1. modify code with PWM function

refer readme.txt

(1) open PWM

path:
git/som1_ek/buildroot-at91/output/build/linux-linux4sam-2020.10/arch/arm/boot/dts

file: sama5d2.dtsi

```
//orig
pwm0: pwm@f802c000 {
        compatible = "atmel,sama5d2-pwm";
        reg = <0xf802c000 0x4000>;
        interrupts = <38 IRQ_TYPE_LEVEL_HIGH 7>;
        #pwm-cells = <3>;
        clocks = <&pmc PMC_TYPE_PERIPHERAL 38>;
        status = "disabled";
    };

//改 status = "okay";
pwm0: pwm@f802c000 {
        compatible = "atmel,sama5d2-pwm";
        reg = <0xf802c000 0x4000>;
        interrupts = <38 IRQ_TYPE_LEVEL_HIGH 7>;
        #pwm-cells = <3>;
        clocks = <&pmc PMC_TYPE_PERIPHERAL 38>;
        status = "okay";
    };
```

(2) configure and open pin in PWM

path:
git/som1_ek/buildroot-at91/output/build/linux-linux4sam-2020.10/arch/arm/boot/dts

file: at91-sama5d27_som1_ek.dts

```
//orig
pwm0: pwm@f802c000 {
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_mikrobus1_pwm &pinctrl_mikrobus2_pwm>;
        status = "disabled"; /* Conflict with leds. */
    };

//改 status = "okay";
pwm0: pwm@f802c000 {
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_mikrobus1_pwm &pinctrl_mikrobus2_pwm>;
```

status = "okay"; /* Conflict with leds.*/

// PWM和LED可以使用相同的針腳，可能存在配置衝突

};

(3) LEDS off

path: git/som1_ek/buildroot-at91/output/build/linux-linux4sam-2020.10/arch/arm/boot/dts

file: at91-sama5d27_som1_ek.dts

```
//orig
leds {
    compatible = "gpio-leds";
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_led_gpio_default>;
    status = "okay"; /* Conflict with pwm0. */

//改 status = "disabled"
leds {
    compatible = "gpio-leds";
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_led_gpio_default>;
    status = "disabled"; /* Conflict with pwm0. */
```

(4) recompile and effective process of dts

buildroot-at91/output/build/linux-linux4sam-2020.10

```
# rm .stamp_built
# rm .stamp_target_installed
```

buildroot-at91/output/build/dt-overlay-at91-linux4sam-2020.10

```
# rm .stamp_built
# rm .stamp_target_installed
```

2. observe how the PWM functionality was configured in the Linux kernel

refer to https://microchipdeveloper.com/32mpu:apps-pwm

```
cd /git/som1_ek/buildroot-at91$ make linux-menuconfig
```

3. compile

```
cd /git/som1_ek/buildroot-at91$ make
```

4. SD1 burning via Etcher tool

5. change SD1 to Microchip and try to boot it

the following will setup a 10 kHz 90% duty cycle PWM output:

```
Welcome to the Microchip Demo
sama5 login: root
#
#
# cd /sys/class/pwm/
# ls
pwmchip0
#
# cd pwmchip0/
# echo 1 > export
# ls
device     npwm       pwm1       uevent
export     power      subsystem  unexport
# cd pwm1
# ls
capture     enable      polarity    uevent
duty_cycle  period      power
# echo 100000 > period
# echo 90000 > duty_cycle
# echo 1 > enable
#
```