

# Device resources

---

This chapter describes a way to access hardware resources via Linux's sysfs.

## GPIO Table

This table shows what GPIO number corresponds with port number on the case.

**DO = Digital Output**

**DI = Digital Input**

**AI = Analog Input**

Port on case	GPIO number
DI1	18
DI2	19
DI3	20
DI4	21
DI5	12
DI6	13
DO7	38
DO8	37
DO9	22
DO10	23
DO11	24
DO12	25

This information is supported by latch GPIO numbers (latch change is required for physical effect on port):

Latch nr.	GPIO number
Latch #1	2
Latch #2	3

Additional elements table:

Element	GPIO number
LED1	43
LED2	47
BUZZER	27

## [What is 'export' file?](#)

Export creates a new folder for the exported pin, and creates files for each of its control functions (i.e. active\_low, direction, edge, power, subsystem, uevent, and value). Upon creation, the control files can be read by all users (not just root), but can only be written to by user root, the file's owner.

Nevertheless, once created, it is possible to allow users other than root, to also write inputs to the control files, by changing the ownership or permissions of these files. Changes to the file's ownership or permissions must initially be done as root, as their owner and group is set to root upon creation. Typically you might change the owner to be the (non root) user controlling the GPIO, or you might add write permission, and change the group ownership to one of which the user controlling the GPIO is a member.

## [Reading DI\[1-6\]](#)

Reading Digital Inputs is quite easy using sysfs.

### **Example: Reading DI1**

- For this example, we'll use DI1 (GPIO18)
- First, export GPIO number (which is 18 for this example), using this command:

```
echo 18 > /sys/class/gpio/export
```

- there should be a folder, called "gpio18" in /sys/class/gpio
- to read DI1 value, execute this command:

```
cat /sys/class/gpio/gpio18/value
```

Sample output:

```
root@modberry ~ $ echo 18 > /sys/class/gpio/export
root@modberry ~ $ cat /sys/class/gpio/gpio18/value
1
root@modberry ~ $
```

## [Setting DO\[7-8\]](#)

Setting Digital Outputs 7 and 8 values are a little bit more tricky. To change their values, you need to reset the two latches.

### **Example: Setting DO7**

- For this example, we'll use DO7 (GPIO38)
- First, export GPIO number (which is 38 for this example), using this command:

```
echo 38 > /sys/class/gpio/export
```

- there should be a folder, called "gpio38" in /sys/class/gpio

- change direction to “out”, otherwise you won't be able to change its value:

```
echo out > /sys/class/gpio/gpio38/direction
```

- change value (either 0 or 1):

```
echo 1 > /sys/class/gpio/gpio38/value
```

- export latches GPIO numbers:

```
echo 2 > /sys/class/gpio/export  
echo 3 > /sys/class/gpio/export
```

- set them both to out:

```
echo out > /sys/class/gpio/gpio2/direction  
echo out > /sys/class/gpio/gpio3/direction
```

- reset them:

```
echo 1 > /sys/class/gpio/gpio2/value  
echo 0 > /sys/class/gpio/gpio2/value  
echo 1 > /sys/class/gpio/gpio3/value  
echo 0 > /sys/class/gpio/gpio3/value
```

- The DO7 should be physically set to logical 1 now

Sample:

```
root@modberry ~ $ echo 38 > /sys/class/gpio/export  
root@modberry ~ $ echo out > /sys/class/gpio/gpio38/direction  
root@modberry ~ $ echo 1 > /sys/class/gpio/gpio38/value  
root@modberry ~ $ echo 2 > /sys/class/gpio/export  
root@modberry ~ $ echo 3 > /sys/class/gpio/export  
root@modberry ~ $ echo out > /sys/class/gpio/gpio2/direction  
root@modberry ~ $ echo out > /sys/class/gpio/gpio3/direction  
root@modberry ~ $ echo 1 > /sys/class/gpio/gpio2/value  
root@modberry ~ $ echo 0 > /sys/class/gpio/gpio2/value  
root@modberry ~ $ echo 1 > /sys/class/gpio/gpio3/value  
root@modberry ~ $ echo 0 > /sys/class/gpio/gpio3/value  
root@modberry ~ $ cat /sys/class/gpio/gpio38/value  
1  
root@modberry ~ $
```

## Setting DO[9-12]

Setting Digital Outputs values are just a little less hassle than 7-8. To change their values, you need to reset one of the two latches.

## Example: Setting DO9

- For this example, we'll use DO9 (GPIO22)
- First, export GPIO number (which is 22 for this example), using this command:

```
echo 22 > /sys/class/gpio/export
```

- there should be a folder, called "gpio22" in /sys/class/gpio
- change direction to "out", otherwise you won't be able to change its value:

```
echo out > /sys/class/gpio/gpio22/direction
```

- change value (either 0 or 1):

```
echo 1 > /sys/class/gpio/gpio22/value
```

- export latch GPIO number:

```
echo 2 > /sys/class/gpio/export
```

- set it to out:

```
echo out > /sys/class/gpio/gpio2/direction
```

- reset them:

```
echo 1 > /sys/class/gpio/gpio2/value  
echo 0 > /sys/class/gpio/gpio2/value
```

- The DO9 should be physically set to logical 1 now

Sample:

```
root@modberry ~ $ echo 22 > /sys/class/gpio/export  
root@modberry ~ $ echo out > /sys/class/gpio/gpio38/direction  
root@modberry ~ $ echo 1 > /sys/class/gpio/gpio38/value  
root@modberry ~ $ echo 2 > /sys/class/gpio/export  
root@modberry ~ $ echo out > /sys/class/gpio/gpio2/direction  
root@modberry ~ $ echo 1 > /sys/class/gpio/gpio2/value  
root@modberry ~ $ echo 0 > /sys/class/gpio/gpio2/value  
root@modberry ~ $ cat /sys/class/gpio/gpio38/value  
1  
root@modberry ~ $
```

## LED Control

To control LED1:

- export LED GPIO number:

```
echo 43 > /sys/class/gpio/export
```

- change direction to 'out':

```
echo out > /sys/class/gpio43/direction
```

- change it's value [0 - LED off, 1 - LED on]

```
echo 1 > /sys/class/gpio43/value
```

Analogically you can control LED2:

- export LED GPIO number:

```
echo 47 > /sys/class/gpio/export
```

- change direction to 'out':

```
echo out > /sys/class/gpio47/direction
```

- change it's value [0 - LED off, 1 - LED on]

```
echo 1 > /sys/class/gpio47/value
```

## RTC

The DS1307 Real Time Clock is connected via i2c bus.  
You can find it in

```
/sys/bus/i2c/devices/0-0068/
```

It's NVRAM is located at:

```
/sys/bus/i2c/devices/0-0068/nvram
```

The time data RTC holds (time, date, seconds since UNIX epoch) are at:

```
/sys/bus/i2c/devices/0-0068/rtc/rtc0/time
```

```
/sys/bus/i2c/devices/0-0068/rtc/rtc0/date
```

```
/sys/bus/i2c/devices/0-0068/rtc/rtc0/since_epoch
```

## Analog Inputs [1-4]

**TO BE DONE**