NAME

pwmc - PWM (Pulse Width Modulation) control device driver

SYNOPSIS

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device pwmbus
device pwmc
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
pwmc_load="YES"
```

DESCRIPTION

The **pwmc** driver provides device-control access to a channel of PWM hardware. Each instance of a **pwmc** device is associated with a single PWM output channel.

Some PWM hardware is organized with multiple channels sharing a common clock or other resources. In such cases, a separate **pwmc** instance will exist for each channel, but changing the period or duty cycle of any one channel may affect other channels within the hardware which share the same resources. Consult the documentation for the underlying PWM hardware device driver for details on channels that share resources.

An instance of **pwmc** creates a character device named */dev/pwm/pwmcX.Y* where *X* is a sequential number assigned to each PWM hardware controller as it is discovered by the system, and *Y* is the channel number within that hardware controller. The driver can be configured to create aliases that point to the *pwmcX.Y* entries, in effect creating named channels.

The **pwmc** driver provides control of a PWM channel with the following ioctl(2) calls and data structures, defined in <*dev/pwm/pwmc.h*>:

PWMGETSTATE (*struct pwm_state*)
: Retrieve the current state of the channel.

PWMSETSTATE (*struct pwm_state*)
: Set the current state of the channel. All parameters are updated on every call. To change just one of the values, use PWMGETSTATE to get the current state and then submit the same data back with just the appropriate value changed.

The *pwm_state* structure is defined as follows:

```
struct pwm_state {
        u_int           period;
        u_int           duty;
        uint32_t  flags;
        bool            enable;
};
```

period The duration, in nanoseconds, of one complete on-off cycle.

duty The duration, in nanoseconds, of the on portion of one cycle.

flags Flags that affect the output signal can be bitwise-ORed together. The following flags are currently defined:

PWM_POLARITY_INVERTED Invert the signal polarity.

enable
False to disable the output signal or true to enable it.

HINTS CONFIGURATION

On a device.hints(5) based system, such as MIPS, these values are configurable for **pwmc**:

hint.pwmc.%d.at
The pwmbus instance the **pwmc** instance is attached to.

hint.pwmc.%d.channel
The hardware channel number the instance is attached to. Channel numbers count up from zero.

hint.pwmc.%d.label
If this optional hint is set, the driver creates an alias in */dev/pwm* with the given name, which points to the instance.

FDT CONFIGURATION

On an fdt(4) based system, a **pwmc** device is described with a child node of the pwm hardware controller node. When the hardware supports multiple channels within the controller, it is not necessary to include a **pwmc** child node for every channel the hardware supports. Define only the channels you need to control.

The following properties are required for a **pwmc** device node:

compatible
Must be the string "freebsd,pwmc".

reg The hardware channel number.

The following properties are optional for the **pwmc** device node:

label A string containing only characters legal in a file name. The driver creates an alias with the given name in */dev/pwm* which points to the instance's */dev/pwm/pwmcX.Y* device entry.

Example of a PWM hardware node containing one **pwmc** child node:

```
&ehrpwm0 {
   status = "okay";
   pinctrl-names = "default";
   pinctrl-0 = <&ehrpwm0_AB_pins>;

   pwmcontrol@0 {
      compatible = "freebsd,pwmc";
      reg = <0>;
      label = "backlight";
   };
};
```

FILES

*/dev/pwm/pwmc**

SEE ALSO

fdt(4), device.hints(5), pwm(8), pwm(9)

HISTORY

The **pwmc** driver appeared in FreeBSD 13.0.