

CPU Fan control

- Operation confirmed with testing in our Ubuntu image on Kernel **4.14.13** or higher.

Pulse Width Modulation (PWM) PWM is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off.

It is a hardware feature used to control servomotors, for voltage regulation, and so on. The most well-known applications of PWM are:

```
Motor speed control
Light dimming
Voltage regulation
```

The preceding figure describes a complete PWM cycle:

```
Ton: This is the duration during which the signal is high.
Toff: This is the duration during which the signal is low.
Period: This is the duration of a complete PWM cycle. It represents the
sum of
Ton and Toff of the PWM signal.
Duty cycle: It is represented as a percentage of the time signal that
remains on
```

during the period of the PWM signal.

PWM: period: $Ton + Toff$

Duty cycle: $D = (Ton / (Ton + Toff)) * 100 = (Ton / period) * 100$

PWM cpu fans control

Pulse-width modulation (PWM) is a common method of controlling computer fans on Odroid XU3/XU4. Thermal controlled fans are used by sensing the temperature of computing devices and increasing or decreasing the speed of fan control to regulate the temperature inside the computer to keep devices from failing from over heating. The control signal is a square wave operating at 20 kHz, with the duty cycle determining the fan speed. Typically a fan can be driven between about 30% and 100% of the rated fan speed, using a signal with up to 100% duty cycle. The exact speed behavior (linear, off until a threshold value, or a minimum speed until a threshold) at low control levels is manufacturer dependent.

```
fan0: pwm-fan {
    compatible = "pwm-fan";
    pwms = <&pwm 0 20972 0>;
    cooling-min-state = <0>;
    cooling-max-state = <3>;
    #cooling-cells = <2>;
```

```
        cooling-levels = <0 120 180 240>;
};
```

On 4.x kernel as we have mapped the pwm-fan module with the thermal zone to support Active cooling of CPU/GPU.

```
Cooling Device list
-----
cooling_device0
- pwm-fan
cooling_device1
- thermal-cpufreq-0
cooling_device2
- thermal-cpufreq-1
```

It depend on the thermal controller trips temperature to generate pwm signal.

On Odroid XU3/XU4 we would typically change the trips temperature and cooling-level to generate pwm values. Difference cooling level help calculate the duty cycle, once the trip alert cross it will trigger Ton signal and reduce the clk frequency to work on lower as the temperature cools down the it triggers the Toff signal.

The fan speed settings of #cooling-level are specified in percent, so should be in the range 0-254. The PWM duty cycle is specified in the range 0-255, and in automatic mode is calculated by multiplying the fan speed setting by 255 and dividing by 100. For example, in the default case, when the temperature hits 65 degrees it turns the fan on to 60%, which equals a PWM duty cycle of $60 \times 255 / 100 = 153$.

We could change the fan speed on 4.x

```
root@odroid:~# cat /sys/devices/platform/pwm-fan/hwmon/hwmon0/fan_speed
0 120 180 240 root@odroid:~#
root@odroid:~#
root@odroid:~# echo "0 170 210 254" > /sys/devices/platform/pwm-
fan/hwmon/hwmon0/fan_speed
root@odroid:~#
root@odroid:~# cat /sys/devices/platform/pwm-fan/hwmon/hwmon0/fan_speed
0 170 210 254 root@odroid:~#
```

We could alter the thermal trip via below as trip are writable.

```
root@odroid:~# echo 75000 >
/sys/class/thermal/thermal_zone0/trip_point_1_temp
root@odroid:~# cat /sys/class/thermal/thermal_zone0/trip_point_1_temp
75000
```

From:

<http://wiki.odroid.com/> - **ODROID Wiki**

Permanent link:

http://wiki.odroid.com/odroid-xu4/software/pwm_cpufan

Last update: **2018/01/17 14:31**

