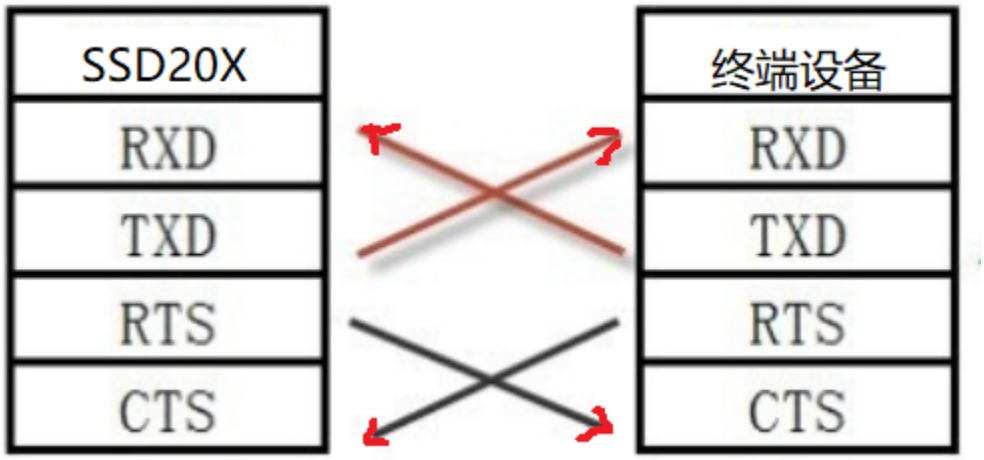
fuart & uart流控使用参考

Fuart流控流程

HW连接如下:



参数说明：

RTS （Require ToSend，发送请求）为输出信号，用于指示本设备准备好可接收数据，低电平有效，**低电平**说明**本设备可以接收数据**。
CTS （Clear ToSend，发送允许）为输入信号，用于判断是否可以向对方发送数据，低电平有效，**低电平**说明**本设备可以向对方发送数据**。

流控制方式：

如果**本端可以处理数据流**（可以接收对方数据），那么**本端将RTS 置为低电平**,对方设备看到自己的CTS脚是低电平，那么它（**对方）可以发送数据**。

如果**本端不可以处理数据流**（不想接收对方数据），那么**本端将RTS置为高电平**，对方设备看到自己的CTS脚是高电平，那么它**（对方）停止发送数据**。等待CTS的脚位变化。

测试步骤：FUART 测试方式

注意事项：

1：确认pin脚配置Fuart

kernel\arch\arm\boot\dts\infinity2m-ssc011a-s01a-padmux-display.dtsi :

```
<PAD_FUART_RX PINMUX_FOR_FUART_MODE_1 MDRV_PUSE_FUART_RX>,
<PAD_FUART_TX PINMUX_FOR_FUART_MODE_1 MDRV_PUSE_FUART_TX>,
<PAD_FUART_CTS PINMUX_FOR_FUART_MODE_1 MDRV_PUSE_FUART_CTS>,
<PAD_FUART_RTS PINMUX_FOR_FUART_MODE_1 MDRV_PUSE_FUART_RTS>,
```

kernel\arch\arm\boot\dts\infinity2m-ssc011a-s01a-padmux-display.dtsi :

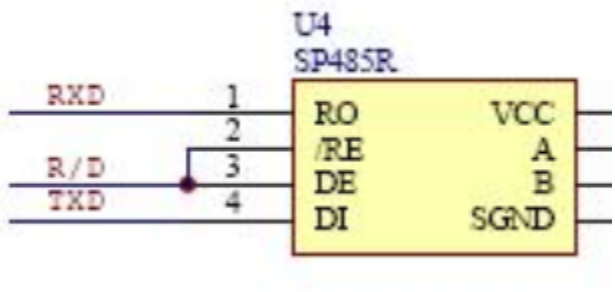
```
fuart: uart2@1F220400 {
compatible = "sstar,uart";
reg = <0x1F220400 0x100>, <0x1F220600 0x100>;
interrupts = <GIC_SPI INT_IRQ_FUART IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI INT_IRQ_URDMA IRQ_TYPE_LEVEL_HIGH>;
clocks = <&CLK_fuart>;
dma = <0>;
sctp_enable = <1>;//rts cts enable is 1
//pad = <PAD_FUART_RX>;//fuart mode2
pad = <PAD_FUART_CTS>;//fuart mode 1
tolerance = <2>;
status = "ok";
};
```

2：确认软件使能cts/rts，如果没有使能cts/rts功能 ，他们默认值为高电平状态。

3：如果要保证ssd20x处于可接受状态，使用命令：cat /dev/ttyS2 &　　，如果没有一直开启ttyS2的设备，那么Fuart的中断状态会自动清除。获取register[101e 2]==0x0000;正常为0x0005.

接RS485芯片，半双工通信流程

连接方法如图：



RS485属于master-salver模式，同时mater作为发起端，salver应答端。网络中同时只有一台在发送数据，其他设备属于监听状态。

主要流程如下：所有行为需要等待master端的通知。

host/master 给rs485芯片发送时，先把R/D拉高。将数据发送完毕后，将R/D 拉低，Host/Master进入监听状态，等待slaver应答。

控制slave收发的这根pin需要放到uart dirver去控制，不然会有延迟的问题。如题措施如下：
http://hcgit04-master:9080/#/c/mstar/kernel/linux-4.9/+/85932/

app层使用方法如下：

```
struct serial_rs485 rs485conf;

memset(&rs485conf,0,sizeof(rs485conf));

rs485conf.padding[0] = 17;      //用来控制slaver收发的gpio index
rs485conf.delay_rts_after_send = 2000;     //发送完最后一个字节需要的delay，单位：us
rs485conf.flags |= SER_RS485_RTS_ON_SEND;  //发送前拉高gpio，打开SER_RS485_RTS_AFTER_SEND指的是发送后拉高
rs485conf.flags |= SER_RS485_ENABLED; // 使能本串口485模式，默认禁用
ioctl(iHandle, TIOCSRS485, &rs485conf);
```