Struct uart\_ops {

Void (\*set\_mctl)(struct uart\_port \*,unsigned int mctrl);

};

Struct uart\_port {

Struct uart\_ops \*ops;

};

Uart的rts/cts，贺式smartmodem rs232，目前采用贺式smartmodem

Rts为输出信号，用于指示本串口可以接受数据，低电平表示可以接受数据

Cts为输入信号，用于判断是否可以向对方传输数据，低电平有效，低电平表示可以向对端传输数据。

Drivers/tty/serial/amba-pl011.c

Drivers/tty/serial/serial\_core.c

## 不能接受了，需要进行uart\_throttle

不能接受了，说明uart buf满了，在n\_tty层进行uart\_throttle,拉高RTS，对端收到RTS，会停止发送。

Void uart\_throttle(struct tty\_struct \*tty)

{

Struct uart\_port \*port;

If(I\_IXOFF(tty))

Mask |= UPSTAT\_AUTOXOFF;

If(C\_CRTSCTS(tty))

Mask |= UPSTAT\_AUTORTS;

If(mask & UPSTAT\_AUTORTS)

Uart\_clear\_mctrl(port,TIOCM\_RTS);

If(mask & UPSTAT\_AUTOXOFF)

Uart\_send\_xchar(tty,STOP\_CHAR(tty));

}

Uart\_clear\_mctrl(port,TIOCM\_RTS)

->uart\_update\_mctrl(port,0, TIOCM\_RTS)

{

Unsigned int old;

Old = port->mctrl;

Port->mctrl = (old &~clear) |set;

If(old != port->mctl)

Port->ops->set\_mctrl(port,port->mctrl);

}

Port->ops->set\_mctrl(port,port->mctrl) ->

Pl011\_set\_mctrl(port,port->mctrl)

{

Struct uart\_amba\_port \*uap = container\_of(port,struct uart\_amba\_port,port);

Cr = pl011\_read(uap,REG\_CR);

#define TIOCMBIT(tiocmbit,uartbit) \

If(mctrl & tiocmbit) \

Cr | =uartbit;

Else

Cr &= ~uartbit;

TIOCMBIT(TIOCM\_RTS,UART011\_CR\_RTS);

If(port->status &UPSTAT\_AUTORTS)

TIOCMBIT(TIOCM\_RTS,UART011\_CR\_RTSEN);

Pl011\_write(cr,uap,REG\_CR);

}

## 不能发送了

对端拉高RTS，收到RTS后，即表示不能发送。

Pl011\_int()->

Pl011\_modem\_status

Pl011\_modem\_status(struct uart\_amba\_port \*uap)

{

Uart\_handle\_cts\_change(&uap->port,status&uap->vendor->fr\_cts);

}