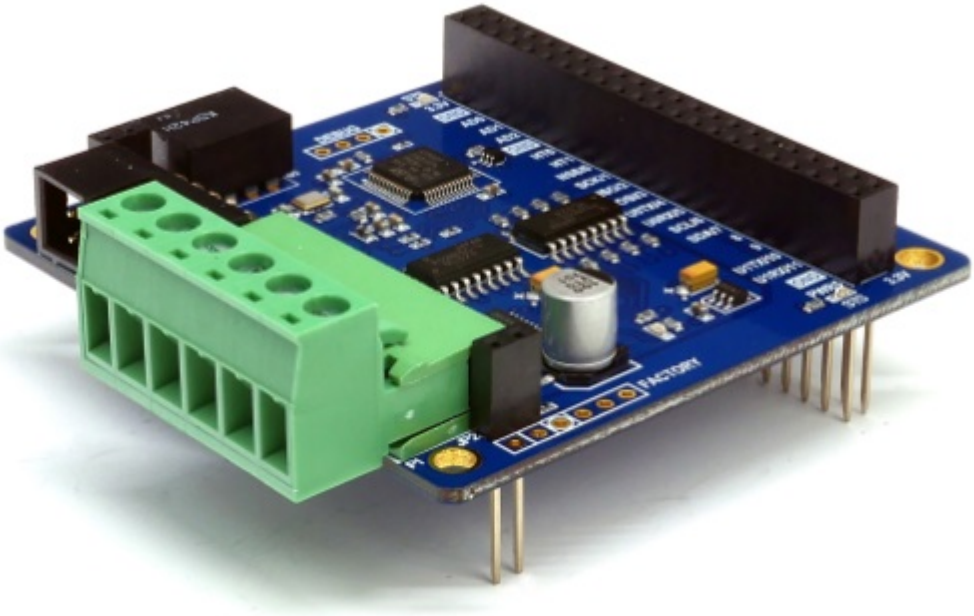
Introduction



PES-2404

PES-2404, DC motor controller, is a smart expansion board for PHPoC boards. With this board, you can easily control two brushed DC motors.

Highlights of PES-2404

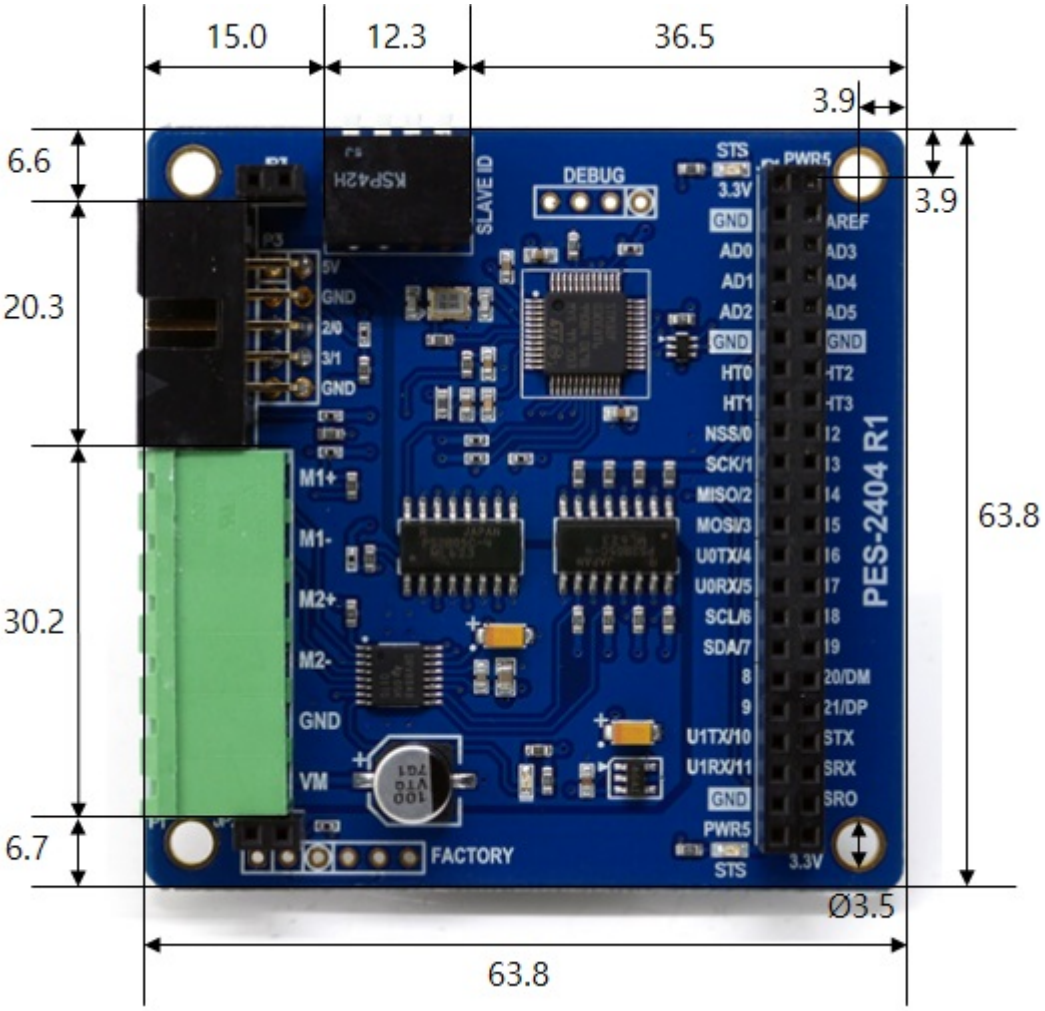
- brushed DC motor controller
- dual DC motor ports with encoder ports
- motor voltage: DC 4 ~ 18[V]
- maximum current on each port: 1[A]
- current comsumption: approximately 65[mA]

Caution: PES-2404 requires a PHPoC board which has a firmware 1.3.0 or higher version.

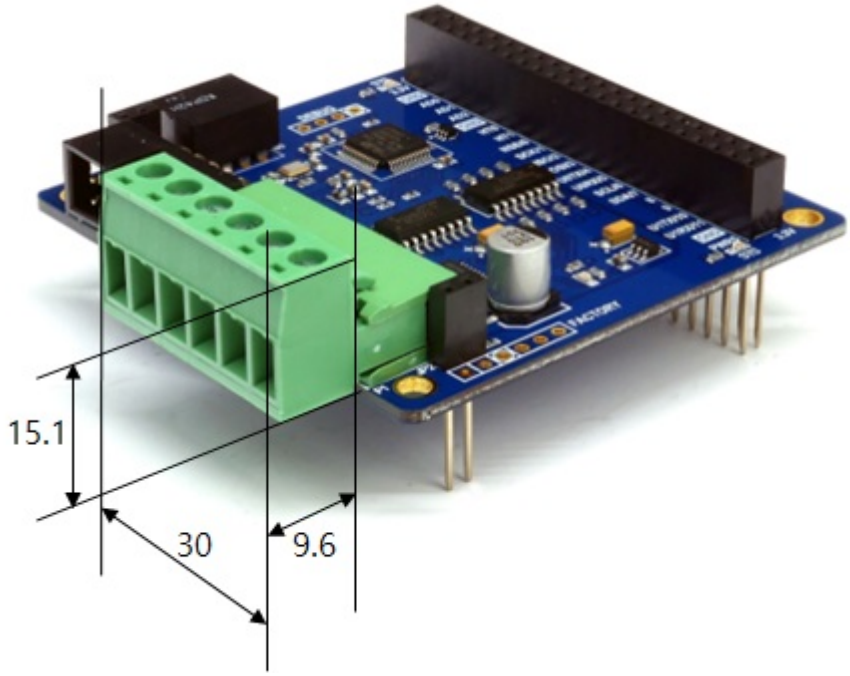
> What is the Smart Expansion Board?
>
> A smart expansion board has own devices and firmware unlike the other expansion boards. This board communicate in a master-slave protocol through the designated port. Two or more smart expansion boards can be connected to one PHPoC board and each of them required to be setting a slave id.

Dimension

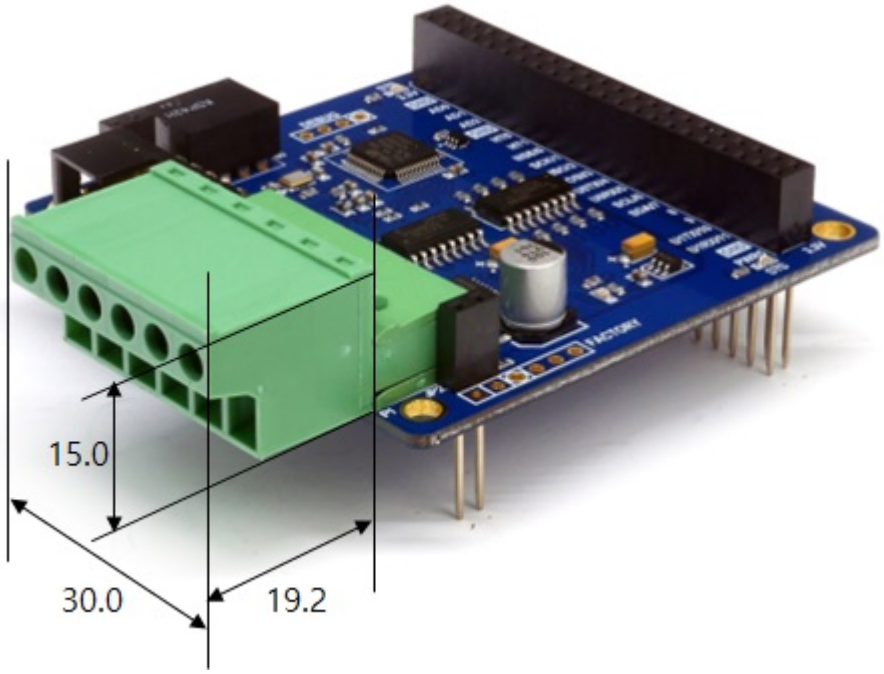
Body



with Terminal Block (T type)



with Terminal Block (S type)



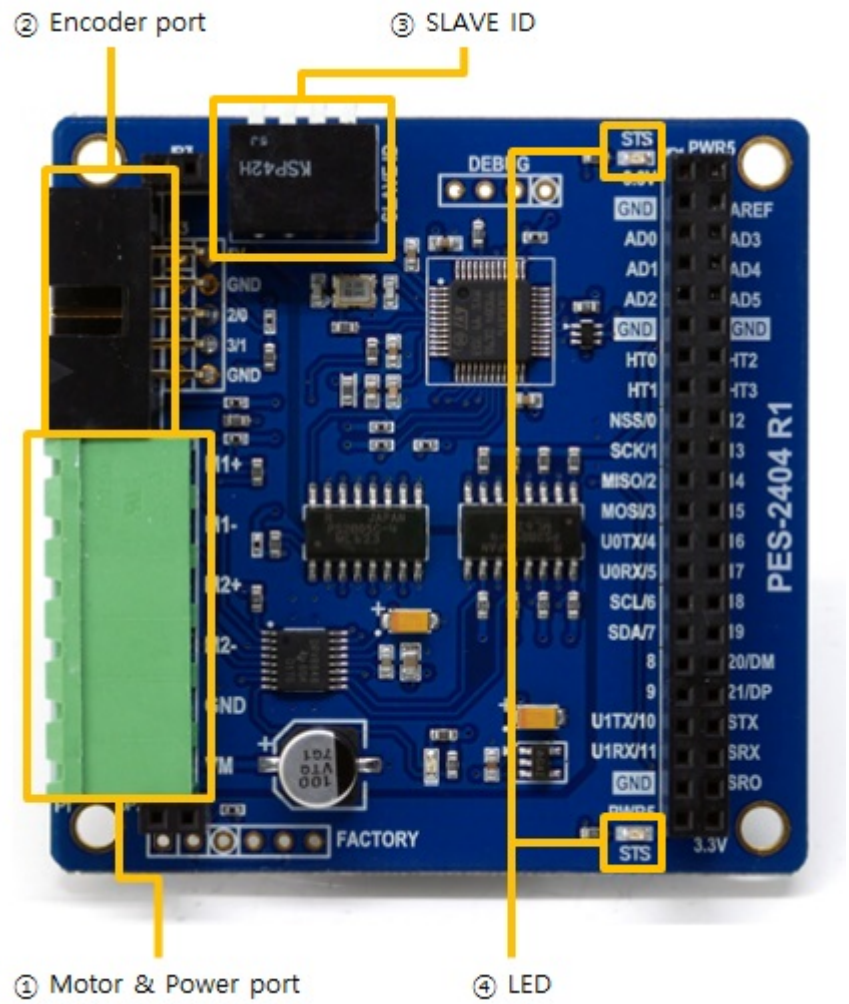
※ Dimensions(unit : mm) may vary according to a method of measurement.

Schematic

This is the schematic of PES-2404.

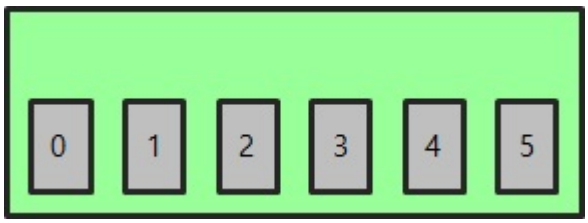
- PES-2404-R2-PO.pdf

Layout


② Encoder port
③ SLAVE ID
① Motor & Power port
④ LED
STS
PWR5
DEBUG
GND
AD0
AD1
AD2
HT0
HT1
NSS/0
SCK/1
MISO/2
MOSI/3
U0TX/4
U0RX/5
SCL/6
SDA/7
U1TX/10
U1RX/11
AREF
AD3
AD4
AD5
PES-2404 R1
FACTORY
3.3V

1. Motor & Power port

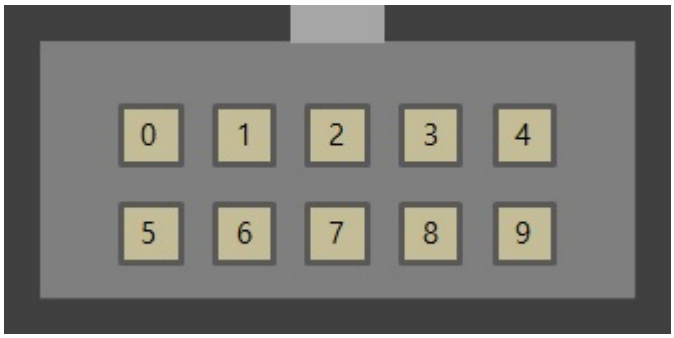
This port is for connecting PES-2404 to motor and the motor's power supply. It has six terminals and a 5mm pitch.



Number	Name	I/O	Description
0	M1+	Out	Motor port 1, positive(+)
1	M1-	Out	Motor port 1, negative(-)
2	M2+	Out	Motor port 2, positive(+)
3	M2-	Out	Motor port 2, negative(-)
4	GND	-	Power supply for motors (Ground)
5	VM	In	Power supply for motors (DC 4 ~ 18V)

2. Encoder port

This port is for connecting encoders.

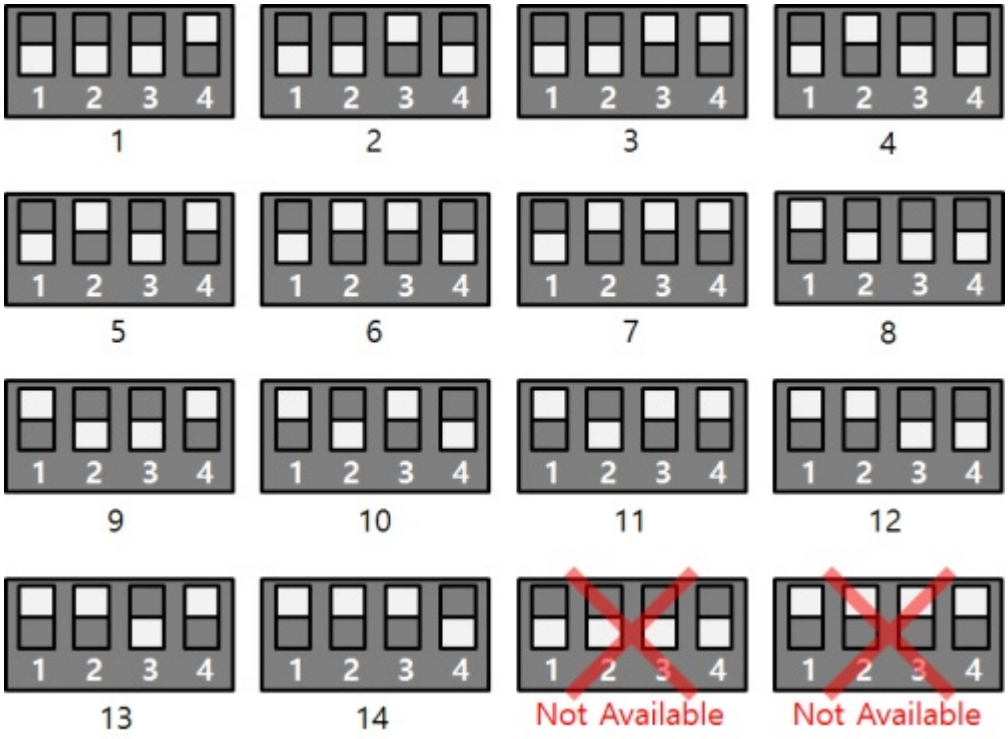


Number	Name	I/O	Description
0	5V	Out	Power supply for encoders (DC 5V)
1	GND	-	Power supply for encoders (Ground)
2	1A	In	Encoder A phase for motor port 1
3	1B	In	Encoder B phase for motor port 1
4	GND	-	Power supply for encoders (Ground)
5	5V	Out	Power supply for encoders (DC 5V)
6	GND	-	Power supply for encoders (Ground)
7	2A	In	Encoder A phase for motor port 2
8	2B	In	Encoder A phase for motor port 2
9	GND	-	Power supply for encoders (Ground)

※ 1A, 1B, 2A and 2B are internally pulled up.

3. SLAVE ID Switch

A slave ID is used when PHPoC board identifies each smart expansion board. So, each smart expansion board, which is connected to a PHPoC board, should have a unique slave ID. The slave ID can be set one of the numbers from 1 to 14 by 4 DIP switches as follows:



4. LED

The PES-2404 board has two STS LEDs. The one, on the top of the board, is connected to 3.3V and the other one, on the bottom of the board is connected to 5V. The operations of both LEDs are the same and they are as follows:

State	Operation
Normal	Repeat On/Off in every second
Invalid slave ID	Blinks very quickly
Fail to communicate with PHPoC	Off

How to Use

The steps for using PES-2404 are as follows.

1. Connect to a PHPoC board

PES-2404 cannot be used alone. Be sure to connect to PHPoC board.

2. Install Software (IDE)

PHPoC Debugger is a software which is used for configuring PHPoC products and developing PHPoC script. It is required to install this software on your PC because PES-2404 must be controlled by PHPoC.

- PHPoC Debugger Download Page
- PHPoC Debugger Manual Page

3. Use SPC Library and Sample Codes

The SPC library is for smart expansion boards such as PES-2404. This library makes it easy for you to use smart expansion boards. Refer to the manual page of SPC library for more information.

- SPC Library Manual Page

Commands and Functions

You can use spc_request_dev or spc_request_sys function when setting or using a smart expansion board.

```
spc_request_dev($sid, $cmd)
spc_request_sys($sid, $cmd)
```

- $sid: slave ID
- $cmd: the command string to transfer to the board.

Common Commands of Smart Expansion Boards

Common commands can be used with spc_request_sys function and a list of the commands is as follows:

Command	Option	Description
get	did	get a device ID
get	uid	get a unique ID

PES-2404 Commands

Dedicated commands for each smart expansion board can be used with spc_request_dev. A list of dedicated commands for PES-2404 is as follows:

- Commands and Functions
- Controlling Motor and Setting PWM
- Setting and Monitoring a Low-pass Filter

PES-2404 Commands

cmd	arg1	arg2	arg3
pwm	set	pol	(+ or -)
		dir	(+ or -)
		period	(1 ~ 1000000)
		width	(1 ~ 1000000)
		decay	(fast or slow)
enc	set	pol	(+, - or 0)
		pos	(-1000000000 ~ +1000000000)
		psr	(1 ~ 64)
	get	pos	-
		period	-
lpf	set	freq	n
		pnc	n
	get	pnc	-

When creating a command string using the above commands, you must specify the port number of the DC motor at the beginning of the string. Port 1 uses dc1 and port 2 uses dc2. The following is an example of a command string.

```
"dc1 pwm set pol +"
"dc2 enc set pos 500"
```

Controlling Motors and Setting PWM

The command for controlling motors and setting PWM is pwm.

The related commands are setting polarity, setting direction of rotation, setting period and HIGH duration of PWM.

Setting PWM polarity

The command to set the PWM polarity is set pol.

```
"dc1 pwm set pol (polarity)"
```

Specify the polarity (+ or -) for polarity. The default value is +. When set to -, the polarity is reversed.

- an example of setting PWM polarity

```
spc_request_dev($sid, "dc1 pwm set pol +"); // normal polarity
spc_request_dev($sid, "dc1 pwm set pol -"); // reverse plarity
```

Setting the direction of rotation

The command to set the direction of rotation is set dir.

```
"dc1 pwm set dir (direction)"
```

- an example of setting direction of rotation

```
spc_request_dev($sid, "dc1 pwm set dir +"); // forward
spc_request_dev($sid, "dc1 pwm set dir -"); // reverse
```

Specify the direction (+ or -) for direction. The default value is +. When set to -, the direction of rotation is reversed.

The direction of rotation is affected by both set pol and set dir.

The value of set pol	The value of set dir	Direction of rotation
+	+	clockwise
+	-	counter clockwise
-	+	counter clockwise
-	-	clockwise

Setting PWM period

The command to set the PWM period is set period.

> "dc1 pwm set period (period_us)"

Specify the period for period_us. The unit is micro-second.

- an example of setting PWM period

```
spc_request_dev($sid, "dc1 pwm set period 10000"); // period: 10 ms
```

Controlling motors(Setting HIGH duration of PWM)

The command to set HIGH duration for motor control is set width.

The HIGH duration is the time during which the HIGH signal is output within one cycle of the PWM signal. Setting the HIGH duration determines the duty cycle of the PWM signal.

> Duty Cycle(%) = HIGH duration / period * 100

In addition, PWM output starts simultaneously with this setting, so this command drives the motor.

- an example of controlling motors

```
<?php
include "/lib/sd_spc.php";
spc_reset();
spc_sync_baud(115200);

$sid = 1;
$width = 3000;

spc_request_dev($sid, "dc1 pwm set pol +");
spc_request_dev($sid, "dc1 pwm set dir +");
spc_request_dev($sid, "dc1 pwm set period 10000");
spc_request_dev($sid, "dc1 pwm set width $width");

while(1)
{
    $width -= 100;

    if($width <= 0)
        break;

    spc_request_dev($sid, "dc1 pwm set width $width");
    usleep(100000);
```

```
}
?>
```

Setting decay mode

The command to set the decay mode is set decay.

```
"dc1 pwm set decay (mode)"
```

Specify the decay mode for mode.

mode	description
fast	fast decay
slow	slow decay

- an example of setting decay mode

```
spc_request_dev($sid, "dc1 pwm set decay fast"); // fast decay
spc_request_dev($sid, "dc1 pwm set decay slow"); // slow decay
```

Setting and Monitoring Encoders

The command for setting and monitoring encoders is enc.

The related commands are setting count direction, setting counter value, monitoring counter values and monitoring periods.

Setting count direction

The command to set the count direction is set pol.

```
"dc1 enc set pol (polarity)"
```

Specify the count direction (+ or -) for polarity. The default value is +.

For an encoder with one sensor, specify 0 for polarity. In this case, the value of pos always increases.

Direction	Direction of rotation	Counter value
+	forward	increase
+	reverse	decrease
-	forward	decrease
-	reverse	increase
0	forward	increase
0	reverse	increase

- an example of setting count direction

```
spc_request_dev($sid, "dc1 enc set pol +");
spc_request_dev($sid, "dc1 enc set pol -");
```

Setting counter values

The command to set the encoder counter value is set pos.

```
"dc1 enc set pos (value)"
```

Specify the counter value for value.

You can set the counter value from -1000000000 (-1 billion) to +1000000000 (1 billion).

- an example of setting a counter value

```
spc_request_dev($sid, "dc1 enc set pos -5000");
```

```
spc_request_dev($sid, "dc1 enc set pos 3000");
```

Setting sampling count

The command to set the encoder sampling count is set psr.

```
"dc1 enc set psr (value)"
```

Specify a sampling count to value.

The encoder sampling count is the number of pulses used to measure the period of the encoder output pulse. The more sampling counts, the smaller the error. The encoder sampling count can be set from 1 to 64.

- an example of setting sampling count

```
spc_request_dev($sid, "dc1 enc set psr 16");
```

Monitoring counter values of encoders

The command to monitor counter values of encoders is get pos.

```
"dc1 enc get pos"
```

- an example of monitoring counter values

```
<?php
include "/lib/sd_spc.php";
spc_reset();
spc_sync_baud(115200);
$sid = 1;

spc_request_dev($sid, "dc1 pwm set period 10000");
spc_request_dev($sid, "dc1 pwm set width 1000");

while(1)
{
    $count = spc_request_dev($sid, "dc1 enc get pos");
    echo "$count₩r₩n";
}
?>
```

Monitoring periods of encoder outputs.

The command to monitor periods of encoder outputs is get period.

```
"dc1 enc get period"
```

The unit of the return values is is microseconds (us).

To reduce the error when monitoring periods of encoder outputs, set the encoder sampling count to a large value.

- an example of monitoring periods of encoder outputs

```
<?php
include "/lib/sd_spc.php";
spc_reset();
spc_sync_baud(115200);
$sid = 1;

spc_request_dev($sid, "dc1 pwm set period 10000");
spc_request_dev($sid, "dc1 pwm set width 1000");
spc_request_dev($sid, "dc1 enc set psr 4");

while(1)
{
    $count = spc_request_dev($sid, "dc1 enc get period");
    echo "$count₩r₩n";
}
?>
```

Setting and Monitoring a Low-pass Filter

The low-pass filter setup and monitoring command for encoder monitoring is "lpf".

The related commands are setting cut-off frequency, setting noise counter and monitoring noise counter.

Setting a cut-off frequency

The command to set a cut-off frequency of the low-pass filter is set freq.

```
"dc1 lpf set freq (frequency)"
```

Specify a cut-off frequency for frequency.

When a cut-off frequency of the low-pass filter is set, signals corresponding to frequencies higher than the cut-off frequency are not counted during encoder monitoring.

- an example of a cut-off frequency

```
spc_request_dev($sid, "dc1 lpf freq 5000");
```

Setting a noise counter

The command to set a noise counter value of the low-pass filter is set pnc.

```
"dc1 lpf set pnc (value)"
```

Specify a noise counter value for value.

- an example of setting a noise counter

```
spc_request_dev($sid, "dc1 lpf set pnc 0");
```

Monitoring noise counter values

The command to monitor noise counter values of the low-pass filter is get pnc.

```
"dc1 lpf get pnc"
```

- an example of monitoring noise counter values

```
<?php
include "/lib/sd_spc.php";
spc_reset();
spc_sync_baud(115200);
$sid = 1;

spc_request_dev($sid, "dc1 pwm set period 10000");
spc_request_dev($sid, "dc1 pwm set width 1000");

$freq = 1000;
$count_prev = 0;

while(1)
{
    if($freq > 7000)
    {
        spc_request_dev($sid, "dc1 pwm set width 0");
        break;
    }
    spc_request_dev($sid, "dc1 lpf set freq $freq");
    $count = (int)spc_request_dev($sid, "dc1 lpf get pnc");
    $diff = $count - $count_prev;
    echo "noise count at freq $freq: $diff₩r₩n";

    $freq += 200;
    $count_prev = $count;
    usleep(200000);
}
?>
```