CONFIG_PM or !CONFIG_PM

Media Summit 2024
Vienna, Austria

Laurent Pinchart
laurent.pinchart@ideasonboard.com

```
kernel test robot noticed the following build warnings:

[auto build test WARNING on 431c1646e1f86b949fa3685efc50b660a364c2b6]

url:    https://github.com/intel-lab-lkp/linux/commits/Tomi-Valkeinen/media-uapi-Add-meta-formats-for-PiSP-FE-config-
and-stats/20240904-192729
base:   431c1646e1f86b949fa3685efc50b660a364c2b6
patch link:    https://lore.kernel.org/r/20240904-rp1-cfe-v4-3-f1b5b3d69c81%40ideasonboard.com
patch subject: [PATCH v4 3/4] media: raspberrypi: Add support for RP1-CFE
config: m68k-allmodconfig (https://download.01.org/0day-ci/archive/20240905/202409051822.ZzUGw3XQ-lkp@intel.com/config)
compiler: m68k-linux-gcc (GCC) 14.1.0
reproduce (this is a W=1 build): (https://download.01.org/0day-ci/archive/20240905/202409051822.ZzUGw3XQ-lkp@intel.com/
reproduce)

If you fix the issue in a separate patch/commit (i.e. not just a new version of
the same patch/commit), kindly add following tags
| Reported-by: kernel test robot <lkp@intel.com>
| Closes: https://lore.kernel.org/oe-kbuild-all/202409051822.ZzUGw3XQ-lkp@intel.com/

All warnings (new ones prefixed by >>):

>> drivers/media/platform/raspberrypi/rp1-cfe/cfe.c:2445:12: warning: 'cfe_runtime_resume' defined but not used [-
Wunused-function]
    2445 | static int cfe_runtime_resume(struct device *dev)
         |            ^~~~~~~~~~~~~~~~~~
>> drivers/media/platform/raspberrypi/rp1-cfe/cfe.c:2435:12: warning: 'cfe_runtime_suspend' defined but not used [-
Wunused-function]
    2435 | static int cfe_runtime_suspend(struct device *dev)
```

Problem statement

```
static int cfe_runtime_suspend(struct device *dev)
{
        ...
}

static int cfe_runtime_resume(struct device *dev)
{
        ...
}

static const struct dev_pm_ops cfe_pm_ops = {
        SET_RUNTIME_PM_OPS(cfe_runtime_suspend, cfe_runtime_resume, NULL)
};

static struct platform_driver cfe_driver = {
        .probe          = cfe_probe,
        .remove         = cfe_remove,
        .driver = {
                .name   = CFE_MODULE_NAME,
                .of_match_table = cfe_of_match,
                .pm = &cfe_pm_ops,
        },
};
```

Original code

```
static int __maybe_unused cfe_runtime_suspend(struct device *dev)
{
        ...
}

static int __maybe_unused cfe_runtime_resume(struct device *dev)
{
        ...
}

static const struct dev_pm_ops cfe_pm_ops = {
        SET_RUNTIME_PM_OPS(cfe_runtime_suspend, cfe_runtime_resume, NULL)
};

static struct platform_driver cfe_driver = {
        .probe          = cfe_probe,
        .remove         = cfe_remove,
        .driver = {
                .name   = CFE_MODULE_NAME,
                .of_match_table = cfe_of_match,
                .pm = &cfe_pm_ops,
        },
};
```

Solution 1 – Deprecated

```
static int cfe_runtime_suspend(struct device *dev)
{
        ...
}

static int cfe_runtime_resume(struct device *dev)
{
        ...
}

static const struct dev_pm_ops cfe_pm_ops = {
        RUNTIME_PM_OPS(cfe_runtime_suspend, cfe_runtime_resume, NULL)
};

static struct platform_driver cfe_driver = {
        .probe          = cfe_probe,
        .remove         = cfe_remove,
        .driver = {
                .name   = CFE_MODULE_NAME,
                .of_match_table = cfe_of_match,
                .pm = pm_ptr(&cfe_pm_ops),
        },
};
```

Solution 2 – Recommended

```
config VIDEO_RP1_CFE
        tristate "Raspberry Pi RP1 Camera Front End (CFE) video capture driver"
        depends on VIDEO_DEV
        depends on PM
        select VIDEO_V4L2_SUBDEV_API
        select MEDIA_CONTROLLER
        select VIDEOBUF2_DMA_CONTIG
        select V4L2_FWNODE
        help
          Say Y here to enable support for the Raspberry Pi RP1 Camera Front End.

          To compile this driver as a module, choose M here. The module will be
          called rp1-cfe.
```

Solution 3 - ?

```
static int cfe_runtime_resume(struct device *dev)
{
        ...
        ret = clk_prepare_enable(cfe→clk);
        ...
}

static int cfe_probe(struct platform_device *pdev)
{
        ...

        /* Enable the block power domain */
        pm_runtime_enable(&pdev->dev);

        ret = pm_runtime_resume_and_get(&cfe->pdev->dev);
        if (ret)
                goto err_runtime_disable;

        /* ... Detect and initialize the device ... */

        pm_runtime_put(&cfe->pdev->dev);

        return 0;
}
```

There’s more

```
static int cfe_runtime_resume(struct device *dev)
{
        ...
        ret = clk_prepare_enable(cfe→clk);
        ...
}

static int cfe_probe(struct platform_device *pdev)
{
        ...

        /* Enable the block power domain */
        pm_runtime_enable(&pdev->dev);

        ret = pm_runtime_resume_and_get(&cfe->pdev->dev);
        if (ret)
                goto err_runtime_disable;

        /* ... Detect and initialize the device ... */

        pm_runtime_put(&cfe->pdev->dev);

        return 0;
}
```

◀ **No-op on !CONFIG_PM**

Failure on !CONFIG_PM

```
static int imx290_probe(struct i2c_client *client)
{
        ...
        /*
         * Enable power management. The driver supports runtime PM, but needs to
         * work when runtime PM is disabled in the kernel. To that end, power
         * the sensor on manually here.
         */
        ret = imx290_power_on(imx290);

        /*
         * Enable runtime PM. As the device has been powered manually, mark it
         * as active, and increase the usage count without resuming the device.
         */
        pm_runtime_set_active(dev);
        pm_runtime_get_noresume(dev);
        pm_runtime_enable(dev);

        /* ... Initialize and register the V4L2 subdev ... */

        /*
         * Decrease the PM usage count. The device will get suspended, turning
         * the power off.
         */
        pm_runtime_put(dev);
        ...
}
```

Solution 1 – Recommended

```
static void imx290_remove(struct i2c_client *client)
{
        struct v4l2_subdev *sd = i2c_get_clientdata(client);
        struct imx290 *imx290 = to_imx290(sd);

        v4l2_async_unregister_subdev(sd);
        imx290_subdev_cleanup(imx290);

        /*
         * Disable runtime PM. In case runtime PM is disabled in the kernel,
         * make sure to turn power off manually.
         */
        pm_runtime_disable(imx290->dev);
        if (!pm_runtime_status_suspended(imx290->dev))
                imx290_power_off(imx290);
        pm_runtime_set_suspended(imx290->dev);
}
```

Solution 1 – Recommended

```
static int imx290_probe(struct i2c_client *client)
{
        ...
        /* Enable runtime PM. */
        pm_runtime_enable(dev);

        /* Power the device up. */
        ret = pm_runtime_resume_and_get(dev);
        if (ret < 0) {
                dev_err(dev, "Could not power on the device\n");
                return ret;
        }

        /* ... Initialize and register the V4L2 subdev ... */

        /*
         * Decrease the PM usage count. The device will get suspended, turning
         * the power off.
         */
        pm_runtime_put(dev);
        ...
}
```

Solution 2 – Requires CONFIG_PM

```
static void imx290_remove(struct i2c_client *client)
{
	struct v4l2_subdev *sd = i2c_get_clientdata(client);
	struct imx290 *imx290 = to_imx290(sd);

	v4l2_async_unregister_subdev(sd);
	imx290_subdev_cleanup(imx290);

	/* Disable runtime PM. */
	pm_runtime_disable(imx290->dev);
}
```

Solution 2 – Requires CONFIG_PM

It gets more complex with:

- Runtime PM autosuspend
- Power management of parents
- ACPI platforms (see drivers/media/i2c/ov8856.c)
- Privacy light (see https://lore.kernel.org/r/20240903-imx290-avail-v4-0-e4a6c0837f0b@skidata.com/)

Not the full story

Can we require CONFIG_PM:

- For platform drivers ?
- For USB drivers ?
- For sensor drivers ?

Handling CONFIG_PM

Architecture support

Architectures that support PM

- arc
- arm
- arm64
- loongarch
- m68k (!MMU only)
- mips
- powerpc
- riscv
- sh
- sparc (sparc64 only)
- um
- x86
- xtensa

Architectures that do not support PM

- alpha
- csky
- hexagon
- m68k (MMU)
- microblaze
- nios2
- openrisc
- parisc
- s390
- sparc (!sparc64)

Camera sensor drivers used by USB drivers:

- mt9v011
- ov2640

Handling CONFIG_PM

laurent.pinchart@ideasonboard.com

Contact

? !

Vielen Danke