EC2x&AG35-Quecopen

设备树配置指导说明

LTE 系列

版本：EC2x&AG35-Quecopen_设备树配置指导说明_V1.1

日期：2018-09-28

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

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

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

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2017-12-20	高飞虎	初始版本
1.1	2018-09-28	张文郑	新增 PCM 相关属性以及使能设备节点

目录

1 设备树相关文件

QuecOpen 基于 Qualcomm 平台开发，9x07 平台设备树源文件位于 ql-ol-kernel/arch/arm/boot/dts/qcom/ 中，9x28 平台设备树源文件位于 ql-ol-kernel/msm-3.18/arch/arm/boot/dts/qcom/中，使用的 DTB 是 mdm9607-mtp.dtb。

该文件主要由下面几个文件编译生成：

arch/arm/boot/dts/qcom/mdm9607-mtp.dts # dts source file,root of DTB. 仅定义了根节点属性
arch/arm/boot/dts/qcom/mdm9607-pinctrl.dtsi # Pin function define. 主要定义各个 PINs 的功能
arch/arm/boot/dts/qcom/mdm9607.dtsi # platform feature define 平台各部分的配置定义
arch/arm/boot/dts/qcom/mdm9607-mtp.dtsi # list of interface features 功能接口的总开关列表.

要配置项目的 Device Tree，一般只需修改 mdm9607-pinctrl.dtsi，mdm9607.dtsi 和 mdm9607-mtp.dtsi 这三个文件。

2 外围接口的功能对应关系

2.1 9x07 平台

OpenLinux 所有开放的 I/O 接口种类及数量请参考“EC2x_OpenLinux_GPIO_Assignment_Speadsheet”。其中 I/O 接口在设备树上的节点对应关系如下表：

表 1：I/O 接口在设备树上的节点对应关系

PIN Num	Interface	Primary Function	GPIO	Address	Device Node	备注
37	SPI 6 Interface	SPI6_CS	GPIO_22	Spi6 spi@78ba000	/dev/spi	默认使能；与 uart6 复用
38		SPI6_MOSI	GPIO_20			
39		SPI6_MISO	GPIO_21			
40		SPI6_CLK	GPIO_23			
41	I2C-2 interface, host only	I2C2_SCL	GPIO_7	I2c2 i2c@78b6000	/dev/i2c-2	默认使能；用来与 codec 通信
42		I2C2_SDA	GPIO_6			
12	UART-5 interface (DEBUG UART interface)	UART5_TX	GPIO_8	blsp1_uart5: serial@78b3000	/dev/ttyHSL0	默认使能；调试串口，不要修改
11		UART5_RX	GPIO_9			
63	UART-2 interface	UART2_TX	GPIO_4	blsp1_uart2: serial@78b0000	/dev/ttyHSL1	默认使能
66		UART2_RX	GPIO_5			
38	UART-6 interface	UART6_TX	GPIO_20	blsp1_uart6: serial@78b4000	/dev/ttyHSL2	默认不使能；与 SPI 复用，需要修改设备树使能
39		UART6_RX	GPIO_21			

67	UART-3 interface (MAIN UART-interface)	UART3_TXD	GPIO_0	blsp1_uart3: serial@78b1000	/dev/ttyHS0	默认使能
68		UART3_RXD	GPIO_1			
65		UART3_RTS	GPIO_2			
64		UART3_CTS	GPIO_3			

2.2. 9x28 平台

OpenLinux 所有开放的 I/O 接口种类及数量请参考“AG35_QuecOpen_GPIO_Assignment_Speadsheet”。其中 I/O 接口在设备树上的节点对应关系如下表：

表 2：I/O 接口在设备树上的节点对应关系

PIN num	interface	Primary Function	GPIO	Address	Device node	备注
79	SPI 6 Interface	SPI6_CS	GPIO_22	Spi6 spi@78ba000	/dev/spi	默认使能;与 uart6 复用
77		SPI6_MOSI	GPIO_20			
78		SPI6_MISO	GPIO_21			
80		SPI6_CLK	GPIO_23			
74	I2C-2 interface, host only	I2C2_SCL	GPIO_7	I2c2 i2c@78b6000	/dev/i2c-2	默认使能
73		I2C2_SDA	GPIO_6			
43	I2C-4 interface, host only	I2C4_CLK	GPIO_19	I2c4 i2c@78b8000	/dev/i2c-4	默认使能；用来与 codec 通信
42		I2C4_SDA	GPIO_18			
163	UART-5 interface	UART5_TX	GPIO_8	blsp1_uart5: serial@78b3000	/dev/ttyHSL1	默认不使能
165		UART5_RX	GPIO_9			
71	UART-2 interface (DEBUG UART interface)	UART2_TX	GPIO_4	blsp1_uart2: serial@78b0000	/dev/ttyHSL0	默认使能；调试串口，不要修改
72		UART2_RX	GPIO_5			

77	UART-6 interface	UART6_TX	GPIO_20	blsp1_uart6: serial@78b4000	/dev/ttyHSL2	默认不使能；与 SPI 复用，需要修改设备树使能
78		UART6_RX	GPIO_21			
60	UART-3 interface (MAIN UART-interface)	UART3_TXD	GPIO_0	blsp1_uart3: serial@78b1000	/dev/ttyHS0	默认使能
58		UART3_RXD	GPIO_1			
57		UART3_RTS	GPIO_2			
56		UART3_CTS	GPIO_3			

3 设备树相关文件的修改

3.1 UART 相关属性以及使能设备节点

以 uart6 为例，默认不使能（注：此组管脚已被使能为 SPI）；

3.1.1 pin 脚 active 及 sleep 状态时 function 配置

文件路径：arch/arm/boot/dts/qcom/mdm9607-pinctrl.dtsi

uart6 管脚的节点定义，如下图所示。节点 blsp1_uart_active 定义了 uart6 接口处于工作状态的管脚配置，功能配置成 “blsp_uart6”模式；blsp1_uart_sleep 则是 uart6 接口系统挂起时的管脚配置，功能配置成 “GPIO“模式。

```
/* UART6: add gpio20,21 uart6 by gale */
blsp1_uart6_active: blsp1_uart6_active {
        mux {
                pins = "gpio20", "gpio21";
                function = "blsp_uart6";
        };

        config {
                pins = "gpio20", "gpio21";
                drive-strength = <2>;
                bias-pull-down;
        };
};

blsp1_uart6_sleep: blsp1_uart6_sleep {
        mux {
                pins = "gpio20", "gpio21";
                function = "gpio";
        };
        config {
                pins = "gpio20", "gpio21";
                drive-strength = <2>;
                bias-pull-down;
        };
};
```

系统active时配置为的funtion

系统suspend时配置为的funtion

3.1.2 配置其他属性

文件路径：arch/arm/boot/dts/qcom/mdm9607.dtsi

配置其他属性，如匹配的 driver，寄存器地址，中断，时钟等

```
/* UART6: add gpio20,21 uart6 by gale */
blsp1_uart6: serial@78b4000 { /* BLSP1 UART6 */
        compatible = "qcom,msm-lsuart-v14";
        reg = <0x78b4000 0x200>;
        interrupts = <0 122 0>;
        clocks = <&clock_gcc clk_gcc_blsp1_uart6_apps_clk>,
              <&clock_gcc clk_gcc_blsp1_ahb_clk>;
        clock-names = "core_clk", "iface_clk";
        status = "disabled";
};
```

这里配置一系列其他属性，以及对应的driver

3.1.3 UART 总开关

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

使用 mdm9607-pinctl.dtsi 配置的 function,并配置串口功能使能状态；这里 status=“disabled”,表示不使能串口功能。

备注

gpio20,gpio21 复用为 uart，spi，gpio，只能使能一种。要是作为普通的 gpio 使用，就需要将 UART,SPI 对应的 status 设置为 disabled。

```
/* UART6: enable gpio20, 21 uart6 by gale */
&blsp1_uart6 {
        status = "disabled";
        pinctrl-names = "default";
        pinctrl-0 = <&blsp1_uart6_active>;
};
```

这里选中在mdm9607-pinctl.dtsi里面所配置的function； 默认情况下此uart不使能

3.2. I2C 相关属性以及使能设备节点

9x07 模块 I2C 2，用来与 codec 通信
9x28 模块 I2C 4，用来与 codec 通信

这里以 I2C 4 修改为例：

3.2.1 pin 脚 active 及 sleep 状态时 function 配置

文件路径: arch/arm/boot/dts/qcom/mdm9607-pinctrl.dtsi

I2C4 管脚的节点定义，如下图所示。节点 i2c_4_active 定义了 I2C4 接口处于工作状态的管脚配置，功能配置成“blsp_i2c4”模式；i2c_4_sleep 则是 I2C4 接口在系统挂起时的管脚配置，功能配置成“GPIO”模式。


```
i2c_4 {
        i2c_4_active: i2c_4_active {
                /* active state */
                mux {
                        pins = "gpio18", "gpio19";
                        function = "blsp_i2c4";
                };
                config {
                        pins = "gpio18", "gpio19";
                        drive-strength = <2>;
                        bias-disable;
                };
        };

        i2c_4_sleep: i2c_4_sleep {
                /* suspended state */
                mux {
                        pins = "gpio18", "gpio19";
                        function = "gpio";
                };
                config {
                        pins = "gpio18", "gpio19";
                        drive-strength = <2>;
                        bias-pull-down;
                };
        };
};
```

3.2.2 配置其他属性

文件路径：arch/arm/boot/dts/qcom/mdm9607.dtsi

配置其他属性,如 mdm9607-pinctl.dtsi 中的 function，匹配的 driver，寄存器地址，中断，时钟等；

```
i2c_4: i2c@78b8000 { /* BLSP1 QUP4 */
        compatible = "qcom,i2c-msm-v2";
        #address-cells = <1>;
        #size-cells = <0>;
        reg-names = "qup_phys_addr";
        reg = <0x78b8000 0x600>;
        interrupt-names = "qup_irq";
        interrupts = <0 98 0>;
        qcom,clk-freq-out = <400000>;
        qcom,clk-freq-in  = <19200000>;
        clock-names = "iface_clk", "core_clk";
        clocks = <&clock_gcc clk_gcc_blsp1_ahb_clk>,
                 <&clock_gcc clk_gcc_blsp1_qup4_i2c_apps_clk>;

        pinctrl-names = "i2c_active", "i2c_sleep";
        pinctrl-0 = <&i2c_4_active>;
        pinctrl-1 = <&i2c_4_sleep>;
        qcom,noise-rjct-scl = <0>;
        qcom,noise-rjct-sda = <0>;
        qcom,master-id = <86>;
        dmas = <&dma_blsp1 18 64 0x20000020 0x20>,
               <&dma_blsp1 19 32 0x20000020 0x20>;
        dma-names = "tx", "rx";
        status = "disabled";

        //2016-02-23, add by jun.wu
        alc5616_codec@1b{
                compatible = "quec,quec-alc5616-i2c";
                reg = <0x1b>;
        };

        //2016-02-23, add by jun.wu
        nau8814_codec@1a{
                compatible = "quec,quec-nau8814-i2c";
                reg = <0x1a>;
        };
```

i2c其他配置，并选择mdm9607-pinctl.dtsi里面定义的function

3.2.3 I2C 总开关

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

配置串口功能使能状态；这里 status="ok",表示使能 I2C 功能， 如要作为普通 GPIO 使用，将 status 改为 disabled，以及该管脚复用的功能关闭。

```
&i2c_4 {
        status = "ok";
};
```

i2c_4总开关，默认使能

3.3. SPI 相关属性以及使能设备节点

默认使能 SPI，并需要加载 kmod，应用编程请参考 KBA_QuecOpen_Peripheral_API_指导说明_V1.0

3.3.1 pin 脚 active 及 sleep 状态时 function 配置

修改文件：arch/arm/boot/dts/qcom/mdm9607-pinctrl.dtsi

spi6 管脚的节点定义，如下图所示。节点 spi6_default 定义了 spi6 接口处于工作状态的管脚配置，功能配置成“blsp_spi6”模式；spi_sleep 则是 spi6 接口在系统挂起时的管脚配置，功能配置成“GPIO”模式；并配置 CS 片选脚的状态；

```
spi6 {
        spi6_default: spi6_default {
                /* active state */
                mux {
                        /* MOSI, MISO CLK */
                        pins = "gpio20","gpio21","gpio23";
                        function = "blsp_spi6";
                };

                config {
                        pins = "gpio20", "gpio21","gpio23";
                        drive-strength = <12>; /* 12 MA */
                        bias-disable = <0>; /* No PULL */
                };
        };


        spi6_sleep: spi6_sleep {
                /* suspended state */
                mux {
                        /* MOSI, MISO, CLK */
                        pins = "gpio20","gpio21","gpio23";
                        function = "gpio";
                };

                config {
                        pins = "gpio20","gpio21","gpio23";
                        drive-strength = <2>; /* 2 MA */
                        bias-pull-down; /* PULL Down */
                };
        };
        spi6_cs0_active: cs0_active {
                /* CS */
                mux {
                        pins = "gpio22";
                        function = "blsp_spi6";
                };

                config {
                        pins = "gpio22";
                        drive-strength = <2>;
                        bias-disable = <0>;
                };
```

系统active时配置的function，以及驱动能力，上下拉

系统suspend时配置的function，以及驱动能力，上下拉

片选脚配置

3.3.2 配置其他属性

文件路径：arch/arm/boot/dts/qcom/mdm9607.dtsi

配置其他属性,如 mdm9607-pinctl.dtsi 中的 function，匹配的 driver，寄存器地址，中断，时钟等；

```
        /*BLSP QUP6*/
    spi_6: spi@78ba000 {
            compatible = "qcom,spi-qup-v2";
            #address-cells = <1>;
            #size-cells = <0>;
            reg-names = "spi_physical", "spi_bam_physical";
            reg = <0x78ba000 0x600>,
                  <0x7884000 0x2b000>;
            interrupt-names = "spi_irq", "spi_bam_irq";
            interrupts = <0 100 0>, <0 238 0>;
            spi-max-frequency = <19200000>;
            pinctrl-names = "spi_default", "spi_sleep";
            pinctrl-0 = <&spi6_default &spi6_cs0_active>;
            pinctrl-1 = <&spi6_sleep &spi6_cs0_sleep>;
            clocks = <&clock_gcc clk_gcc_blsp1_ahb_clk>,
                     <&clock_gcc clk_gcc_blsp1_qup6_spi_apps_clk>;
            clock-names = "iface_clk", "core_clk";
            qcom,infinite-mode = <0>;
            qcom,use-bam;
            qcom,use-pinctrl;
            qcom,ver-reg-exists;
            qcom,bam-consumer-pipe-index = <22>;
            qcom,bam-producer-pipe-index = <23>;
            qcom,master-id = <86>;
            status = "disabled";
    };
     /*end cullen.wang*/
```

spi其他配置，以及选择mdm9607-pinctl.dtsi中的function

3.3.3 Spi6 总开关

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

配置 SPI 功能使能状态；这里 status=“ok”,表示使能 SPI 功能，如是作为普通的 GPIO 使用，需要将该管脚所有复用的功能关闭。

```
//add by cullen
&spi_6 {
        status = "ok";
};
```

spi总开关，默认使能

3.4. PCM 相关属性以及使能设备节点

默认使能 PCM，用来与 codec 通信,详细说明请参考 Quectel_EC2X&AG35-QuecOpen_PCM 接口配置_V1.1_Preliminary.docx

3.4.1 pin 脚 active 及 sleep 状态时 function 配置

修改文件：arch/arm/boot/dts/qcom/mdm9607-pinctrl.dtsi

PCM 管脚的节点定义，如下图所示。节点 sec_auxpcm_xx_active 定义了 PCM 接口处于工作状态的

管脚配置，功能配置成”sec_mi2s”模式；sec_auxpcm_xx_sleep 则是 PCM 接口在系统挂起时的管脚配置，功能配置成"sec_mi2s"模式；

```
pmx_sec_auxpcm {
                sec_auxpcm_ws_active: sec_auxpcm_ws_active {
                mux {
                        pins = "gpio79";
                        function = "sec_mi2s";
                };
                config {
                        pins = "gpio79";
                        drive-strength = <8>;   /* 8 MA */
                        bias-disable;           /* No PULL */
                        output-high;
                };
        };

        sec_auxpcm_sck_active: sec_auxpcm_sck_active {
                mux {
                        pins = "gpio78";
                        function = "sec_mi2s";
                };

                config {
                        pins = "gpio78";
                        drive-strength = <8>;   /* 8 MA */
                        bias-disable;           /* No PULL */
                        output-high;
                };
        };

        sec_auxpcm_dout_active: sec_auxpcm_dout_active {
                mux {
                        pins = "gpio77";
                        function = "sec_mi2s";
                };
                config {
                        pins = "gpio77";
                        drive-strength = <8>;   /* 8 MA */
                        bias-disable;           /* No PULL */
                        output-high;
                };
        };

        sec_auxpcm_ws_sleep: sec_auxpcm_ws_sleep {
                mux {
                        pins = "gpio79";
                        function = "sec_mi2s";
                };
                config {
                        pins = "gpio79";
                        drive-strength = <2>;   /* 2 MA */
                        bias-pull-down;         /* PULL DOWN */
                };
        };

        sec_auxpcm_sck_sleep: sec_auxpcm_sck_sleep {
                mux {
                        pins = "gpio78";
                        function = "sec_mi2s";
                };
                config {
                        pins = "gpio78";
                        drive-strength = <2>;   /* 2 MA */
                        bias-pull-down;         /* PULL DOWN */
                };
        };
```

```
            sec_auxpcm_dout_sleep: sec_auxpcm_dout_sleep {
                    mux {
                            pins = "gpio77";
                            function = "sec_mi2s";
                    };

                    config {
                            pins = "gpio77";
                            drive-strength = <2>;   /* 2 MA */
                            bias-pull-down;         /* PULL DOWN */
                    };
            };
    };

    pmx_sec_auxpcm_din {
                    sec_auxpcm_din_active: sec_auxpcm_din_active {
                    mux {
                            pins = "gpio76";
                            function = "sec_mi2s";
                    };
                    config {
                            pins = "gpio76";
                            drive-strength = <8>;   /* 8 MA */
                            bias-disable;           /* No PULL */
                    };
            };

            sec_auxpcm_din_sleep: sec_auxpcm_din_sleep {
                    mux {
                            pins = "gpio76";
                            function = "sec_mi2s";
                    };
                    config {
                            pins = "gpio76";
                            drive-strength = <2>;   /* 2 MA */
                            bias-pull-down;         /* PULL DOWN */
                    };
            };
    };
```

3.4.2 配置其他属性

文件路径：arch/arm/boot/dts/qcom/mdm9607.dtsi

配置其他属性,如 mdm9607-pinctl.dtsi 中的 function，匹配的 driver,时钟等；

```
    dai_sec_auxpcm: qcom,msm-sec-auxpcm {
            compatible = "qcom,msm-auxpcm-dev";
            qcom,msm-cpudai-auxpcm-mode = <0>, <0>;
            qcom,msm-cpudai-auxpcm-sync = <1>, <1>;
            qcom,msm-cpudai-auxpcm-frame = <5>, <5>;
            qcom,msm-cpudai-auxpcm-quant = <2>, <2>;
            qcom,msm-cpudai-auxpcm-num-slots = <1>, <1>;
            qcom,msm-cpudai-auxpcm-slot-mapping = <1>, <1>;
            qcom,msm-cpudai-auxpcm-data = <0>, <0>;
            qcom,msm-cpudai-auxpcm-pcm-clk-rate = <2048000>, <40960000>;
            qcom,msm-auxpcm-interface = "secondary";
            pinctrl-names = "default", "idle";
                    pinctrl-0 = <&sec_auxpcm_ws_active
                                 &sec_auxpcm_sck_active
                                 &sec_auxpcm_dout_active
                                 &sec_auxpcm_din_active>;
                    pinctrl-1 = <&sec_auxpcm_ws_sleep
                                 &sec_auxpcm_sck_sleep
                                 &sec_auxpcm_dout_sleep
                                 &sec_auxpcm_din_sleep>;
    };
```

3.4.3 PCM 总开关

PCM 的功能默认是打开的，目前没有直接 status 状态开关。

如想将 PCM 功能对应的 pin 脚作为普通的 GPIO 使用，可以修改如下：

```
diff --git a/mdm9607.dtsi b/mdm9607.dtsi
index 0995a2a..cc7903c 100755
--- a/mdm9607.dtsi
+++ b/mdm9607.dtsi
@@ -1169,7 +1169,7 @@
                qcom,msm-cpudai-auxpcm-pcm-clk-rate = <2048000>, <40960000>;
                qcom,msm-auxpcm-interface = "secondary";
                pinctrl-names = "default", "idle";
-               pinctrl-0 = <&sec_auxpcm_ws_active
+               /*      pinctrl-0 = <&sec_auxpcm_ws_active
                                        &sec_auxpcm_sck_active
                                        &sec_auxpcm_dout_active
                                        &sec_auxpcm_din_active>;
@@ -1177,6 +1177,7 @@
                                        &sec_auxpcm_sck_sleep
                                        &sec_auxpcm_dout_sleep
                                        &sec_auxpcm_din_sleep>;
+               */
        };
```

由于此管脚是一个复用的，所以还需要确认其他地方是否注释，下面代码修改需要注意，先确认 pinctrl-0 在 mdm9607-pinctrl.dtsi 文件中设置的 GPIO 是否为 gpio76/77/78/79，如果是，要修改如下：

```
index 6418b04..8a561b3 100755
--- a/mdm9607.dtsi
+++ b/mdm9607.dtsi
@@ -1218,7 +1218,7 @@
                qcom,msm-cpudai-auxpcm-data = <0>, <0>;
                qcom,msm-cpudai-auxpcm-pcm-clk-rate = <2048000>, <4096000>;     //for pcm 16k, clock rate nede 4096000
                qcom,msm-auxpcm-interface = "primary";
-               /*pinctrl-names = "default", "idle";
+               /* pinctrl-names = "default", "idle";
                        pinctrl-0 = <&pri_auxpcm_ws_active
                                        &pri_auxpcm_sck_active
                                        &pri_auxpcm_dout_active
@@ -1227,7 +1227,7 @@
                                        &pri_auxpcm_sck_sleep
                                        &pri_auxpcm_dout_sleep
                                        &sec_auxpcm_din_sleep>;
-               */
+               */
        };
```

4 重新生成 boot.img

4.1 编译 kernel

ql-ol-sdk$ make kernel
生成 mdm9607-perf-boot.img 在 target 目录下；

4.2 烧写固件

用 target 目录下的 boot.img 替换原固件包中的 boot.img
使用 Quectel_Customer_FW_Download_Tool_V4.30 工具烧写，或者使用 fastboot 烧写。