高速串口唤醒后不能使用：

tty\_open

--->uart\_open

--->uart\_startup(tty, state, 0);

--->uart\_port\_startup(tty, state, init\_hw);

---->msm\_hs\_startup

--->msm\_hs\_resource\_vote(msm\_uport);

--->msm\_hs\_config\_uart\_gpios(uport);

--->msm\_hs\_config\_uart\_gpios(uport);

--->msm\_hs\_spsconnect\_tx(msm\_uport);

--->disconnect\_rx\_endpoint(msm\_uport);

--->msm\_hs\_spsconnect\_rx(uport);

msm\_hs\_pm\_sys\_resume\_noirq

----->msm\_uport->pm\_state = MSM\_HS\_PM\_SUSPENDED;

msm\_hs\_pm\_sys\_suspend\_noirq

---->msm\_hs\_pm\_suspend(dev);

msm\_hs\_pm\_resume

------>toggle\_wakeup\_interrupt(msm\_uport);

------>msm\_hs\_clk\_bus\_vote(msm\_uport);

----->msm\_hs\_resource\_on(msm\_uport);

读msm\_uport->client\_req\_state

设置：

：msm\_uport->pm\_state = MSM\_HS\_PM\_ACTIVE;

msm\_hs\_pm\_suspend

----->msm\_hs\_resource\_off(msm\_uport);

----->msm\_hs\_clk\_bus\_unvote(msm\_uport);

----->toggle\_wakeup\_interrupt(msm\_uport);

msm\_uport->client\_req\_state读该状态

msm\_uport->pm\_state=MSM\_HS\_PM\_SUSPENDED;==设置

Sps中断:

-->msm\_hs\_sps\_tx\_callback

-->queue\_kthread\_work(&msm\_uport->tx.kworker, &msm\_uport->tx.kwork);

--->init\_kthread\_work(&tx->kwork, msm\_serial\_hs\_tx\_work);

--->msm\_serial\_hs\_tx\_work()

--->msm\_hs\_submit\_tx\_locked

--->msm\_hs\_queue\_rx\_desc

--->msm\_hs\_start\_tx\_locked

--->msm\_hs\_sps\_tx\_callback

发送

---->sps\_event->callback = msm\_hs\_sps\_rx\_callback;

---->msm\_hs\_sps\_rx\_callback

---->queue\_kthread\_work(&msm\_uport->rx.kworker,&msm\_uport->rx.kwork);

--->init\_kthread\_work(&rx->kwork, msm\_serial\_hs\_rx\_work);

--->msm\_serial\_hs\_rx\_work

---->msm\_hs\_start\_rx\_locked(uport);

----> msm\_hs\_queue\_rx\_desc(msm\_uport);

---->msm\_hs\_post\_rx\_desc(msm\_uport, rx->rx\_inx);

--->sps\_transfer\_one(rx->prod.pipe\_handle, rbuff\_addr,..)

--->sps\_transfer\_one

---->sps\_bam\_pipe\_transfer\_one

---->bam\_pipe\_set\_desc\_write\_offset

----->bam\_write\_reg\_field

msm\_hs\_pm\_suspend

msm\_hs\_resource\_off

-->msm\_hs\_disconnect\_rx(uport);

Uart open

---->sps\_connect

---->sps\_h2bam

---->sps\_register\_event

---->sps\_get\_config

---->sps\_set\_config

BAM device of this pipe is NULL.sps\_tx\_disconnect():

Cat 出错：

----->sps\_connect

---->sps\_h2bam

----->sps\_register\_event

----->sps\_transfer\_one

----->sps\_get\_config

----->sps\_disconnect

BAM device of this pipe is NULL.

cat ttyHS0 ok

---->sps\_connect

---->sps\_h2bam

----->sps\_connect

----->sps\_h2bam

----->sps\_register\_event

----->sps\_connect

----->sps\_h2bam

----->sps\_connect

----->sps\_h2bam

----->sps\_register\_event

----->sps\_transfer\_one

----->sps\_disconnect

----->sps\_connect

----->sps\_h2bam

----->sps\_connect

------>sps\_h2bam

------>sps\_register\_event

------>sps\_get\_config

------>sps\_set\_config

------>sps\_disconnect

Echo ok

--->sps\_connect

---->sps\_h2bam

---->sps\_register\_event

---->sps\_h2bam

----->sps\_transfer\_one

----->sps\_disconnect

----->sps\_connect

----->sps\_h2bam

----->sps\_register\_event

----->sps\_transfer\_one

----->sps\_get\_config

----->sps\_set\_config

----->sps\_disconnect

----->sps\_h2bam

接受

uart\_add\_one\_port(&msm\_hs\_driver, uport);

---->uart\_add\_one\_port(&msm\_hs\_driver, uport);

---->msm\_hs\_config\_port

uartdm\_init\_port

----->init\_kthread\_work(&rx->kwork,msm\_serial\_hs\_rx\_work);

----->init\_kthread\_work(&tx->kwork, msm\_serial\_hs\_tx\_work);

Uart注册

msm\_hs\_probe

----->msm\_hs\_sps\_init(msm\_uport);

---->sps\_register\_bam\_device(&bam, &bam\_handle);

---->msm\_hs\_sps\_init\_ep\_conn(msm\_uport,&msm\_uport->rx.prod,UART\_SPS\_PROD\_PERIPHERAL);

---->msm\_hs\_sps\_init\_ep\_conn(msm\_uport, &msm\_uport->tx.cons,UART\_SPS\_CONS\_PERIPHERAL);

SPS驱动注册

Data Mover Local(DML)

Write/read event

Descriptor fifo

pipe

Consumer

Producer

bam