

# AG35-QuecOpen

## UART 开发指导

**Automotive 模块系列**

版本: AG35-QuecOpen\_UART\_开发指导\_V1.0

日期: 2019-03-04

状态: 临时文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市徐汇区虹梅路 1801 号宏业大厦 7 楼 邮编：200233

电话：+86 21 51086236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：

<http://www.quectel.com/cn/support/sales.htm>

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://www.quectel.com/cn/support/technical.htm>

或发送邮件至：[support@quectel.com](mailto:support@quectel.com)

## 前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

## 版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2019，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2019.**

# 文档历史

## 修订记录

| 版本  | 日期         | 作者  | 变更表述                                      |
|-----|------------|-----|---|
| 1.0 | 2018-02-28 | 高飞虎 | 初始版本                                      |
| 1.1 | 2019-03-02 | 张文郑 | 添加 AG35 项目 UART 列表和 DTSI 文件配置<br>添加串口使用说明 |

## 目录

|   |           |
|---|-----------|
| 文档历史 .....                                  | 1         |
| 目录 .....                                    | 2         |
| 表格索引 .....                                  | 3         |
| 图片索引 .....                                  | 4         |
| <b>1 引言 .....</b>                           | <b>5</b>  |
| <b>2 AG35 -QuecOpen 串口说明 .....</b>          | <b>6</b>  |
| 2.1. 串口使用说明 .....                           | 6         |
| <b>3 硬件电路设计推荐 .....</b>                     | <b>7</b>  |
| 3.1. 带电平转换的串口设计电路 .....                     | 7         |
| 3.1.1. 推荐使用 TI 公司的 TXS0108EPWR 电平转换芯片 ..... | 7         |
| 3.1.2. 其他电平转换电路：三极管电平转换参考电路 .....           | 7         |
| 3.2. 无电平转换芯片参考电路 .....                      | 8         |
| <b>4 驱动层及设备树软件适配 .....</b>                  | <b>9</b>  |
| 4.1. UART 管脚使用 .....                        | 9         |
| 4.1.1. 重要说明 .....                           | 9         |
| 4.1.2. 串口 1 引脚定义 .....                      | 9         |
| 4.1.3. 调试串口引脚定义 .....                       | 9         |
| 4.1.4. 串口 2 (RTS/CTS 与 SPI 复用) 引脚定义 .....   | 10        |
| 4.1.5. 串口 3 (与 SPI 复用) 引脚定义 .....           | 10        |
| 4.1.6. 串口 4 (与 SDC1 复用) 引脚定义 .....          | 10        |
| 4.1.7. 串口 5 (与 SDC1 复用) 引脚定义 .....          | 11        |
| 4.1.8. 串口逻辑电平 .....                         | 11        |
| 4.2. UART 设备树配置方法 .....                     | 11        |
| 4.2.1. 配置说明 .....                           | 11        |
| 4.2.2. 串口 1 配置 .....                        | 11        |
| 4.2.3. 调试串口配置 .....                         | 13        |
| 4.2.4. 串口 2 配置 .....                        | 13        |
| 4.2.5. 串口 3 配置 .....                        | 14        |
| <b>5 QuecOpen 应用层 API .....</b>             | <b>16</b> |
| 5.1. 用户编程说明 .....                           | 16        |
| 5.2. UART API 介绍 .....                      | 16        |
| <b>6 UART 功能测试验证 .....</b>                  | <b>19</b> |
| 6.1. example 介绍及编译 .....                    | 19        |
| 6.2. 功能测试 .....                             | 20        |
| 6.2.1. 未使能流控 .....                          | 20        |
| 6.2.2. 使能硬件流程 .....                         | 21        |
| 6.2.3. 使能软件流控 .....                         | 23        |
| 6.2.3.1. 软件流控 XON/XOFF 字符说明 .....           | 23        |
| 6.2.3.2. 软件流控测试 .....                       | 23        |

## 表格索引

|   |    |
|---|----|
| 表 1: AG35 串口 1(软件 DTSI 配置使用 BLSP1_UART3 字段)引脚定义 ..... | 9  |
| 表 2: AG35 调试串口(软件 DTSI 配置使用 BLSP1_UART2 字段)引脚定义 ..... | 9  |
| 表 3: AG35 串口 2(软件 DTSI 配置使用 BLSP1_UART5 字段)引脚定义 ..... | 10 |
| 表 4: AG35 串口 3(软件 DTSI 配置使用 BLSP1_UART6 字段)引脚定义 ..... | 10 |
| 表 5: AG35 串口 4(软件 DTSI 配置使用 BLSP1_UART1 字段)引脚定义 ..... | 10 |
| 表 6: AG35 串口 5(软件 DTSI 配置使用 BLSP1_UART4 字段)引脚定义 ..... | 11 |
| 表 7: 串口逻辑电平 .....                                     | 11 |

Quectel  
Confidential

## 图片索引

|                                 |   |
|---------------------------------|---|
| 图 1: TXS0108EPWR 电平转换芯片电路 ..... | 7 |
| 图 2: 三极管电平转换参考电路 .....          | 8 |
| 图 3: 1.8V 直连电路 .....            | 8 |

Quectel  
Confidential

# 1 引言

文档从用户开发角度出发，介绍了硬件，软件驱动层，软件应用层等；可以帮助客户简易而快速的进行开发。

本文档主要适用于海外市场，目前支持该接口的 Automotive 模块包括：

- AG35

Quectel  
Confidential

## 2 AG35 -QuecOpen 串口说明

AG35-QuecOpen 模块提供 6 路串口：调试串口、串口 1、串口 2、串口 3、串口 4 和串口 5。串口 1、串口 2、串口 3 和串口 4 功能相同，硬件上均支持 RTS/CTS，可作为外设通信串口。其中，串口 1 与 SPI 复用，串口 2 与 SPI 复用，串口 4 和串口 5 与 SDC1 复用，更多的 pin 脚复用情况介绍，请参阅第 4 章节。如下描述了这四个串口的主要特性。

- 调试串口支持 115200bps 波特率，用于 Linux 控制和日志输出。
- 串口 1 支持 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps, 230400bps, 460800bps 和 921600bps 波特率，默认波特率为 115200bps；支持 RTS, CTS 流控。用做普通外设通信串口。
- 串口 2 支持 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps, 230400bps, 460800bps 和 921600bps 波特率，默认波特率为 115200bps；支持 RTS, CTS 流控。用做普通外设通信串口。
- 串口 3 支持 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps, 230400bps, 460800bps 和 921600bps 波特率，默认波特率为 115200bps；支持 RTS, CTS 流控。用做普通外设通信串口。
- 串口 4 支持 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps, 230400bps, 460800bps 和 921600bps 波特率，默认波特率为 115200bps；支持 RTS, CTS 流控。用做普通外设通信串口。
- 串口 5 支持 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps。不支持 RTS 和 CTS 流控。

### 2.1. 串口使用说明

串口分为：High-speed uart 和 Low-speed uart 之分，这个两种的模式主要区别是：High-speed uart 是 BAM pipe 方式传输；Low-speed uart 是 FIFO 方式传输。生成的设备节点不同：Low-speed uart 设备名字是/dev/ttyHSLx；High-speed uart 设备名字是/dev/ttyHSx。两个模式在硬件连接上，没有区别，唯一需要确保收发管脚信号完整性。

另外，对于 High-speed uart 和 Low-speed uart 选择，推荐如下：如果 uart 的波特率超过 115200bps 波特率或者涉及到的数据包，建议使用 High-speed uart，并且推荐使用硬件流控传输，例如：蓝牙等通信。如果 uart 的波特率小于或等于 115200bps 并且小的数据包，可以使 Low-speed uart，例如：调试串口等。



## 3 硬件电路设计推荐

注意：QuecOpen 4G 模块 CTS 与 RTS 已经做了对调，所以用户在连的时候，是将 MCU 的 CTS 接到 4G 模块的 CTS 上，RTS 接 RTS。

### 3.1. 带电平转换的串口设计电路

#### 3.1.1. 推荐使用 TI 公司的 TXS0108EPWR 电平转换芯片

AG35-QuecOpen 模块的串口电平为 1.8V。若客户主机系统电平为 3.3V，则需在模块和主机的串口连接中增加电平转换器，推荐使用 TI 公司的 TXS0108EPWR。下图为使用电平转换芯片的参考电路设计。

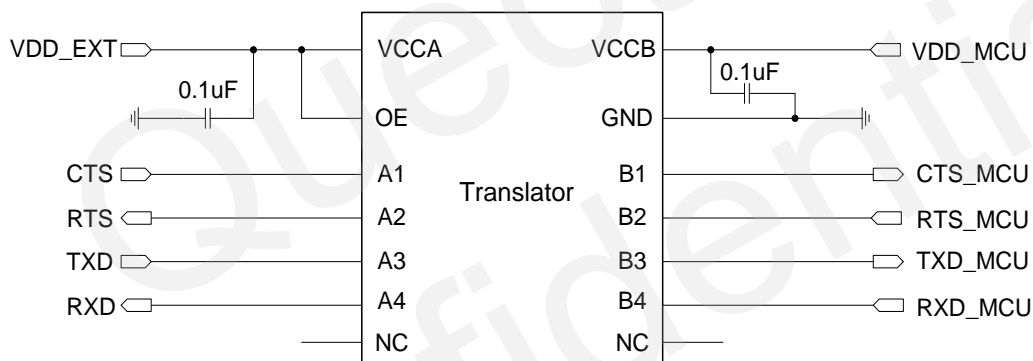


图 1: TXS0108EPWR 电平转换芯片电路

更多信息请访问 <http://www.ti.com>.

#### 3.1.2. 其他电平转换电路：三极管电平转换参考电路

另一种电平转换电路如下图所示。如下虚线部分的输入和输出电路设计可参考实线部分，但需注意连接方向。图中的三极管电平转换电路不适用于波特率超过 460Kbps 的应用。

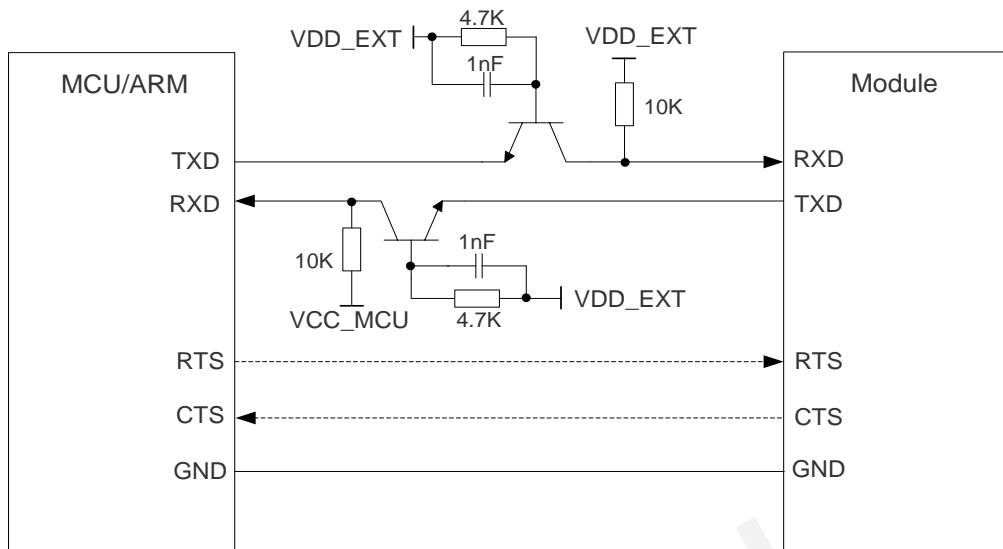


图 2：三极管电平转换参考电路

### 3.2. 无电平转换芯片参考电路

模块的串口电平为 1.8V，若客户主机系统电平也为 1.8v，直连即可：

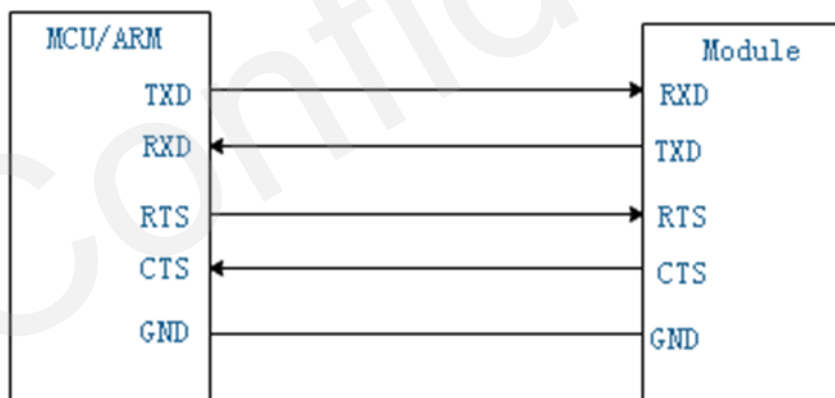


图 3：1.8v 直连电路

# 4 驱动层及设备树软件适配

## 4.1. UART 管脚使用

### 4.1.1. 重要说明

- 1.以下表格中，非默认的复用功能需要在软件配置后才有效，请参考对应的功能章节进行软件配置；
- 2.AG35-QuecOpen 中，主串口不再具有 AT 功能，仅作为普通外设通信串口；
- 3.具体管脚使用参考《Quectel\_AG35 \_QuecOpen\_GPIO\_Assignment\_Speadsheet》。

### 4.1.2. 串口 1 引脚定义

表 1：AG35 串口 1(软件 dtsi 配置使用 blsp1\_uart3 字段)引脚定义

| 管脚名称      | 管脚号 | I/O | 功能描述           |        |                |
|-----------|-----|-----|----------------|--------|----------------|
|           |     |     | 复用功能 1（默认）     | 复用功能 2 | 复用功能 3         |
| UART1_CTS | 56  | DO  | UART_CTS_BLSP3 | GPIO_3 | SPI_CLK_BLSP3  |
| UART1_RTS | 57  | DI  | UART_RTS_BLSP3 | GPIO_2 | SPI_CS_N_BLSP3 |
| UART1_TXD | 60  | DO  | UART_TXD_BLSP3 | GPIO_0 | SPI_MOSI_BLSP3 |
| UART1_RXD | 58  | DI  | UART_RXD_BLSP3 | GPIO_1 | SPI_MISO_BLSP3 |

### 4.1.3. 调试串口引脚定义

表 2：AG35 调试串口(软件 dtsi 配置使用 blsp1\_uart2 字段)引脚定义

| 引脚名     | 引脚号 | I/O | 描述     | 备注       |
|---------|-----|-----|--------|----------|
| DBG_TXD | 71  | DO  | 模块发送数据 | 1.8V 电源域 |
| DBG_RXD | 72  | DI  | 模块接收数据 | 1.8V 电源域 |

#### 4.1.4. 串口 2 (RTS/CTS 与 SPI 复用) 引脚定义

表 3: AG35 串口 2(软件 dtsi 配置使用 blsp1\_uart5 字段)引脚定义

| 引脚名       | 引脚号 | I/O | 功能描述           |         |                |
|-----------|-----|-----|----------------|---------|----------------|
|           |     |     | 复用功能 1 (默认)    | 复用功能 2  | 复用功能 3         |
| UART2_CTS | 164 | DO  | UART_CTS_BLSP5 | GPIO_11 | SPI_CLK_BLSP5  |
| UART2_RTS | 166 | DI  | UART_RTS_BLSP5 | GPIO_10 | SPI_CS_N_BLSP5 |
| UART2_TXD | 163 | DO  | UART_TXD_BLSP5 | GPIO_8  | SPI_MOSI_BLSP5 |
| UART2_RXD | 165 | DI  | UART_RXD_BLSP5 | GPIO_9  | SPI_MISO_BLSP5 |

#### 4.1.5. 串口 3 (与 SPI 复用) 引脚定义

表 4: AG35 串口 3(软件 dtsi 配置使用 blsp1\_uart6 字段)引脚定义

| 引脚名      | 引脚号 | I/O | 功能描述           |         |                |            |
|----------|-----|-----|----------------|---------|----------------|------------|
|          |     |     | 复用功能 1 (默认)    | 复用功能 2  | 复用功能 3         | 复用功能 4     |
| SPI_CS_N | 79  | DO  | SPI_CS_N_BLSP6 | GPIO_22 | UART_RTS_BLSP6 | PCM_1_DOUT |
| SPI_MOSI | 77  | DO  | SPI_MOSI_BLSP6 | GPIO_20 | UART_TXD_BLSP6 | PCM_1_SYNC |
| SPI_MISO | 78  | DI  | SPI_MISO_BLSP6 | GPIO_21 | UART_RXD_BLSP6 | PCM_1_DIN  |
| SPI_CLK  | 80  | DO  | SPI_CLK_BLSP6  | GPIO_23 | UART_CTS_BLSP6 | PCM_1_CLK  |

#### 4.1.6. 串口 4 (与 SDC1 复用) 引脚定义

表 5: AG35 串口 4(软件 dtsi 配置使用 blsp1\_uart1 字段)引脚定义

| 引脚名        | 引脚号 | I/O | 功能描述        |         |                |
|------------|-----|-----|-------------|---------|----------------|
|            |     |     | 复用功能 1 (默认) | 复用功能 2  | 复用功能 3         |
| SDC1_DATA0 | 20  | IO  | SDC1_DATA0  | GPIO_15 | UART_CTS_BLSP1 |
| SDC1_DATA1 | 21  | IO  | SDC1_DATA1  | GPIO_14 | UART_RTS_BLSP1 |
| SDC1_DATA2 | 22  | IO  | SDC1_DATA2  | GPIO_13 | UART_RXD_BLSP1 |

SDC1\_DATA3 23 IO SDC1\_DATA3 GPIO\_12 UART\_TXD\_BLSP1

#### 4.1.7. 串口 5（与 SDC1 复用）引脚定义

表 6: AG35 串口 5(软件 dtsi 配置使用 blsp1\_uart4 字段)引脚定义

| 引脚名      | 引脚号 | I/O | 功能描述       |         |                |
|----------|-----|-----|------------|---------|----------------|
|          |     |     | 复用功能 1（默认） | 复用功能 2  | 复用功能 3         |
| SDC1_CMD | 18  | IO  | SDC1_CMD   | GPIO_17 | UART_RXD_BLSP4 |
| SDC1_CLK | 19  | DO  | SDC1_CLK   | GPIO_16 | UART_TXD_BLSP4 |

#### 4.1.8. 串口逻辑电平

表 7: 串口逻辑电平

| 参数              | 最小值  | 最大值  | 单位 |
|-----------------|------|------|----|
| V <sub>IL</sub> | -0.3 | 0.6  | V  |
| V <sub>IH</sub> | 1.2  | 2.0  | V  |
| V <sub>OL</sub> | 0    | 0.45 | V  |
| V <sub>OH</sub> | 1.35 | 1.8  | V  |

## 4.2. UART 设备树配置方法

### 4.2.1. 配置说明

关于串口的配置，如所兼容的 driver，引脚的选择，寄存器地址，串口中断号，CLK，以及系统休眠和工作时的配置等 quectel 都已经做好了，用户不需要关心太多，简单按以下方式关闭打开即可。

### 4.2.2. 串口 1 配置

默认情况下串口 1 支持硬件流控，在模块内部显示为/dev/ttyHS0 设备节点。

```
root@mdm9607-perf:~# ls /dev/tty*
/dev/tty      /dev/tty19  /dev/tty3   /dev/tty40  /dev/tty51  /dev/tty62
/dev/tty0     /dev/tty2   /dev/tty30  /dev/tty41  /dev/tty52  /dev/tty63
/dev/tty1     /dev/tty20  /dev/tty31  /dev/tty42  /dev/tty53  /dev/tty7
/dev/tty10    /dev/tty21  /dev/tty32  /dev/tty43  /dev/tty54  /dev/tty8
/dev/tty11    /dev/tty22  /dev/tty33  /dev/tty44  /dev/tty55  /dev/tty9
/dev/tty12    /dev/tty23  /dev/tty34  /dev/tty45  /dev/tty56  /dev/ttyGS0
/dev/tty13    /dev/tty24  /dev/tty35  /dev/tty46  /dev/tty57  /dev/ttyHS0
/dev/tty14    /dev/tty25  /dev/tty36  /dev/tty47  /dev/tty58  /dev/ttyHSL0
/dev/tty15    /dev/tty26  /dev/tty37  /dev/tty48  /dev/tty59
/dev/tty16    /dev/tty27  /dev/tty38  /dev/tty49  /dev/tty6
/dev/tty17    /dev/tty28  /dev/tty39  /dev/tty5
/dev/tty18    /dev/tty29  /dev/tty4   /dev/tty50  /dev/tty61
```

一般情况下，我们推荐客户将此串口用到的 4 个管脚作为串口（默认情况）使用，或者修改为 GPIO 使用，下面介绍一下如何修改串口 1 为 GPIO。

```
--- a/ql-ol-kernel/arch/arm/boot/dts/qcom/mdm9607-mtp.dtsi
+++ b/ql-ol-kernel/arch/arm/boot/dts/qcom/mdm9607-mtp.dtsi
@@ -52,7 +52,7 @@
 };

 &blsp1_uart3 {
-     status = "ok";
+     status = "disabled";
 };
```

编译内核并烧录，具体参考 *KBA\_QuecOpen\_EC2X&AG35\_Quick\_Start* 编译方法；

烧录新固件，如下图发现/dev/ttyHS0 已经不存在，此时串口 1 已经关闭了，这 4 个管脚可以作为 GPIO 使用了。

```
root@mdm9607-perf:~# ls /dev/tty*
/dev/tty      /dev/tty19  /dev/tty3   /dev/tty40  /dev/tty51  /dev/tty62
/dev/tty0     /dev/tty2   /dev/tty30  /dev/tty41  /dev/tty52  /dev/tty63
/dev/tty1     /dev/tty20  /dev/tty31  /dev/tty42  /dev/tty53  /dev/tty7
/dev/tty10    /dev/tty21  /dev/tty32  /dev/tty43  /dev/tty54  /dev/tty8
/dev/tty11    /dev/tty22  /dev/tty33  /dev/tty44  /dev/tty55  /dev/tty9
/dev/tty12    /dev/tty23  /dev/tty34  /dev/tty45  /dev/tty56  /dev/ttyGS0
/dev/tty13    /dev/tty24  /dev/tty35  /dev/tty46  /dev/tty57  /dev/ttyHSL0
/dev/tty14    /dev/tty25  /dev/tty36  /dev/tty47  /dev/tty58
/dev/tty15    /dev/tty26  /dev/tty37  /dev/tty48  /dev/tty59
/dev/tty16    /dev/tty27  /dev/tty38  /dev/tty49  /dev/tty6
/dev/tty17    /dev/tty28  /dev/tty39  /dev/tty5
/dev/tty18    /dev/tty29  /dev/tty4   /dev/tty50  /dev/tty61
```

如将串口 1 当作普通 GPIO 使用，需要注意下面的配置是否为 disabled 的状态：

```
--- a/mdm9607-mtp.dtsi
+++ b/mdm9607-mtp.dtsi
@@ -101,7 +101,7 @@

 /* Add spi3 node ---> gpio0,1,2,3, gale 2018-3-14 */
 &spi_3 {
-     status = "disabled";
+     status = "disabled";
 };
```



### 4.2.3. 调试串口配置

默认用于模块 Linux 调试和日志输出，不建议做任何改动：

调试串口未使能硬件流控，在模块内部显示为/dev/ttyHSL0 设备节点。

```
root@mdm9607-perf:~# ls /dev/tty*
/dev/tty      /dev/tty19   /dev/tty3    /dev/tty40   /dev/tty51   /dev/tty62
/dev/tty0      /dev/tty2    /dev/tty30   /dev/tty41   /dev/tty52   /dev/tty63
/dev/tty1      /dev/tty20   /dev/tty31   /dev/tty42   /dev/tty53   /dev/tty7
/dev/tty10     /dev/tty21   /dev/tty32   /dev/tty43   /dev/tty54   /dev/tty8
/dev/tty11     /dev/tty22   /dev/tty33   /dev/tty44   /dev/tty55   /dev/tty9
/dev/tty12     /dev/tty23   /dev/tty34   /dev/tty45   /dev/tty56   /dev/ttyGS0
/dev/tty13     /dev/tty24   /dev/tty35   /dev/tty46   /dev/tty57   /dev/ttyHS0
/dev/tty14     /dev/tty25   /dev/tty36   /dev/tty47   /dev/tty58   /dev/ttyHSL0
/dev/tty15     /dev/tty26   /dev/tty37   /dev/tty48   /dev/tty59
/dev/tty16     /dev/tty27   /dev/tty38   /dev/tty49   /dev/tty6
/dev/tty17     /dev/tty28   /dev/tty39   /dev/tty5
/dev/tty18     /dev/tty29   /dev/tty4
/dev/tty50
/dev/tty60
/dev/tty61
```

调试串口在设备树对应的是：

文件路径：ql-ol-kernel/arch/arm/boot/dts/qcom/mdm9607-mtp.dtsi

```
&blsp1_uart2 {
    status = "ok";
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&blsp1_uart2_active>;
    pinctrl-1 = <&blsp1_uart2_sleep>;
};
```

### 4.2.4. 串口 2 配置

如果产品定义有 BT 功能，并且采用 UART 方式通信，那么推荐使用这一路 UART 和 蓝牙通信。

默认情况下，这一路串口是打开的，打开后会有如下的设备出现：

```
~ # ls /dev/tty*
/dev/tty      /dev/tty19   /dev/tty3    /dev/tty40   /dev/tty51   /dev/tty62
/dev/tty0      /dev/tty2    /dev/tty30   /dev/tty41   /dev/tty52   /dev/tty63
/dev/tty1      /dev/tty20   /dev/tty31   /dev/tty42   /dev/tty53   /dev/tty7
/dev/tty10     /dev/tty21   /dev/tty32   /dev/tty43   /dev/tty54   /dev/tty8
/dev/tty11     /dev/tty22   /dev/tty33   /dev/tty44   /dev/tty55   /dev/tty9
/dev/tty12     /dev/tty23   /dev/tty34   /dev/tty45   /dev/tty56   /dev/ttyGS0
/dev/tty13     /dev/tty24   /dev/tty35   /dev/tty46   /dev/tty57   /dev/ttyHS0
/dev/tty14     /dev/tty25   /dev/tty36   /dev/tty47   /dev/tty58   /dev/ttyHSL0
/dev/tty15     /dev/tty26   /dev/tty37   /dev/tty48   /dev/tty59
/dev/tty16     /dev/tty27   /dev/tty38   /dev/tty49   /dev/tty6
/dev/tty17     /dev/tty28   /dev/tty39   /dev/tty5
/dev/tty18     /dev/tty29   /dev/tty4
/dev/tty50
/dev/tty60
/dev/tty61
```

如果没有上面对应的设备，并且需要打开，可以检查如下代码是否打开：

```

--- a/mdm9607-mtp.dtsi
+++ b/mdm9607-mtp.dtsi
@@ -57,7 +57,7 @@

/* larryzhang-20190225:The default is high speed for BLSP1_UART5(gpio8,9,10,11)*/
&blsp1_uart5 {
-   status = "disabled";
+   status = "ok";
};

/* UART6: enable gpio20, 21 uart6 by gale */
@@ -106,7 +106,7 @@

/* Add spi5 node ---> gpio8,9,10,11, gale 2018-3-14 */
&spi_5 {
-   status = "disabled";
+   status = "disabled";
};

//add by cullen

```

编译内核并烧录，具体参考 *KBA\_QuecOpen\_EC2X&AG35\_Quick\_Start* 编译方法；  
烧录新固件，如发现多出/dev/ttyHS1 设备节点，此时串口 2 可以工作了。

#### 4.2.5. 串口 3 配置

默认情况下串口 3 未使能，在模块内部不显示串口设备节点，PIN77,PIN78,PIN79,PIN80 被配置为 SPI 接口。

```

~ #
~ # ls /dev/tty*
/dev/tty      /dev/tty19   /dev/tty3    /dev/tty40   /dev/tty51   /dev/tty62
/dev/tty0     /dev/tty2    /dev/tty30   /dev/tty41   /dev/tty52   /dev/tty63
/dev/tty1     /dev/tty20   /dev/tty31   /dev/tty42   /dev/tty53   /dev/tty7
/dev/tty10    /dev/tty21   /dev/tty32   /dev/tty43   /dev/tty54   /dev/tty8
/dev/tty11    /dev/tty22   /dev/tty33   /dev/tty44   /dev/tty55   /dev/tty9
/dev/tty12    /dev/tty23   /dev/tty34   /dev/tty45   /dev/tty56   /dev/ttyGS0
/dev/tty13    /dev/tty24   /dev/tty35   /dev/tty46   /dev/tty57   /dev/ttyHS0
/dev/tty14    /dev/tty25   /dev/tty36   /dev/tty47   /dev/tty58   /dev/ttyHS1
/dev/tty15    /dev/tty26   /dev/tty37   /dev/tty48   /dev/tty59   /dev/ttyHSL0
/dev/tty16    /dev/tty27   /dev/tty38   /dev/tty49   /dev/tty6
/dev/tty17    /dev/tty28   /dev/tty39   /dev/tty5
/dev/tty18    /dev/tty29   /dev/tty4
/dev/tty50
/dev/tty61

```

若用户需要增加额外串口按照以下方式可以打开此串口：

```

--- a/mdm9607-mtp.dtsi
+++ b/mdm9607-mtp.dtsi
@@ -62,7 +62,7 @@

/* UART6: enable gpio20, 21 uart6 by gale */
&blsp1_uart6 {
-   status = "disabled";
+   status = "ok";
+   pinctrl-names = "default","sleep";
+   pinctrl-0 = <&blsp1_uart6_active>;
+   pinctrl-1 = <&blsp1_uart6_sleep>;
@@ -111,7 +111,7 @@

//add by cullen
&spi_6 {
-   status = "ok";
+   status = "disabled";
};

```



编译内核并烧录，具体参考 *KBA\_QuecOpen\_EC2X&AG35\_Quick\_Start* 编译方法。  
烧录新固件，如 `ls /dev/ttyHS*` 多出 `/dev/ttyHSL1` 设备节点，此时串口 3 可以工作了。

另外需要说明的是，模块里面看到的串口序号是串口 driver 根据设备树里多个串口节点排列顺序决定的，若上面的调试串口未使能，那么串口 3 节点名称将变为 `ttyHSL0`。

注：串口 4 和串口 5 不介绍如何修改，对于一般客户，SDC1 接口都是使用在 WIFI 功能上，不单独使用。如果有特别需求，可以联系 FAE 获取修改 patch。

Quectel  
Confidential

# 5 QuecOpen 应用层 API

## 5.1. 用户编程说明

QuecOpen 项目 SDK 中提供了一套完整的用户编程接口；

参考路径：ql-ol-sdk/ql-ol-extsdk/

```
gale@eve-linux02:~/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-extsdk$ ls
docs example include lib target tools
gale@eve-linux02:~/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-extsdk$
```

图中的 lib 目录下包含 quectel 提供的 API 接口库；include 目录是所有 API 的头文件；example 目录是提供的按功能划分的 API 使用示例；

这里我只介绍关于 UART 相关的接口以及参考示例；

在进行 uart 应用程序的编写需要依赖库 libql\_peripheral.a；

头文件：ql\_uart.h

## 5.2. UART API 介绍

流控枚举：选择流控模式

```
typedef enum {
    FC_NONE = 0,    // None Flow Control
    FC_RTSCTS,      // Hardware Flow Control (rtscts)
    FC_XONXOFF      // Software Flow Control (xon/xoff)
}Enum_FlowCtrl;
```

检验位枚举：支持 none,奇校验和偶校验

```
typedef enum {
    PB_NONE = 0,    //none parity check
    PB_ODD,         //odd parity check
    PB_EVEN         //even parity check
}Enum_ParityBit;
```

数据位枚举：支持 5,6,7,8

```
typedef enum {
```

```
DB_CS5 = 5,
DB_CS6 = 6,
DB_CS7 = 7,
DB_CS8 = 8
}Enum_DataBit;
```

停止位枚举：支持 1bit,2bit

```
typedef enum {
    SB_1 = 1,
    SB_2 = 2
}Enum_StopBit;
```

波特率枚举：模块支持的波特率

```
typedef enum {
    B_4800      = 4800,
    B_9600      = 9600,
    B_19200     = 19200,
    B_38400     = 38400,
    B_57600     = 57600,
    B_115200    = 115200,
    B_230400    = 230400,
    B_460800    = 460800,
    B_921600    = 921600,
}Enum_BaudRat;
```

串口属性结构体：

```
typedef struct {
    Enum_BaudRate    baudrate;
    Enum_DataBit     databit;
    Enum_StopBit     stopbit;
    Enum_ParityBit   parity;
    Enum_FlowCtrl    flowctrl;
}ST_UARTDCB;
```

*int QI\_UART\_Open(const char\* port, Enum\_BaudRate baudrate, Enum\_FlowCtrl flowctrl);*

以指定波特率及流控方式打开某串口设备文件，默认检验位为 PB\_NONE，数据位 DB\_CS8，停止位 SB\_1,若需要修改这几个属性，使用 *int QI\_UART\_SetDCB(int fd, ST\_UARTDCB \*dcb)*接口。

参数： port: 设备名，如： /dev/ttyHS0

baudrate: 波特率，见枚举 Enum\_BaudRat，如： B\_9600, B\_115200

flowCtrl: 流控，枚举 Enum\_FlowCtrl

返回值：返回文件描述符，否则返回-1；

*int QI\_UART\_SetDCB(int fd, ST\_UARTDCB \*dcb);*

设置串口属性，包括流控，校验位，数据位，停止位，波特率；

参数: fd: 设备文件描述符;  
dcb: 填充的串口属性结构体;

返回值: 返回 0, 出错返回-1;

*int QI\_UART\_GetDCB(int fd, ST\_UARTDCB \*dcb);*

获取当前串口属性, 包括流控, 校验位, 数据位, 停止位, 波特率;

参数: fd: 设备文件描述符;  
dcb: 串口属性结构体;

返回值: 返回 0, 出错返回-1;

*int QI\_UART\_Read(int fd, char\* buf, unsigned int buf\_len);*

从串口读取指定字节长度内容到 buf; 返回读取的字节长度;

参数: fd: 设备文件描述符;  
buf: 读数据指针;  
buf\_len: 读取长度;

返回值: 返回读取的字节长度;

*int QI\_UART\_Write(int fd, const char\* buf, unsigned int buf\_len);*

从 buf 中写指定长度数据到串口; 返回写入的字节长度;

参数: fd: 设备文件描述符;  
buf: 写数据指针;  
buf\_len: 写入长度;

返回值: 返回写入的字节长度;

*int QI\_UART\_Close(int fd);*

关闭设备文件描述符。

高级串口编程:

以上接口完全可以满足客户的需求;

若果客户为高级 Linux 用户且对串口的特性十分精通, 那么客户可以自行使用下面的接口去设置串口;

*int QI\_UART\_IoCtl(int fd, unsigned int cmd, void\* pValue);*

控制串口设备属性;

参数: fd: 设备文件描述符;  
cmd: 串口 ioctl 请求, 如: TCGETS, TCSETS, TIOCMGET, TIOCMSET 等;  
pValue: 串口设备属性指针

返回值: 成功返回 0, 错误返回-1;

用户可以监听 `uart_fd` 设备文件描述符, 来实现异步通知读取串口数据

参考: `ql-ol-extsdk/example/uart`

# 6 UART 功能测试验证

## 6.1. example 介绍及编译

Example 以主串口为例即/dev/ttyHS0

示例中主线程每隔一秒向串口写数据，打开上位机 COM 口可以收到数据；同时子线程在监听 rx 是否有数据进来，在上位机 COM 口发送数据进来，主串口会收到并打印出来；

```
3
4 #define QL_UART1_DEV "/dev/ttyHS0"
5
6 static int fd_uart = -1;
```

进入到 ql-ol-sdk/ql-ol-extsdk/example/uart 目录，make 生成 example\_uart 可执行程序，可以编译的前提必须是之前进行了交叉编译环境的初始化

source ql-ol-crosstool/ql-ol-crosstool-env-init

```
gale@eve-linux02:~/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-extsdk/example/uart$ make
arm-oe-linux-gnueabi-gcc -march=armv7-a -mfloat-abi=softfp -mfpu=neon -O2 -fexpensive-
de -I/home/gale/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-crosstool/sysroots/armv7a-vfp-neon-oe-
linux-gnueabi/usr/include -I/home/gale/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-crosstool/sysroots/armv7a-vfp-neon-oe-
linux-gnueabi/usr/include/dsutils -I/home/gale/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-crosstool/sysroots/armv7a-vfp-neon-oe-
linux-gnueabi/usr/include/qmi-framework -L./ -L/home/gale/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-extsdk/example/uart/../../lib -lrt -lpthread /home/gale/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-extsdk/example/uart.c -o example_uart
gale@eve-linux02:~/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-extsdk/example/uart$
gale@eve-linux02:~/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-extsdk/example/uart$ ls
example_uart example_uart.c example_uart.o Makefile
```

## 6.2. 功能测试

### 6.2.1. 未使能流控

```
baudrate = atoi(argv[1]);
fd_uart = QL_UART_Open(QL_UART1_DEV, baudrate, FC_NONE);
printf("< open(\"%s\", %d)=%d\n", QL_UART1_DEV, baudrate, fd_uart);
```

1. 编译上传 example\_uart 到模块

使用 adb push <example\_uart 在上位机路径> <模块内部路径, 如/usrdata>

或者

使用串口协议 rz 上传

2. 若使用 OPEN\_EVB, 需要用跳线帽连接 J0202 的 main uart 排针

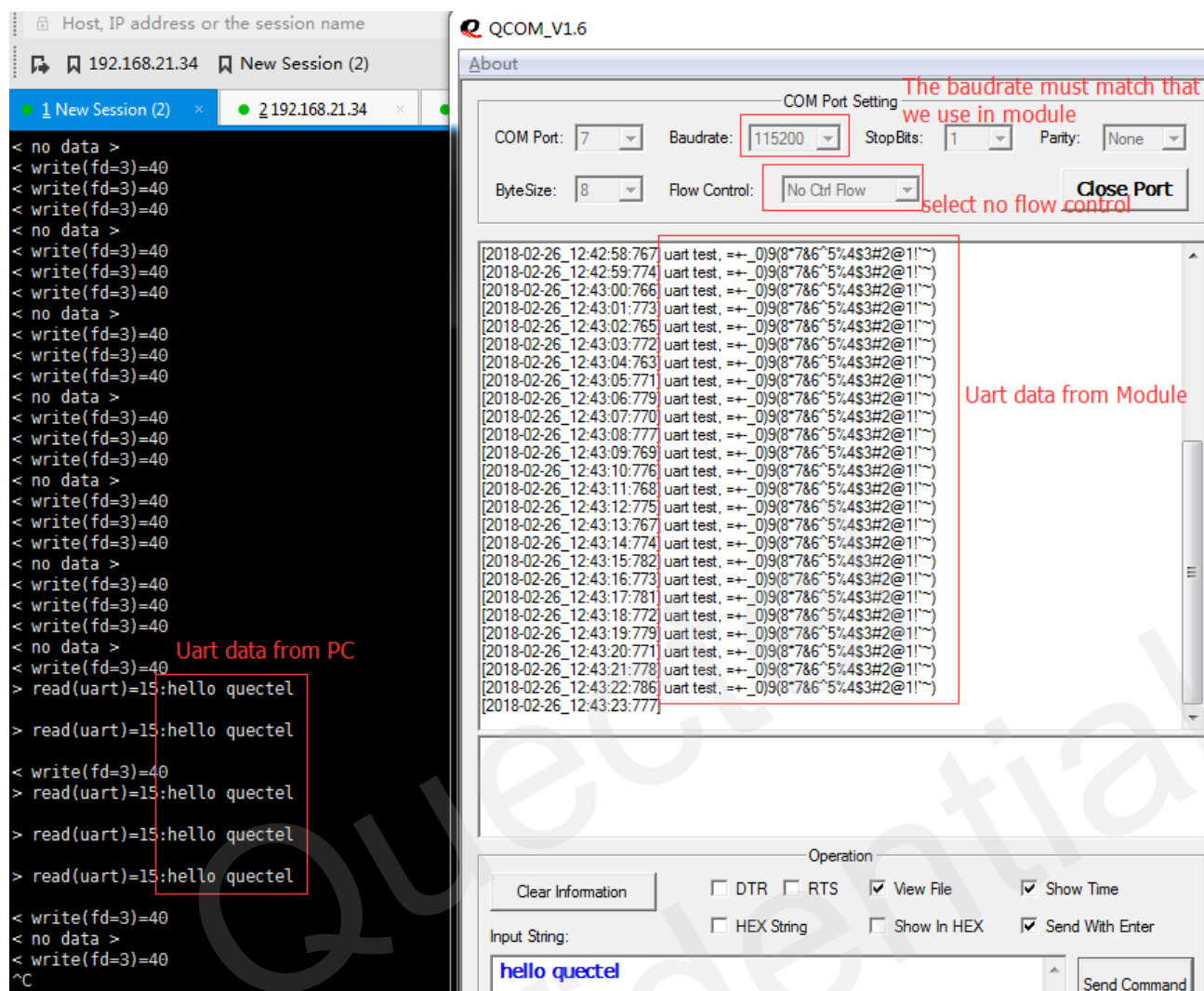
GPIO\_0 连接 MAIN\_TXD

GPIO\_1 连接 MAIN\_RXD

来连通硬件通路

3. 执行 example\_uart 115200, 并在上位机使用对应的波特率以及无流控方式打开对应 COM 口, 如下图所示

```
root@mdm9607-perf:/usrdata# ./example_uart 115200
< OpenLinux: UART example >
< open("/dev/ttyHS0", 115200)=3
```



## 6.2.2. 使能硬件流程

修改 example 来使能硬件流控;

```
fd_uart = QL_UART_Open(QL_UART1_DEV, baudRate, FC_RTSCST);
printf("< open(\"%s\", %d)=%d\n", QL_UART1_DEV, baudRate, fd_uart);
```

1. 编译上传 example\_uart 到模块

使用 adb push <example\_uart 在上位机路径> <模块内部路径, 如/usdata>

或者

使用串口协议 rz 上传

2. 若使用 OPEN\_EVB, 需要用跳线帽连接 J0202 的 main uart 排针

GPIO\_0 连接 MAIN\_TXD

GPIO\_1 连接 MAIN\_RXD

GPIO\_2 连接 MAIN\_RTS

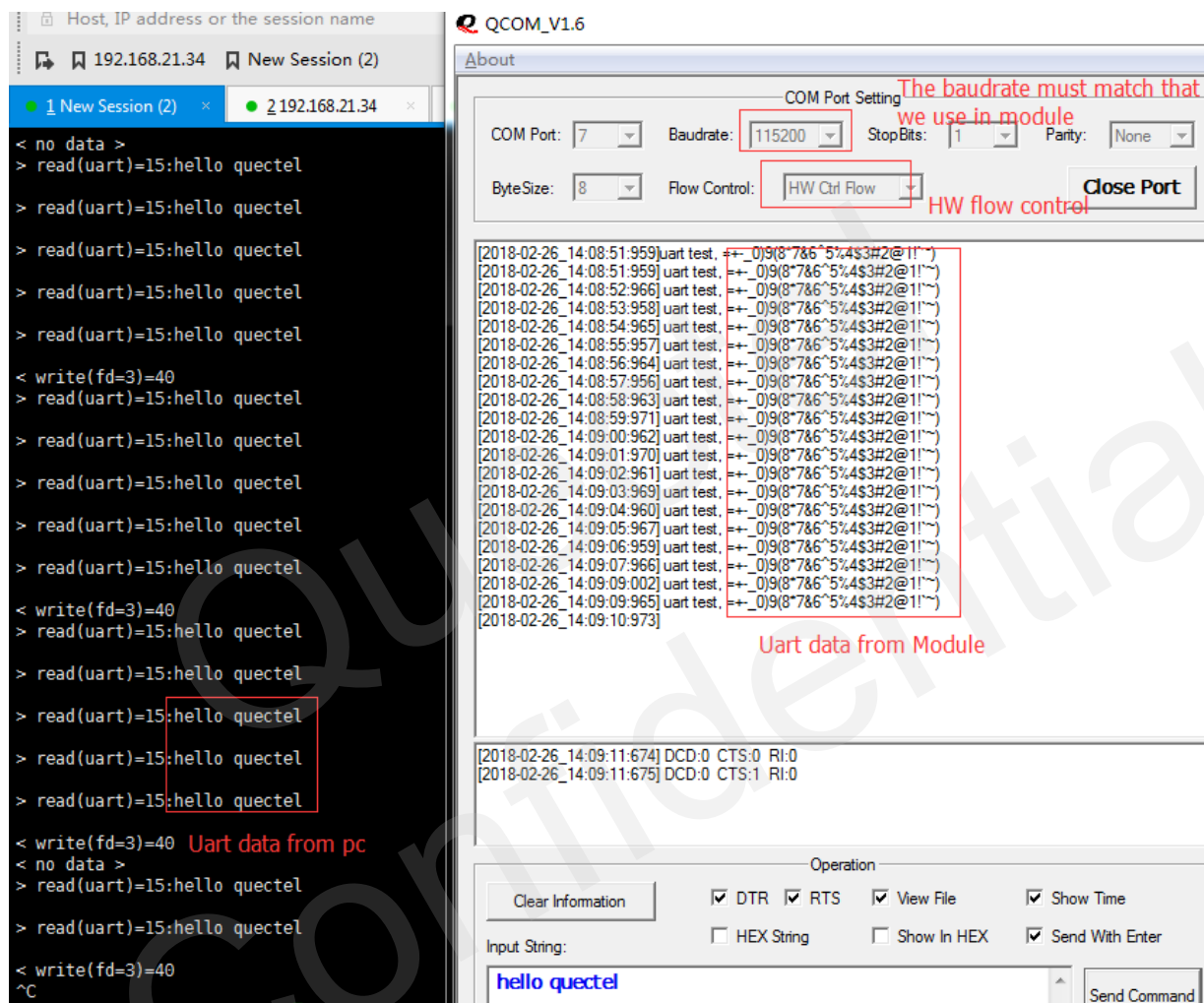
GPIO\_3 连接 MAIN\_CTS

来连通硬件通路



3. 执行 example\_uart 115200, 并在上位机使用对应的波特率以及硬件流控方式打开对应 COM 口, 如下图

```
root@mdm9607-perf:/usrdata# ./example_uart 115200
< OpenLinux: UART example >
< open("/dev/ttyHS0", 115200)=3
```





### 6.2.3. 使能软件流控

#### 6.2.3.1. 软件流控 XON/XOFF 字符说明

XOFF/XON representations in ASCII

| Code | Meaning             | ASCII | Dec | Hex | Keyboard |
|------|---------------------|-------|-----|-----|----------|
| XOFF | Pause transmission  | DC3   | 19  | 13  | Ctrl + S |
| XON  | Resume transmission | DC1   | 17  | 11  | Ctrl + Q |

#### 6.2.3.2. 软件流控测试

修改 example 来使能软件流控；

```
fd_uart = Ql_UART_Open(QL_UART1_DEV, baudRate, FC_XONXOFF);
printf("< open(\"%s\", %d)=%d\n", QL_UART1_DEV, baudRate, fd_uart);
```

1. 编译上传 example\_uart 到模块

使用 adb push <example\_uart 在上位机路径> <模块内部路径，如/usrdta>

或者

使用串口协议 rz 上传

2. 若使用 OPEN\_EVB，需要用跳线帽连接 J0202 的 main uart 排针

GPIO\_0 连接 MAIN\_TXD

GPIO\_1 连接 MAIN\_RXD

来连通硬件通路

3. 执行 example\_uart 115200，并在上位机使用对应的波特率以及硬件流控方式打开对应 COM 口，如下图

```
root@mdm9607-perf:/usrdata# ./example_uart 115200
< OpenLinux: UART example >
< open("/dev/ttyHS0", 115200)=3
```

开启软件流控传输数据时，在上位机串口软件中键盘键入 Ctrl+Shift+S，或者发送十六进制 0x13，模块端将立刻停止数据发送；Ctrl+Shift+Q 或者发送 0x11 又会恢复数据传输，验证软件流控正常。

The screenshot displays the QCOM\_V1.6 software interface, which is used for configuring and monitoring the UART communication between a module and a PC. The interface is divided into several sections:

- Terminal Window (Left):** Shows the command-line interface of the module. It displays a series of commands like `< no data >`, `< write(fd=3)=40`, and `> read(uart)=15:hello quectel`. A red box highlights the `> read(uart)=15:hello quectel` command, with a red arrow pointing to it from the text "Uart data from pc".
- COM Port Setting (Top Right):** A window for configuring the serial port. It shows the COM Port set to 7, Baudrate set to 115200, Stop Bits set to 1, Parity set to None, Byte Size set to 8, and Flow Control set to SW Ctrl Flow. A red box highlights the Baudrate field, with a red arrow pointing to it from the text "The baudrate must match that we use in module". Another red box highlights the Flow Control field, with a red arrow pointing to it from the text "SW flow control".
- Data Log (Center):** A list of log entries showing the time and data received from the module. The data is formatted as `[timestamp] uart test, +=_09(8*786^5%4$3#2@1!~)`. A red box highlights the data log, with a red arrow pointing to it from the text "Uart data from Module".
- Operation (Bottom Right):** A window for controlling the module. It includes checkboxes for DTR, RTS, View File, Show Time, HEX String, Show In HEX, and Send With Enter. A text field for "Input String" contains "hello quectel", and a "Send Command" button is visible.

Annotations and arrows indicate the flow of data and configuration settings:

- A red arrow points from the "Uart data from pc" text to the `> read(uart)=15:hello quectel` command in the terminal window.
- A red arrow points from the "The baudrate must match that we use in module" text to the Baudrate field in the COM Port Setting window.
- A red arrow points from the "SW flow control" text to the Flow Control field in the COM Port Setting window.
- A red arrow points from the "Uart data from Module" text to the data log in the center.
- A red arrow points from the "Input Ctrl+Shift+s stop data immediately from module" text to the data log.
- A red arrow points from the "Input Ctrl+Shift+q restart data immediately from module" text to the data log.