



#### Navigation

[Main Page](#)  
[All pages](#)  
[All categories](#)  
[Popular pages](#)  
[Popular authors](#)  
[Popular categories](#)  
[Category stats](#)  
[Recent changes](#)  
[Random page](#)  
[Help](#)  
[Google Search](#)

[Print/export](#)

[Create a book](#)  
[Download as PDF](#)  
[Printable version](#)

[Toolbox](#)

[What links here](#)  
[Related changes](#)  
[Special pages](#)  
[Permanent link](#)  
[Browse properties](#)

Page [Discussion](#) [ce](#) [View history](#)

# GPIO Driver Guide

(Redirected from [Sitara GPIO Driver User Guide](#))

[GPIO Driver Guide](#)

Translate this page to [cs](#) - [Česky](#)



## Linux PSP GPIO Driver Guide

Linux PSP

### Contents [\[hide\]](#)

- 1 [Introduction](#)
- 2 [Driver Configuration](#)
  - 2.1 [Sysfs entries configuration](#)
- 3 [IRQ handling](#)
- 4 [Driver Usage](#)
  - 4.1 [Kernel Level](#)
  - 4.2 [User Space - Sysfs control](#)
- 5 [References](#)

## Introduction

This wiki page provides the usage information of GPIO Linux driver usage, both in user and kernel space.

Set pinmux configuration for GPIO pin usage, details of pinmux configuration are mentioned on PSP user guide. If the pinmux settings are not proper then expected functionality will not get.

### NOTE

Linux GPIO numbers start from "0".

### NOTE

Some TI SoCs have multiple banks of GPIOs. For example the AM335x has 4 banks of 32 GPIOs, for a total of 128 GPIO, i.e Bank-0[0..31], Bank-1[0..31], Bank-2[0..31], Bank-3[0..31]. In this case Bank-1[0..31] maps to Linux GPIO numbers 32..63, Bank-2[0..31] to Linux GPIO numbers 64..95 etc.

## Driver Configuration

This section describes about the kernel configurations for GPIO driver

The default kernel configuration enables support for GPIO driver (built into the kernel).

To enable or disable GPIO driver from kernel build, follow these steps:

```
$ make CROSS_COMPILE=arm-none-linux-gnueabi- ARCH=arm menuconfig
```

- Select Device Drivers from the main menu.

```
Power management options --->
[*] Networking support --->
Device Drivers --->
File systems --->
...
...
```

- Select GPIO Support from the menu.

```
PPS support --->
PTP clock support --->
-* GPIO Support --->
< > Dallas's 1-wire support --->
...
...
```

## Sysfs entries configuration

GPIO can be access using SYSFS entries from User Space. For that Select `/sys/class/gpio/...` (sysfs interface) from the GPIO support.

```
[ ] Debug GPIO calls
[*] /sys/class/gpio/... (sysfs interface)
*** Memory mapped GPIO drivers: ***
< > Basic memory-mapped GPIO controllers support
...
...
```

- After doing driver selection, exit and save the kernel configuration when prompted.

## IRQ handling

GPIO pin is also used as a interrupt source, these are the general usage of IRQ handling using GPIO lines.

- Map GPIO number to corresponding IRQ number, GPIO 0 need not use IRQ 0

```
irq_num = gpio_to_irq(30)
```

- Request IRQ, make sure that irq\_num should be non-error value

```
request_irq(irq_num, handler, 0, "gpio_test", NULL);
```

- Set IRQ type Raising/Falling/Level triggered

```
set_irq_type(irq_num, IRQ_TYPE_EDGE_RISING);
```

- During the clean-up path free the IRQ and gpio

```
free_irq(irq_num, NULL);
gpio_free(30);
```

## Driver Usage

### Kernel Level

- Allocate memory to GPIO line, can be achieved by doing `gpio_request()`

```
err = gpio_request(30, "sample_name");
```

- Depending on the requirement set GPIO as input or output pin then set gpio value as high or low.

Setting the GPIO pin 30 as input

```
gpio_direction_input(30);
```

Make pin 30 as output and set the value as high.

```
gpio_direction_output(30, 1);
```

Exporting that particular pin (30) to sysfs entry then use this API

```
gpio_export(30, true);
```

Get value from GPIO pin

```
gpio_get_value(30);
```

### User Space - Sysfs control

- Enable GPIO sysfs support in kernel configuration and build the kernel

```
Device Drivers ---> GPIO Support ---> /sys/class/gpio/... (sysfs interface)
```

- Sysfs entries  
Export the particular GPIO pin for user control. GPIO30 is taken as example.

```
$ echo 30 > /sys/class/gpio/export
```

Change the GPIO pin direction to in/out

```
$ echo "out" > /sys/class/gpio/gpio30/direction
```

or

```
$ echo "in" > /sys/class/gpio/gpio30/direction
```

Change the value

```
$ echo 1 > /sys/class/gpio/gpio30/value
```

or

```
$ echo 0 > /sys/class/gpio/gpio30/value
```

Unexport the GPIO pin

```
$ echo 30 > /sys/class/gpio/unexport
```

Note: GPIO's which are used already in the drivers can not be control from sysfs, unless until driver export that particular pin.

Run these commands for knowing what are the GPIO's already requested in the drivers.

```
$ mount -t debugfs debugfs /sys/kernel/debug
```

```
$ cat /sys/kernel/debug/gpio
```

## References

- More details on GPIO APIs usage can be found at [Kernel Documentation](#)
- Refer [kbuild modules](#) document in kernel documentation to know how to build an out-of-tree kernel module.



For technical support please post your questions at <http://e2e.ti.com>. Please post only comments about the article **GPIO Driver Guide** here.

### Links



[ARM Microcontroller MCU](#)

[ARM Processor](#)

[Digital Media Processor](#)

[Digital Signal Processing](#)

[Microcontroller MCU](#)

[Multi Core Processor](#)

[Ultra Low Power DSP](#)

[8 bit Microcontroller MCU](#)

[16 bit Microcontroller MCU](#)

[32 bit Microcontroller MCU](#)

Categories: [TI816X](#) | [TI814X](#) | [AM335x](#) | [Linux](#) | [PSP](#)

[Leave a Comment](#)

This page was last modified on 26 June 2012, at 12:19.

This page has been accessed 7,077 times.

Content is available under [Creative Commons Attribution-Share Alike 3.0 license](#).

[Privacy policy](#) [About Texas Instruments Embedded Processors Wiki](#) [Disclaimers](#)

