态使能后，它将在调试串口的后面，扩展串口变为 ttyS0，而调试串口变为 ttyS1，所以这里需要把调试串口的 ttyS0，变为 ttyS1，具体如下。

```

chosen {
    bootargs = "console=ttyS0,57600";
};

```

默认的 rt5350.dtsi 配置文件

```

chosen {
    bootargs = "console=ttyS1,57600";
};

```

修改后的 rt5350.dtsi 配置文件

这样，rt5350.dtsi 配置文件就改完了，但是扩展串口相应引脚被设置为了 GPIO 模式，怎么样才能变为串口模式了呢？需要修改 MPRA2.dts 这个配置文件，同样位于 openwrt/trunk/target/linux/ramips/dts 目录下。

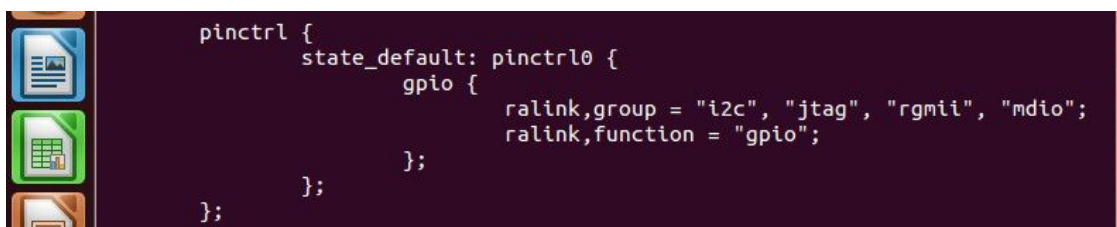
把 GPIO 模式去掉，把设置 GPIO 的设置项中 uartf 去掉，详细如下。

```

pinctrl {
    state_default: pinctrl0 {
        gpio {
            ralink,group = "i2c", "jtag", "rgmii", "mdio", "
            ralink,function = "gpio";
        };
    };
};
uartf;

```

默认的 MPRA2.dts 配置文件



```

pinctrl {
    state_default: pinctrl0 {
        gpio {
            ralink,group = "i2c", "jtag", "rgmii", "mdio";
            ralink,function = "gpio";
        };
    };
};

```

修改后的 MPRA2.dts 配置文件

1.3 串口 2 的简单测试

经过前面的努力，扩展串口 2 就可以使用了，扩展串口 2 的名字为 ttyS0，我们可以通过 `echo hello f403tech > /dev/ttyS0` 这个命令进行测试，如果对方接收到 `hello f403tech` 这个字符串，那说明串口 2 已经可以正常使用了。注意，此时串口 2 默认的波特率是 9600。

```

root@openwrt:~# ls /dev/tty*
/dev/tty /dev/ttyS0 /dev/ttyS1
root@openwrt:~#

```

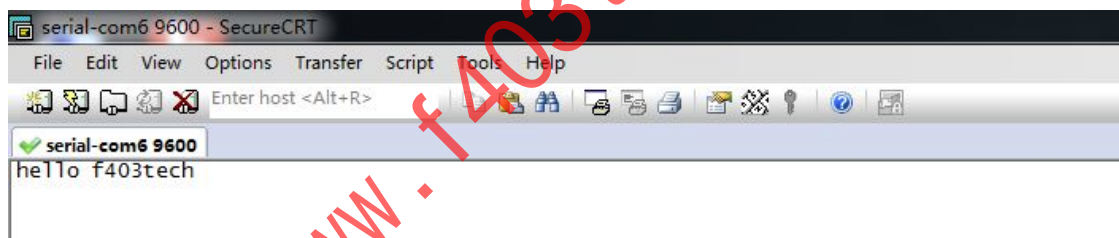
扩展串口 2 为 ttyS0

```

root@openwrt:~#
root@openwrt:~# echo hello f403tech > /dev/ttyS0
root@openwrt:~#

```

输入测试命令



```

serial-com6 9600 - SecureCRT
File Edit View Options Transfer Script Tools Help
Enter host <Alt+R>
serial-com6 9600
hello f403tech

```

测试结果

注意：

1. 该教程为我司 (www.f403tech.com) 原创教程，版权所有；
2. 该教程会不断更新、不断深入，详情请咨询我司客服；
3. 针对该教程，我们还有 QQ 群和论坛，专门负责技术答疑，详情请咨询我司客服。