

How to add an I2C driver on MT8516 Yocto

Chao Zhang

Outline

- Driver dts node
 - Eint
 - gpio/pinctrl
 - power
- Kernel config
- Driver kconfig & makefile

Driver dts node

dts Path: src/kernel/linux/v4.4/arch/arm64/boot/dts/mediatek/xxxx.dts

file: xxxx.dts

```
&i2c0 {  
    ...  
    status = "okay";  
    clock-frequency = <400000>;  
  
    xxx@2c { // add i2c device node into dts file  
        compatible = "xxxx,xxx";  
        reg = <0x2c>; //i2c address  
        ...  
    };  
    ...  
}
```

I2C driver sample code

i2c device driver:

```
static const struct i2c_device_id xxx_id_table[] = {
    { I2C_DRIVER_NAME, 0 },
    {}
};
```

```
static const struct of_device_id xx_of_match[] = {
    {.compatible = "xxxx,xxx"},
    {}
};
```

```
static struct i2c_driver xxx_driver = {
    .driver = {
        .name = DEVICE_DRIVER_NAME,
        .of_match_table = xx_of_match,
        .owner = THIS_MODULE,
    },
    .probe = xxx_probe,
    .remove = xxx_remove,
    .id_table = xxx_id_table,
};
```

```
static void __init xxx_init(void)
{
    int ret = 0;
    //your code here
    i2c_add_driver( &xxx_driver );
    return ret;
}
```

```
static void __exit xxx_exit(void)
{
    //your code here
    i2c_del_driver(&xxx_driver);
}
```

How to config eint

```
5
6     <module_name> {
7         compatible = "mediatek, <moudle_name>";
8         interrupt-parent = <&pio>;
9         interrupts = <eint_num trigger_type>;
10        deb-gpios = <&pio gpio_num flags>;
11        debounce = <debounce time>;
12                                0,16000,32000,64000,256000,512000
13                                unit:us
```

Example:

```
5
6     pmic {
7         compatible = "mediatek,mt6397";
8         interrupt-parent = <&pio>;
9         interrupts = <5 IRQ_TYPE_LEVEL_HIGH>;
10        deb-gpios = <&pio 5 0>;
11        debounce = <16>;
12    };
13
```

Eint usage

- request_irq/request_thread_irq

Method 1:

1. Call platform_get_irq get Linux IRQ number and interrupt flag from xx.dts.
2. Call kernel standard API request_irq/request_thread_irq to register irq handler.

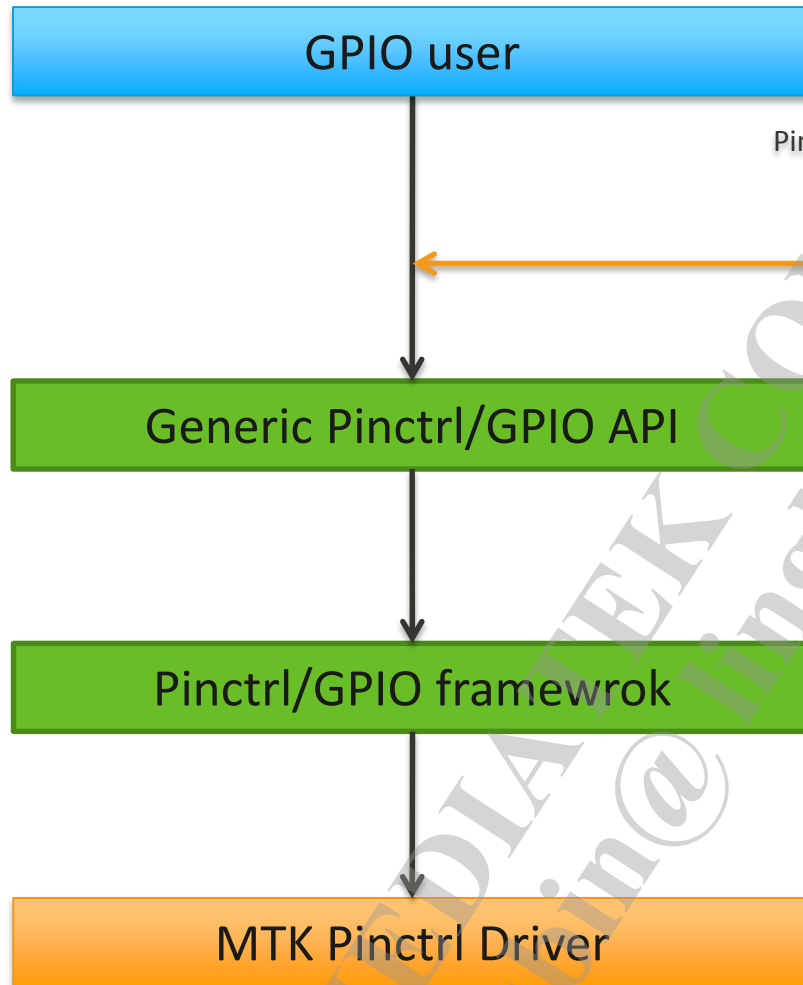
Method 2:

1. Call gpio_to_irq to get Linux IRQ number directly.
2. Call kernel standard API request_irq/request_thread_irq to register irq

```
static int mt6397_probe(struct platform_device *pdev)
{
    u32 ret = 0;
    u32 reg_value = 0;
    int irq;
    void __iomem reg_base;
    struct mt6397_chip *mt6397 = NULL;

    irq = platform_get_irq(pdev, 0);
    ret = request_threaded_irq(irq, NULL, mt6397_irq,
                              IRQF_ONESHOT | IRQF_TRIGGER_NONE, mt6397_irq_chip.name, chip);
    if (ret < 0) {
        pr_info("%s: PMIC master irq request err: %d\n", __func__, ret);
        goto err_free_domain;
    }
}
```

How to config gpio



```
.dts file
&pio {
    pcfg_urx0_sleep: urx0_sleep{
        ...
    };
    pcfg_urx0_default :
    urx0_default{
        ...
    };
};

node1 {
    pinctrl-names = "sleep"
    pinctrl-0 = &
    pcfg_urx0_sleep >;
};

node2 {
    pinctrl-names =
    "default"
    pinctrl-0 =
    &pcfg_urx0_default>;
};

node3 {
    int-gpio = &pio 13 0>;
```

GPIO Usage

- Dts Format

```
node {  
  Names = <&pio, num, flag>;  
};
```

```
&mmc1 {  
  gpios = <&pio 13 0>;  
};
```

```
&u3phy {  
  int-gpio = <&pio 12 0>,  
            <&pio 13 0>,  
            <&pio 14 0>;  
};
```

In you driver:

```
gpio_num = of_get_gpio(np, 0); //to get virtual gpio number  
Or  
gpio_num = of_get_named_gpio(np, "gpios", 0);  
ret = gpio_request(gpio_num, "label");  
If (ret == 0)  
    success;  
set_gpio_value(gpio_num); //to set gpio output value  
Val = get_gpio_value(gpio_num); //to get gpio input/output  
value  
gpio_direction_input(gpio_num);
```

In you driver:

```
gpio_num12 = of_get_named_gpio(np, "int-gpio", 0); //get  
pin12 virtual number  
gpio_num 13 = of_get_named_gpio(np, "int-gpio", 1); //get  
pin12 virtual number  
gpio_num 14 = of_get_named_gpio(np, "int-gpio", 2); //get  
pin12 virtual number
```


Pinctrl APIs Usage

```
76 > /* For a client device requiring named states */
77 > device {
78 >   > pinctrl-names = "active", "idle";
79 >   > pinctrl-0 = <&state_0_node_a>;
80 >   > pinctrl-1 = <&state_1_node_a &state_1_node_b>;
81 > };
```

```
106 > pio: pinctrl@0x10005000 {
107 >   ... /* Standard DT properties */
108 >
109 >   state_0_node_a {
110 >     ...
111 >   };
112 >   state_1_node_a {
113 >     ...
114 >   };
115 >   state_1_node_b {
116 >     ...
117 >   };
118 > }
```

pinctrl_get (struct device *dev)
devm_pinctrl_get (struct device *dev)
eg:
struct pinctrl *p=devm_pinctrl_get(dev);
if(IS_ERR(p)) pr_err("devm_pinctrl_get error");

Pinctrl *

pinctrl_lookup_state (struct pinctrl *p, const char *name)
eg:
pinctrl_state *s = Pinctrl_lookup_state(p, "active");

pinctrl_state *

pinctrl_select_state (struct pinctrl *p, struct pinctrl_state *s)
Eg:
pinctrl_select_state(p, s);

pinctrl_put (struct pinctrl *p) or
devm_pinctrl_put (struct pinctrl *p)

How to config power

Sample code:

```
sound: sound {
    compatible = "mediatek,mt8516-soc-p1";
    mediatek,platform = <&afe>;
    pinctrl-names = "default";
    pinctrl-0 = <&aud_pins_default>;
    extamp-supply = <&extamp_power>;
    //tdmadc-supply = <&tdmadc_power>;
    //tdmadc-1p8v-supply = <&tdmadc_fixed_1v8>;
    //tdmadc-micbias-supply = <&tdmadc_fixed_2v8>;
    cs4382a-rst-supply = <&cs4382a_fixed_3v3>;
    tas5751-rst-supply = <&cs4382a_fixed_3v3>;
    pcm1861-1p8v-supply = <&pcm1861_fixed_1v8>;
    pcm1861-3p3v-supply = <&pcm1861_fixed_3v3>;
    //tdmadc_rst-gpio = <&pio 40 0>;
    mediatek,audio-codec = <&tas5751>;
    status = "okay";
};
```

Power define in pmic(mt6392).dtsi or mt8516-dc2dc.dtsi

Kernel config

Kernel config path:

src/kernel/linux/v4.4/arch/arm64/configs/xxxx_defconfig

src/kernel/linux/v4.4/arch/arm64/configs/xxxx_debug_defconfig

```
398 CONFIG_MMC_MTK=y
399 CONFIG_MMC_MTK_SDIO=y
400 CONFIG_NEW_LEDS=y
401 CONFIG_LEDS_CLASS=y
402 CONFIG_LEDS_LP5521=y
403 CONFIG_LEDS_LP5523=y
404 CONFIG_LEDS_TRIGGERS=y
405 CONFIG_LEDS_TRIGGER_TIMER=y
```

Config LED type is LP5523, because its class is LED, and LP5523 is the next generation of LP5521, so related config are set.

Driver kconfig & make file---case 1

Driver Path: src/kernel/linux/v4.4/drivers/leds/xxxx

Kconfig will explain the CONFIG_LEDS_LP5523

```
248 config LEDS_LP5523
249     tristate "LED Support for TI/National LP5523/55231 LED driver chip"
250     depends on LEDS_CLASS && I2C
251     select LEDS_LP55XX_COMMON
252     help
253         If you say yes here you get support for TI/National Semiconductor
254         LP5523/55231 LED driver.
255         It is 9 channel chip with programmable engines.
256         Driver provides direct control via LED class and interface for
257         programming the engines.
258
```

Makefile was conifged to compile leds-lp5523.c

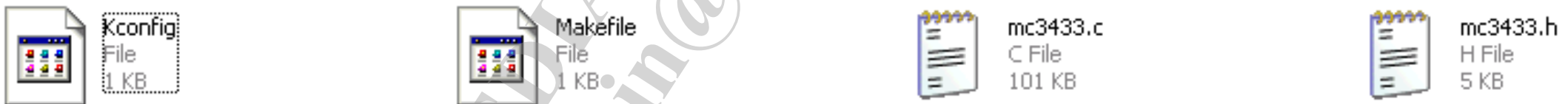
```
30 obj-$(CONFIG_LEDS_LP5521) += leds-lp5521.o
31 obj-$(CONFIG_LEDS_LP5523) += leds-lp5523.o
```

Driver kconfig & make file---Case 2

Driver path: src/kernel/linux/v4.4/drivers/misc/mediatek/accelerometer/xxxx
Take mc3433 as example:



The first layer



The second layer

Driver kconfig & make file ---the first layer

Kconfig of the first layer need to config to search the Kconfig of the second layer

```
19 source "drivers/misc/mediatek/accelerometer/bma2xx/Kconfig"
20 source "drivers/misc/mediatek/accelerometer/mc3433/Kconfig"
21 source "drivers/misc/mediatek/accelerometer/lsm6ds3hg/Kconfig"
22 source "drivers/misc/mediatek/accelerometer/accelhub/Kconfig"
23 source "drivers/misc/mediatek/accelerometer/bmi160-spi/Kconfig"
```

Makefile according to defconfig config to compile mc3433 folder

```
5 ifeq ($(CONFIG_MTK_SCP_SENSORHUB_V1),y)
6 obj-$(CONFIG_MTK_ACCELHUB) += accelhub/
7 else
8 obj-$(CONFIG_MTK_BMA2XX) += bma2xx/
9 obj-$(CONFIG_MTK_MC3433) += mc3433/
10 obj-$(CONFIG_MTK_LSM6DS3HG) += lsm6ds3hg/
11 obj-$(CONFIG_MTK_BMI160_SPI) += bmi160-spi/
12 endif
```

Driver kconfig & make file ---the second layer

Kconfig will explain the CONFIG_MTK_MC3433

```
0 10 20 30 40 50
1 config MTK_MC3433
2     bool "mc3433 g-sensor for MediaTek package"
3     help
4         mc3433 driver support some mc g-sensor chip
5         such as mc34xx, etc .
6         now this driver is used for mc34xx.
```

Makefile config to compile mc3433.c

```
0 10 20 30 40 50 60 70
1 ccflags-y += -I$(srctree)/drivers/misc/mediatek/accelerometer/inc
2 ccflags-y += -I$(srctree)/drivers/misc/mediatek/hwmon/include
3
4 obj-y := mc3433.o
5
```