How to control a GPIO in userspace

Contents

1 Purpose

This article shows two ways to control a GPIO in userspace:

- using libgpiod
- by writing an application

2 GPIO control through libgpiod

libgpiod provides a C library and tools for interacting with the linux GPIO character device (gpiod stands for GPIOdevice). See the libgpiod repository[1] for further explanation.

- **gpiodetect**
  - List all gpiochips present on the system
  - Usage:

```
Board $> gpiodetect
gpiochip11 [GPIOZ] (16 lines)

...
gpiochip0 [GPIOA] (16 lines)
```

- **gpioinfo**
  - list all lines of specified gpiochips, their names, consumers, and their settings
  - Usage:

```
Board $>gpioinfo
gpiochip11 - 16 lines:
        line   0:      unnamed       unused   input  active-high
        line   1:      unnamed       unused   input  active-high
        ...
```

or

```
                                Comments

Board $>gpioinfo gpiochip0     -->to only print gpiochip0 lines
```

- **gpioget**
  - Read the values of the specified GPIO lines (not valid if the line is already requested)

```
                                Comments

Board $>gpioget gpiochip0 5    -->to get value of GPIOA5
0                              -->means the line is driven low
```

- **gpioset**
  - Set the values of the specified GPIO lines, potentially keeping the lines exported, and wait until timeout, user input or signal.

```
Board $>gpioset gpiochip3 8=1  -->to set GPIOD8 high
```

3 GPIO control through your own application

3.1 Purpose

This application toggles GPIO_A_14 (GPIO bank A, line 14).
On STM32MP15_Evaluation_boards orSTM32MP15_Discovery_kits GPIO_A_14 is connected to the green LED.

This application must be cross compiled with same toolchain as the Kernel.

3.2 Code

```
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ioctl.h>
#include <unistd.h>

#include <linux/gpio.h>

int main(int argc, char **argv)
{
    struct gpiohandle_request req;
    struct gpiohandle_data data;
    char chrdev_name[20];
    int fd, ret;

    strcpy(chrdev_name, "/dev/gpiochip0");
```

3.3 Build application

See Adding_Linux_user_space_applications to build this application.

4 References

1. libgpiod repository