PWM Device driver

Simple PWM device driver for Colibri-Vf50

Hardware setup:

1. Colibri Evaluation Board V3.1a
2. Connecting Wires
3. CRO

Creating the lodable kernel module:

Compile the driver source using the Makefile in the source directory, we will obtain kernel object (**.ko**) file in the same directory, copy the kernel object to colibri board.

Load the kernel module using the following command:

```
insmod mvf-pwm.ko
```

Check whether the module is loaded or not using the command:

```
lsmod
```

The above command will list the available loaded modules.
Observe the debug messages using command:

```
dmesg
```

We can find a device file in **/dev** after successful loading of the kernel module.
To unload the module use the command:

```
rmmod mvf-pwm.ko
```

Using the device file of mvf-pwm.ko (/dev/pwm) from userspace appliaction:

A simple user space application (mvf-pwm-demo) for using the **/dev/pwm** device file to do ioctl operations and to configure and enable the PWM channels.

Ioctl flags used in driver are as follows:

PWM_INIT (for the initialization)
PWM_CONFIG (for configuring the period, prescale)
PWM_ENABLE (for enabling the pwm channel)
PWM_DISABLE (for disabling pwm channel)
PWM_RELEASE (for de-initializing)

The structure used for the data variables used by the driver is as follows:

```
typedef struct
{
        unsigned long period_ns, dutycycle_ns;
        unsigned int prescale, pwm_chip, pwm_channel;
        bool cpwm;
} pwm_ops;
```

This structure contains the data needed for initialization/configure/enable/disable pwm.

An example for simple ioctl call for generating a PWM signal of 1K HZ on pwm chip3 channel 0 with 50% duty cycle:

1. Initializing PWM :

```
pwm_ops options;
options.pwm_chip = 3;
ioctl(fileDescriptor, PWM_INIT, &options);
```

2. Configuring PWM:

```
pwm_ops options;
options.pwm_chip = 3;
options.prescale = 128;
options.period_hz = 1000;
ioctl(fileDescriptor, PWM_CONFIG, &options);
```

3. enabling PWM:

```
pwm_ops options;
options.pwm_chip = 3;
options.pwm_channel = 0;
options.dutycycel = 50;
ioctl(fileDescriptor, PWM_ENABLE, &options);
```

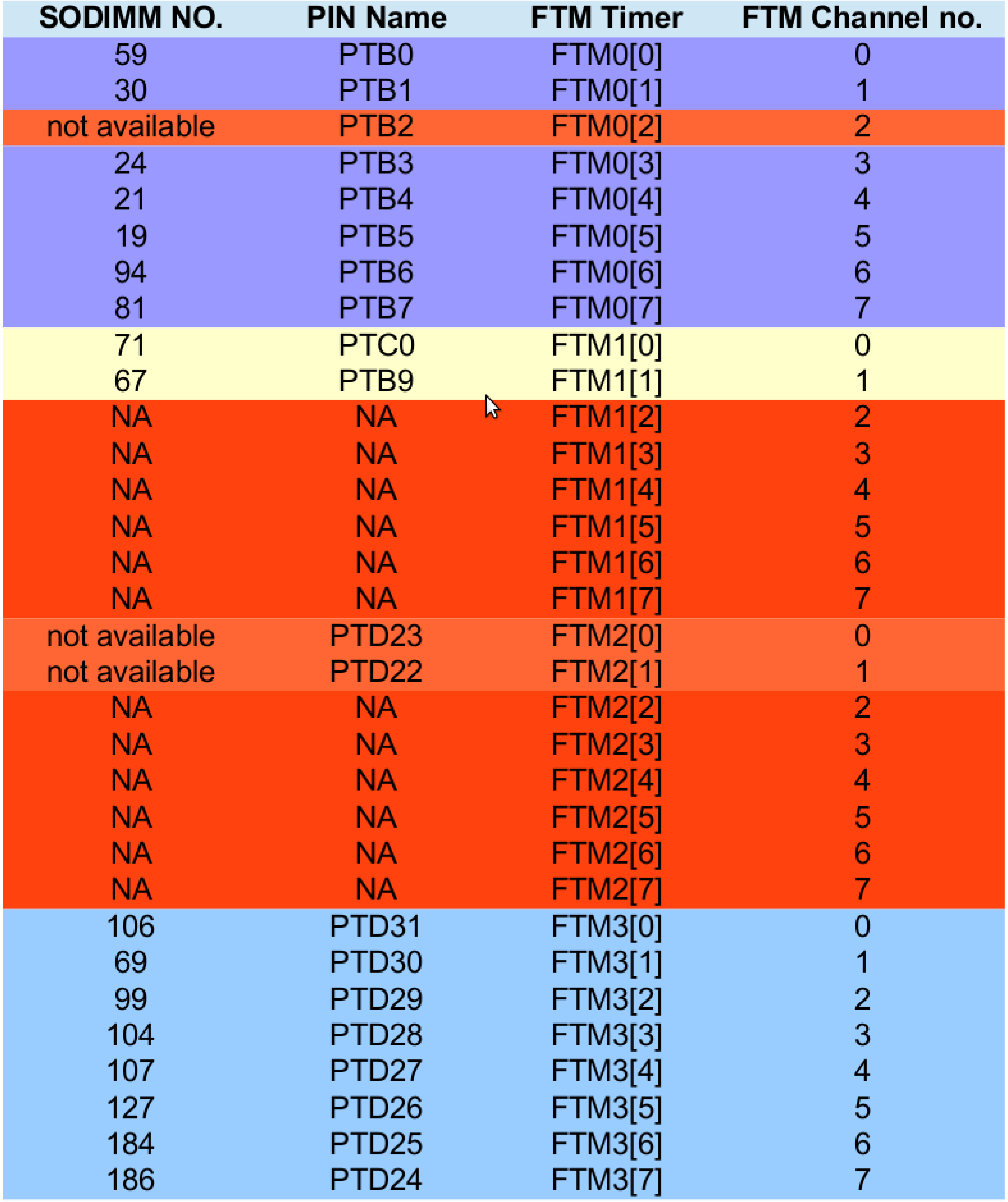
4. Disabling PWM:

```
pwm_ops options;
options.pwm_chip = 3;
options.pwm_channel = 0;
ioctl(fileDescriptor, PWM_DISABLE, &options);
```

5. **Release PWM:**

```
pwm_ops options;
options.pwm_chip = 3;
          ioctl(fileDescriptor, PWM_RELEASE, &options);
```

SODIMM Pin numbers for the PWM channels are shown below:



SODIMM NO.	PIN Name	FTM Timer	FTM Channel no.
59	PTB0	FTM0[0]	0
30	PTB1	FTM0[1]	1
not available	PTB2	FTM0[2]	2
24	PTB3	FTM0[3]	3
21	PTB4	FTM0[4]	4
19	PTB5	FTM0[5]	5
94	PTB6	FTM0[6]	6
81	PTB7	FTM0[7]	7
71	PTC0	FTM1[0]	0
67	PTB9	FTM1[1]	1
NA	NA	FTM1[2]	2
NA	NA	FTM1[3]	3
NA	NA	FTM1[4]	4
NA	NA	FTM1[5]	5
NA	NA	FTM1[6]	6
NA	NA	FTM1[7]	7
not available	PTD23	FTM2[0]	0
not available	PTD22	FTM2[1]	1
NA	NA	FTM2[2]	2
NA	NA	FTM2[3]	3
NA	NA	FTM2[4]	4
NA	NA	FTM2[5]	5
NA	NA	FTM2[6]	6
NA	NA	FTM2[7]	7
106	PTD31	FTM3[0]	0
69	PTD30	FTM3[1]	1
99	PTD29	FTM3[2]	2
104	PTD28	FTM3[3]	3
107	PTD27	FTM3[4]	4
127	PTD26	FTM3[5]	5
184	PTD25	FTM3[6]	6
186	PTD24	FTM3[7]	7