GPIO Aggregator

The GPIO Aggregator provides a mechanism to aggregate GPIOs, and expose them as a new gpio_chip. This supports the following use cases.

Aggregating GPIOs using Sysfs

GPIO controllers are exported to userspace using /dev/gpiochip* character devices. Access control to these devices is provided by standard UNIX file system permissions, on an all-or-nothing basis: either a GPIO controller is accessible for a user, or it is not.

The GPIO Aggregator provides access control for a set of one or more GPIOs, by aggregating them into a new gpio_chip, which can be assigned to a group or user using standard UNIX file ownership and permissions. Furthermore, this simplifies and hardens exporting GPIOs to a virtual machine, as the VM can just grab the full GPIO controller, and no longer needs to care about which GPIOs to grab and which not, reducing the attack surface.

Aggregated GPIO controllers are instantiated and destroyed by writing to write-only attribute files in sysfs.

/sys/bus/platform/drivers/gpio-aggregator/

"new_device" ...

Userspace may ask the kernel to instantiate an aggregated GPIO controller by writing a string describing the GPIOs to aggregate to the "new_device" file, using the format

System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\gpio\[linux-master][Documentation][admin-guide][gpio]gpio-aggregator.rst, line 35)

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

    [<gpioA>] [<gpiochipB> <offsets>] ...
```

Where:

"<gpioA>" ...
: is a GPIO line name,

"<gpiochipB>" ...
: is a GPIO chip label, and

"<offsets>" ...
: is a comma-separated list of GPIO offsets and/or GPIO offset ranges denoted by dashes.

Example: Instantiate a new GPIO aggregator by aggregating GPIO line 19 of "e6052000.gpio" and GPIO lines 20-21 of "e6050000.gpio" into a new gpio_chip:

```
$ echo 'e6052000.gpio 19 e6050000.gpio 20-21' > new_device
```

"delete_device" ...

Userspace may ask the kernel to destroy an aggregated GPIO controller after use by writing its device name to the "delete_device" file.

Example: Destroy the previously-created aggregated GPIO controller, assumed to be "gpio-aggregator.0":

```
$ echo gpio-aggregator.0 > delete_device
```

Generic GPIO Driver

The GPIO Aggregator can also be used as a generic driver for a simple GPIO-operated device described in DT, without a dedicated in-kernel driver. This is useful in industrial control, and is not unlike e.g. spidev, which allows the user to communicate with an SPI device from userspace.

Binding a device to the GPIO Aggregator is performed either by modifying the gpio-aggregator driver, or by writing to the "driver_override" file in Sysfs.

Example: If "door" is a GPIO-operated device described in DT, using its own compatible value:

```
door {
        compatible = "myvendor,mydoor";

        gpios = <&gpio2 19 GPIO_ACTIVE_HIGH>,
                <&gpio2 20 GPIO_ACTIVE_LOW>;
        gpio-line-names = "open", "lock";
};
```

it can be bound to the GPIO Aggregator by either:

1. Adding its compatible value to `gpio_aggregator_dt_ids[]`,
2. Binding manually using "driver_override":

```
$ echo gpio-aggregator > /sys/bus/platform/devices/door/driver_override
$ echo door > /sys/bus/platform/drivers/gpio-aggregator/bind
```

After that, a new gpiochip "door" has been created:

```
$ gpioinfo door
gpiochip12 - 2 lines:
        line   0:       "open"       unused   input  active-high
        line   1:       "lock"       unused   input  active-high
```