GPIOD Usage

GPIOD Introduction

The GPIO sysfs interface has been deprecated since Linux 4.8. Userspace should instead use `libgpiod` to interact with GPIO character devices.

Key features

- Easy to use: `libgpiod` provides a simple API that makes controlling and operating GPIO devices easy to understand. Developers only need to understand basic GPIO concepts and function interfaces to use it.
- Independence: `libgpiod` is a standalone user-space library that does not depend on any particular hardware platform or chip. It is suitable for a wide range of embedded platforms that support GPIO control, such as the Radxa ROCK series, Raspberry Pi, BeagleBone, and so on.
- Multiple Programming Languages Support: `libgpiod` supports C language natively, and also provides Python bindings (`python3-libgpiod`) and wrappers for other languages, which makes it easy to use in different programming languages.
- Efficient Event Listening: `libgpiod` provides functionality for asynchronous event listening, such as waiting for GPIO state changes. This allows developers to implement efficient event-driven applications.
- GPIO specification support: `libgpiod` provides support for different GPIO specifications and numbering methods, including GPIO numbering and physical pin numbering.

Command Line Tools

`libgpiod` provides a variety of command line tools, for viewing and control GPIO devices.

`gpioinfo`

Display GPIO device information:

```
radxa@radxa-zero2:~$ gpioinfo
gpiochip0 - 85 lines:
        line   0:     "PIN_27"       unused   input  active-high
        line   1:     "PIN_28"       unused   input  active-high
        line   2:      "PIN_7"       unused   input  active-high
        line   3:     "PIN_11"       unused   input  active-high
        line   4:     "PIN_13"       unused   input  active-high
        line   5:     "PIN_15"       unused   input  active-high
        line   6:     "PIN_18"       unused   input  active-high
        line   7:     "PIN_40"       unused   input  active-high
        line   8:      unnamed       unused   input  active-high
        line   9:      unnamed       unused   input  active-high
...
```

`gpiodetect`

List all GPIO controllers in the system:

```
radxa@rock-pi-s:~$ gpiodetect
gpiochip0 [gpio0] (32 lines)
gpiochip1 [gpio1] (32 lines)
gpiochip2 [gpio2] (32 lines)
gpiochip3 [gpio3] (32 lines)
gpiochip4 [gpio4] (32 lines)
```

`gpioget`

Read GPIO input:

```
radxa@radxa-zero2:~$ gpioget gpiochip0 0
1
```

`gpioset`

> **INFO**
>
> Unlike the traditional `sysfs`, by default `gpioset` will exit as soon as the GPIO output has been set. Whether the output will stay at the same level is platform-defined, so you cannot rely on this behavior.

> To persistently set the GPIO level, please follow the below example to set the working mode to `signal`.

Set GPIO output:

```
radxa@radxa-zero2:~$ gpioset -m signal gpiochip0 0=1
(Press Ctrl+C to stop)
```

`gpiomon`

Monitor GPIO events:

```
radxa@radxa-zero2:~$ gpiomon gpiochip0 0
(GPIO event output)
```

`gpiofind`

Find GPIO chip and line:

```
radxa@radxa-zero2:~$ gpiofind PIN_27
gpiochip0 0
```

> **TIP**
>
> You can combine `gpiofind` and `gpioget`/`gpioset`/`gpiomon` to avoid hard-coding GPIO line info in your script:
>
> ```
> radxa@radxa-zero2:~$ gpioget $(gpiofind PIN_27)
> 1
> ```

Programming Reference

C Python

The relevant header files need to be installed before use:

```
sudo apt-get update
sudo apt-get install libgpiod-dev
```

- Sample code
- Official documentation

Convert vendor gpio to libgpiod

Rockchip **Amlogic**

- Naming conventions

There are usually 5 groups (also called banks) of GPIOs on the Rockchip platform: GPIO0 ~ GPIO4 (varies from chip to chip), each group has 32 GPIOs. Each group of GPIOs is divided into 4 groups, A/B/C/D, with 8 GPIOs in each group (0~7, also called bank_idx). rockchip.h)). So the GPIOs can be named from GPIO0_A0 to GPIO4_D7. Taking GPIO3_C5 as an example, we can deduce that its bank is 3 and its bank_idx is 21, i.e. GPIO3_C5 is the 21st GPIO in group 3.

- Correspondence to libgpiod

The GPIO bank of Rockchip platform corresponds to gpiochip in libgpiod, and bank_idx corresponds to gpio line. Take GPIO3_C5 as an example, it is in libgpiod, gpiochip is 3 and line is 21.