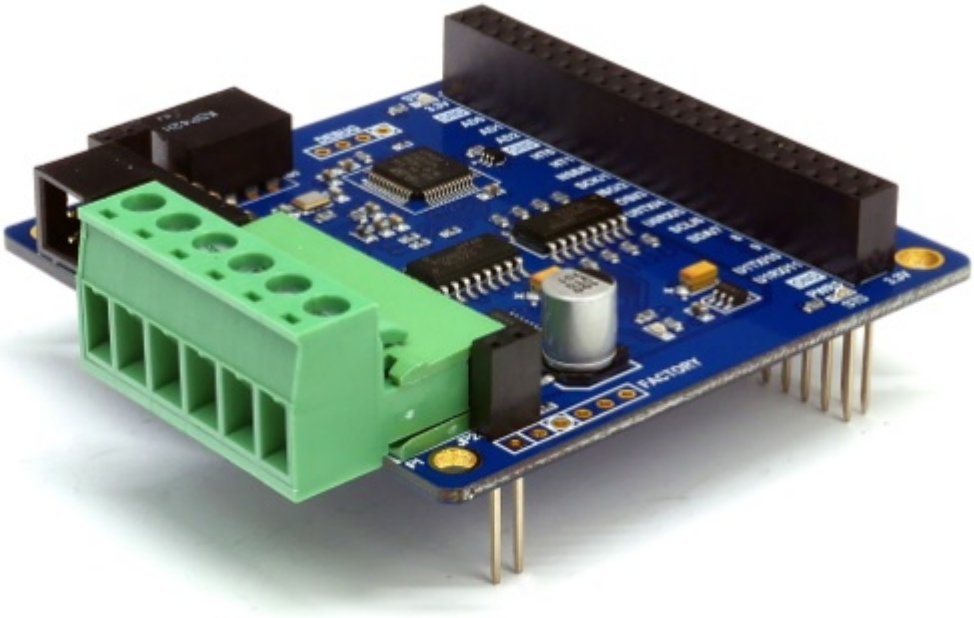
제품 소개



PES-2404

DC모터 제어기 PES-2404는 PHPoC 보드형 제품 전용 스마트 확장보드입니다. 이 보드를 이용하면 2개의 brushed DC모터를 손쉽게 컨트롤 할 수 있습니다.

PES-2404의 주요 특징

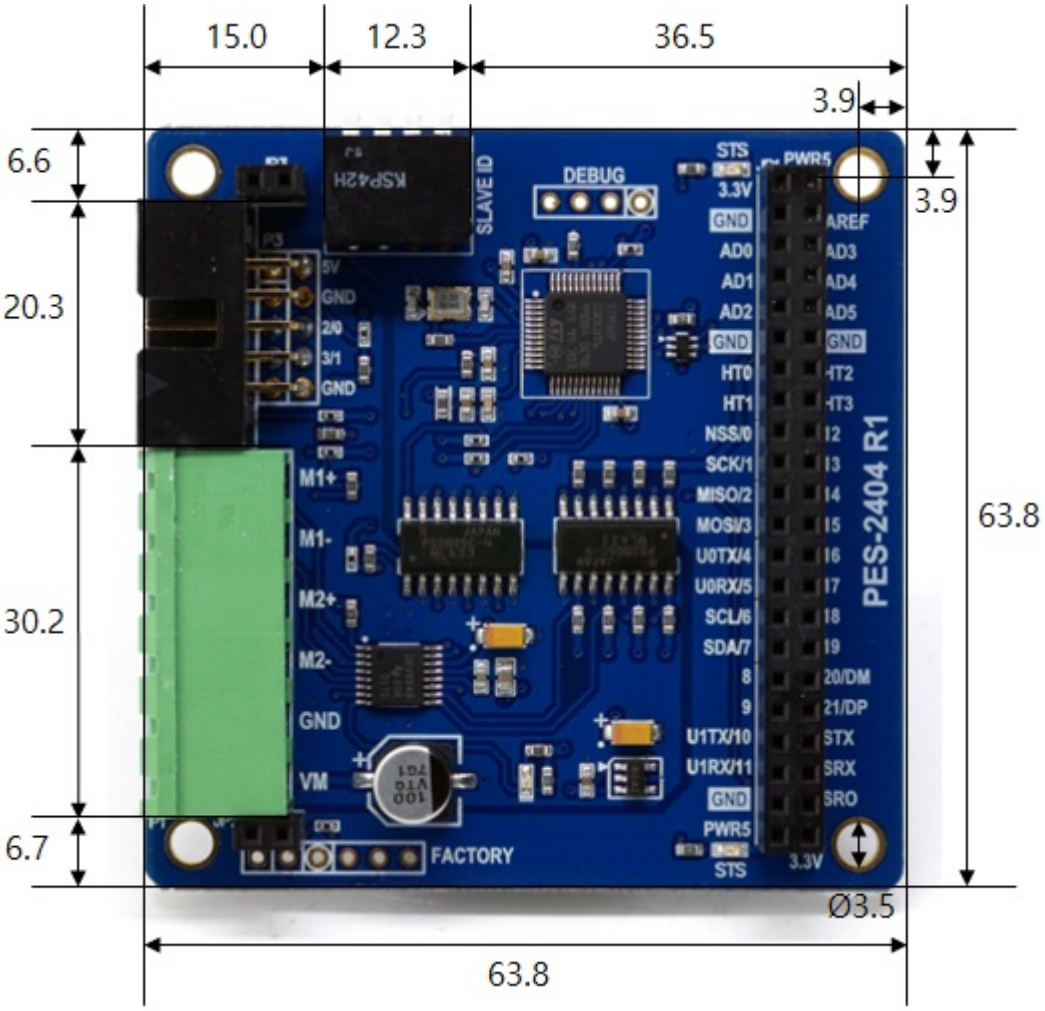
- Brushed DC모터 제어기
- 2개의 DC모터 및 엔코더 연결포트
- 모터 전압: DC 4 ~ 18[V]
- 포트당 최대 모터 전류: 1[A]
- 소비 전류: 약 65[mA]

※ 주의: PES-2404를 사용하기 위해서는 펌웨어 버전 1.3.0이상의 PHPoC 보드가 필요합니다!

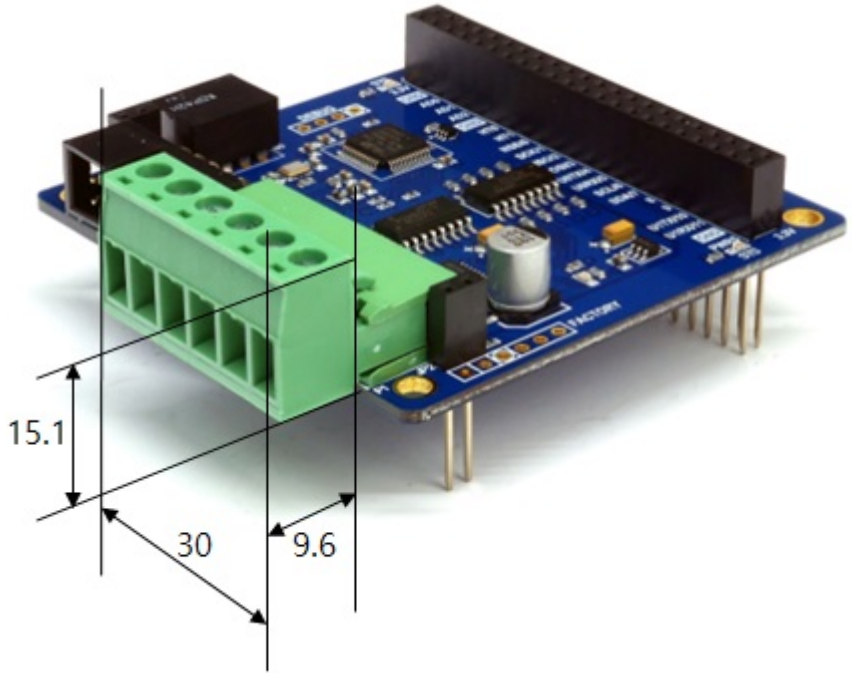
> 스마트 확장보드란?
>
> 스마트 확장보드는 일반 확장보드와는 달리 PHPoC보드의 디바이스 및 펌웨어와는 독립적인 자체 디바이스와 전용 펌웨어를 내장하고 있습니다. 이 보드는 PHPoC 보드와 전용 통신 포트를 이용해 마스터-슬레이브 방식으로 통신합니다. 하나의 PHPoC 보드에 여러 개의 스마트 확장보드를 연결할 수 있으며 각각의 스마트 확장보드에는 반드시 슬레이브 아이디를 설정해야 합니다.

치수

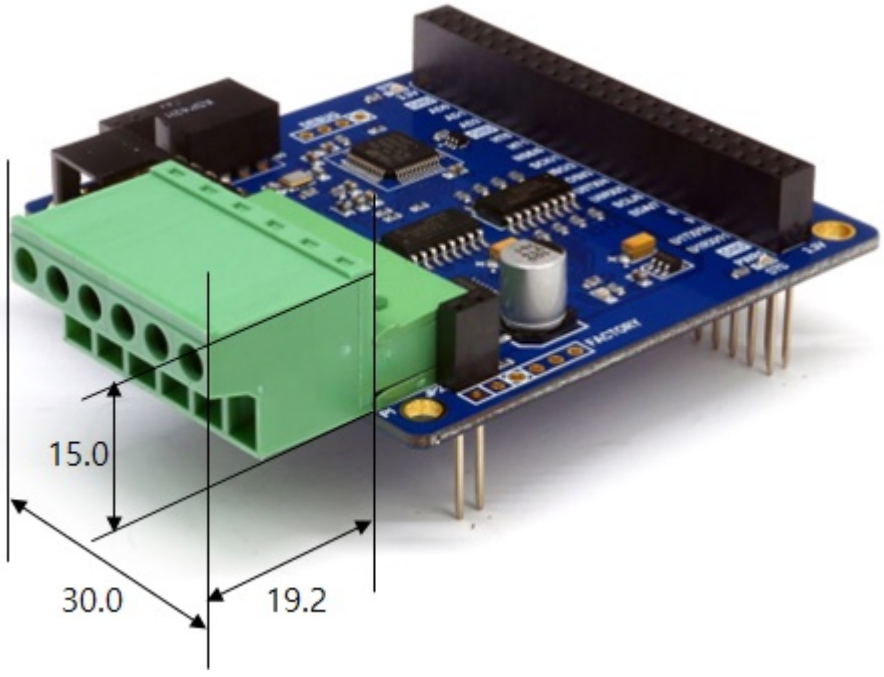
제품 본체


15.0
12.3
36.5
3.9
6.6
20.3
30.2
6.7
3.9
63.8
Ø3.5
63.8
DEBUG
SLAVE ID
FACTORY
PES-2404 R1
M1+
M1-
M2+
M2-
GND
VM
5V
GND
2/0
3/1
GND
STS
3.3V
GND
AD0
AD1
AD2
GND
HT0
HT1
NSS/0
SCK/1
MISO/2
MOSI/3
U0TX/4
U0RX/5
SCL/6
SDA/7
8
9
U1TX/10
U1RX/11
GND
PWR5
STS
PWR5
AREF
AD3
AD4
AD5
GND
HT2
HT3
I2
I3
I4
I5
I6
I7
I8
I9
20/DM
21/DP
STX
SRX
SRO
3.3V

터미널블록(T형)



터미널블록(S형)



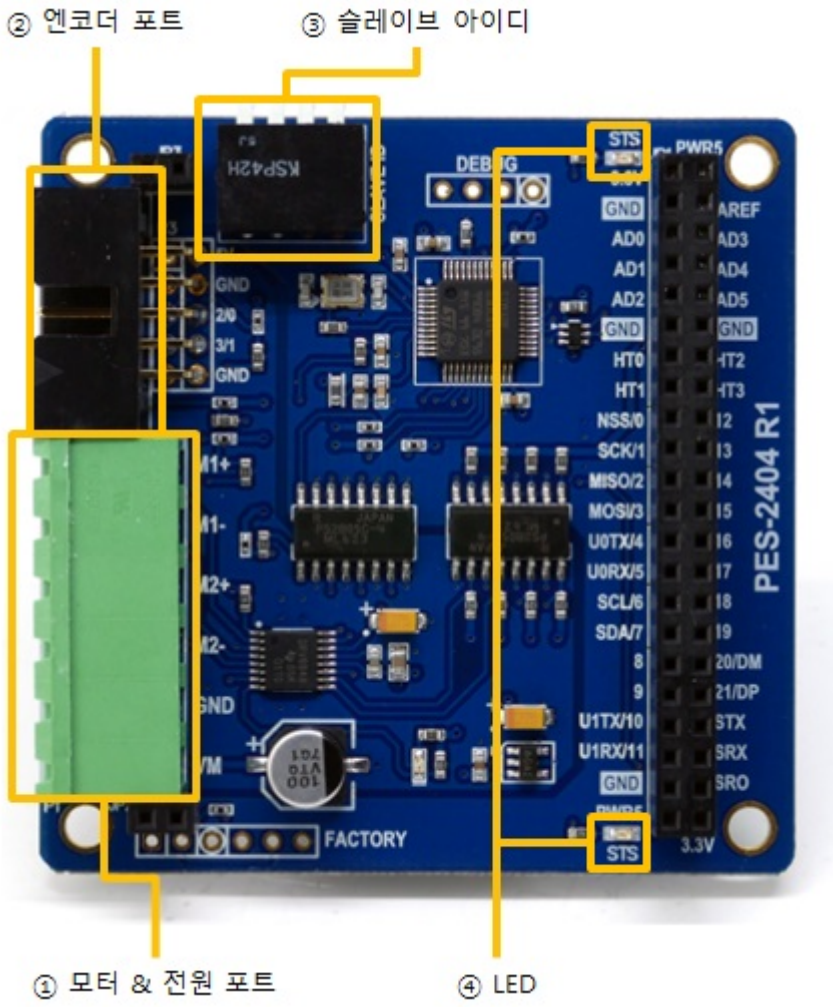
※ 치수(단위 : mm)는 제품 상태 및 재는 각도 등에 따라 약간의 오차가 있을 수 있습니다.

회로도

PES-2404의 회로도 입니다.

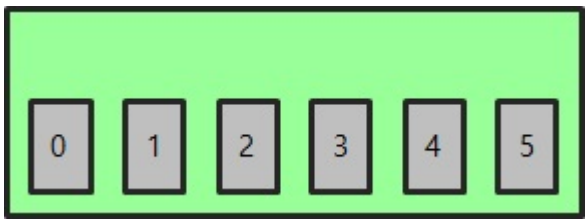
- PES-2404-R2-PO.pdf

레이아웃


② 엔코더 포트
③ 슬레이브 아이디
① 모터 & 전원 포트
④ LED
PES-2404 R1
FACTORY
STS
PWR5
3.3V

1. 모터 & 전원포트

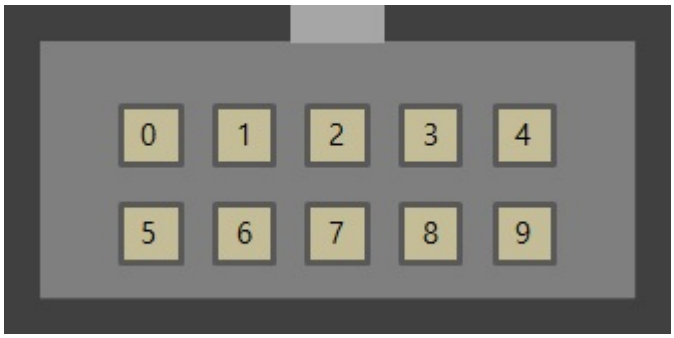
모터 & 전원포트는 모터와 모터 구동을 위한 전원을 연결하는 포트입니다. 이 포트는 5mm간격의 1 by 6 터미널 블록으로 되어 있습니다.



번호	이름	방향	설명
0	M1+	출력	모터 연결포트 1, 양극(+)
1	M1-	출력	모터 연결포트 1, 음극(-)
2	M2+	출력	모터 연결포트 2, 양극(+)
3	M2-	출력	모터 연결포트 2, 음극(-)
4	GND	-	모터 구동용 전원 (Ground)
5	VM	입력	모터 구동용 전원 (DC 4 ~ 18V)

2. 엔코더 연결포트

이 포트는 엔코더를 연결하는 포트입니다.

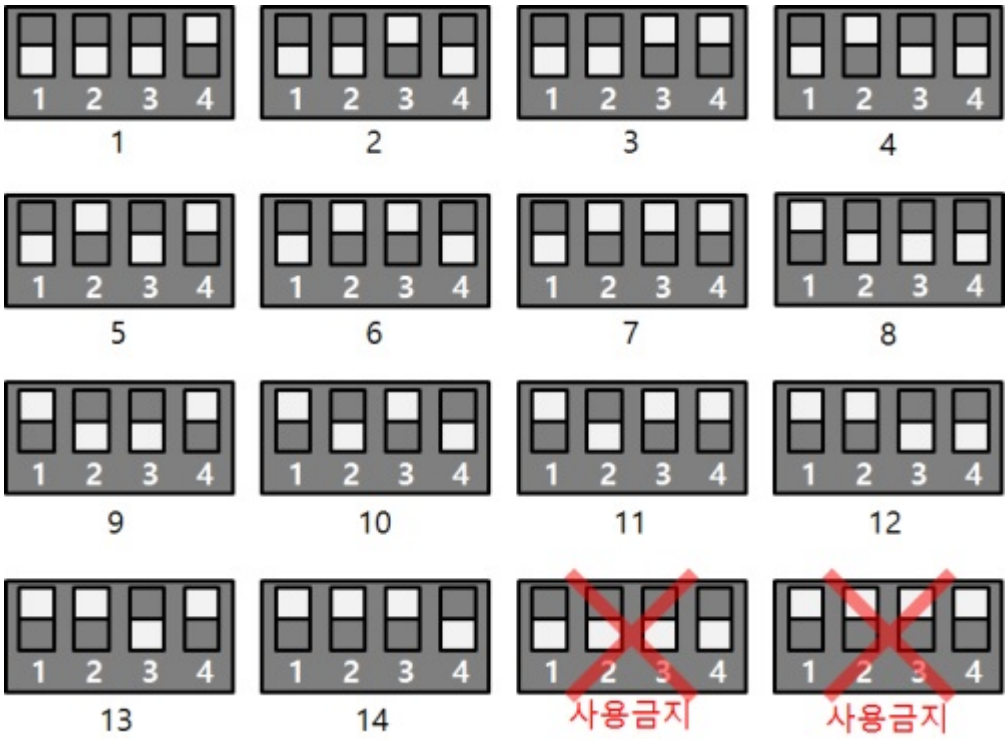


번호	이름	방향	설명
0	5V	출력	엔코더 전원 (DC 5V)
1	GND	-	엔코더 전원 (Ground)
2	1A	입력	포트 1번의 엔코더 A상
3	1B	입력	포트 1번의 엔코더 B상
4	GND	-	엔코더 전원 (Ground)
5	5V	출력	엔코더 전원 (DC 5V)
6	GND	-	엔코더 전원 (Ground)
7	2A	입력	포트 2번의 엔코더 A상
8	2B	입력	포트 2번의 엔코더 B상
9	GND	-	엔코더 전원 (Ground)

※ 1A, 1B, 2A 및 2B는 내부적으로 풀업(pull-up)되어 있습니다.

3. 슬레이브 아이디 스위치

슬레이브 아이디는 마스터인 PHPoC 보드가 스마트 확장보드 각각을 구분하는데 사용됩니다. 하나의 PHPoC 보드에 연결되는 각 스마트 확장보드는 고유한 슬레이브 아이디를 사용해야 합니다.
슬레이브 아이디는 1부터 14까지 14개 중 하나로 설정할 수 있으며 다음과 같이 4개의 딥스위치를 조정하여 설정합니다.



4. LED

PES-2404 보드에는 2개의 STS LED가 있습니다. JP1 상단의 STS LED는 3.3V에, 하단의 STS LED는 5V와 연결되어 있으며 동작은 다음과 같습니다.

상태	LED 동작
정상	1초마다 켜짐/꺼짐 반복
슬레이브 아이디 설정 이상	매우 빠르게 깜박임
PHPoC보드와 통신 불가능	꺼짐

사용하기

PES-2404를 사용하는 방법은 다음과 같습니다.

1. PHPoC 보드에 연결

PES-2404는 단독으로 사용할 수 없습니다. 반드시 PHPoC 보드에 연결하여 사용하시기 바랍니다.

2. 소프트웨어(IDE) 설치

PHPoC 디버거는 PHPoC 제품의 설정 및 개발에 사용되는 소프트웨어 입니다. PES-2404는 PHPoC 보드형 제품을 통해 제어할 수 있으므로 이 보드를 사용하기 위해서는 PC에 PHPoC 디버거를 반드시 설치해야 합니다.

- PHPoC 디버거 다운로드 페이지
- PHPoC 디버거 매뉴얼 페이지

3. SPC라이브러리 및 예제코드 활용

SPC라이브러리는 PES-2404를 비롯한 스마트 확장보드 라이브러리 입니다. 이 라이브러리를 사용하면 비교적 간단하게 PES-2404를 사용할 수 있습니다. 라이브러리와 함수에 대한 자세한 내용은 다음 문서를 참조하시기 바랍니다.

- SPC라이브러리 매뉴얼 페이지

함수와 명령어

스마트 확장보드를 사용하기 위해서는 spc_request_dev함수 또는 spc_request_sys함수가 필요합니다.

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

- $sid: 보드에 설정된 슬레이브 아이디
- $cmd: 보드에 전송할 명령어 문자열

스마트 확장보드 공통 명령어

모든 스마트 확장보드가 공통으로 지원하는 명령어는 spc_request_sys함수를 사용하며, 명령어 목록은 다음과 같습니다.

명령어	인수	설명
get	did	디바이스 아이디 확인
get	uid	유니크 아이디 확인

PES-2404 명령어

스마트 확장보드별로 적용되는 명령어는 spc_request_dev함수를 사용합니다. PES-2404의 설정 및 제어 명령은 다음 3가지로 분류됩니다.

- 모터 제어 및 PWM 설정
- 엔코더 설정 및 모니터링
- 저역통과필터 설정 및 모니터링

PES-2404의 명령어 목록

명령어	인수1	인수2	인수3
pwm	set	pol	(+ 또는 -)
		dir	(+ 또는 -)
		period	(1 ~ 1000000)
		width	(1 ~ 1000000)
		decay	(fast 또는 slow)
enc	set	pol	(+, - 또는 0)
		pos	(-1000000000 ~ +1000000000)
		psr	(1 ~ 64)
	get	pos	-
		period	-
lpf	set	freq	n
		pnc	n
	get	pnc	-

위 명령어들을 이용해 명령어 문자열을 만들 때 해당 문자열의 맨 앞에서 DC모터의 포트번호를 반드시 지정해야 합니다. 포트 1번은 dc1을, 포트 2번은 dc2를 사용합니다. 다음은 명령어 문자열의 예입니다.

```
"dc1 pwm set pol +"
```

```
"dc2 enc set pos 500"
```

모터 제어 및 PWM 설정

PWM 설정 및 제어 명령어는 pwm입니다.
관련된 명령어로는 극성 설정, 회전방향 설정, PWM주기 및 유효시간 설정 등이 있습니다.

PWM 극성 설정

PWM 극성을 설정하는 명령어는 set pol입니다.

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

polarity에 극성(+ 또는 -)을 지정합니다. 기본 값은 +이며 -로 설정하는 경우 극성이 반대로 출력됩니다.

- PWM 극성 설정 예

```
spc_request_dev($sid, "dc1 pwm set pol +"); // 정상 극성
spc_request_dev($sid, "dc1 pwm set pol -"); // 반대 극성
```

회전 방향 설정

회전 방향을 설정하는 명령어는 set dir 입니다.

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

- 회전 방향 설정 예

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

direction에 회전 방향(+ 또는 -)을 지정합니다. 기본 값은 +이며 -로 설정하는 경우 회전방향이 반대가 됩니다.
회전 방향은 set pol과 set dir에 모두 영향을 받습니다.

set pol의 값	set dir의 값	회전 방향
+	+	시계방향
+	-	반시계방향
-	+	반시계방향
-	-	시계방향

PWM 주기 설정

PWM 주기를 설정하는 명령어는 set period입니다.

> "dc1 pwm set period (period_us)"

period_us에 주기를 지정합니다. 설정 단위는 마이크로초 입니다.

- PWM주기 설정 예

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

모터 제어 예

모터 제어를 위한 유효시간을 설정하는 명령어는 set width입니다.

유효시간은 PWM신호의 한 주기 내에서 HIGH신호가 출력되는 시간을 의미합니다. 유효시간을 설정하면 이에 따라서 PWM 신호의 듀티사이클이 결정됩니다.

> 듀티사이클(%) = 유효시간 / 주기 * 100

또한 이 설정과 동시에 PWM출력이 시작되므로 이 명령은 모터를 구동시키는 역할을 합니다.

- 모터 제어(PWM 유효시간 설정) 예

```
<?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);
```

```
}
?>
```

감쇠모드 설정

감쇠모드를 설정하는 명령어는 set decay입니다.

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

mode에 감쇠모드를 지정합니다.

인자값	감쇠모드
fast	빠른 감쇠
slow	느린 감쇠

- 감쇠모드 설정 예

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

엔코더 설정 및 모니터링

엔코더 설정 및 모니터링 명령어는 enc입니다.
관련된 명령어로는 카운트 방향 설정, 카운터값 설정, 카운터값 모니터링 및 주기 모니터링 등이 있습니다.

엔코더 카운트 방향 설정

엔코더 카운트 방향을 설정하는 명령어는 set pol입니다.

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

polarity에 카운트 방향(+ 또는 -)을 지정하며 기본 값은 +입니다.

센서가 하나인 인코더의 경우에는 polarity에 0을 설정합니다. 이 경우 pos의 값은 항상 증가합니다.

카운트 방향	모터 회전방향	카운터값
+	정방향	증가
+	역방향	감소
-	정방향	감소
-	역방향	증가
0	정방향	증가
0	역방향	증가

- 엔코더 카운트 방향 설정 예

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

엔코더 카운터값 설정

엔코더 카운터값을 설정하는 명령어는 set pos입니다.

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

value에 카운터값을 지정합니다. 카운터값은 -1000000000(-10억) 부터 1000000000(10억)까지 설정할 수 있습니다.

- 엔코더 카운터값 설정 예

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

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

엔코더 샘플링 카운트 설정

엔코더 샘플링 카운트를 설정하는 명령어는 set psr입니다.

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

value에 샘플링 카운트를 지정합니다.
엔코더 샘플링 카운트는 엔코더 출력 펄스의 주기를 측정할 때 사용할 펄스의 개수를 의미합니다. 샘플링 카운트가 많을수록 오차는 줄어듭니다. 엔코더 샘플링 카운트는 1부터 64까지 설정할 수 있습니다.

- 엔코더 샘플링 카운트 설정 예

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

엔코더 카운터값 모니터링

엔코더 카운터값 모니터링을 위한 명령어는 get pos입니다.

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

- 엔코더 카운터값 모니터링 예

```
<?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";
}
?>
```

엔코더 출력 펄스 주기 모니터링

엔코더 출력 펄스 주기 모니터링을 위한 명령어는 get period입니다.

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

엔코더 출력 펄스 주기의 단위는 마이크로초(us) 입니다. 엔코더 출력 펄스 주기를 모니터링할 때 오차를 줄이기 위해서는 엔코더 샘플링 카운트를 높은 값으로 설정하시기 바랍니다.

- 엔코더 출력 펄스 주기 모니터링 예

```
<?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";
}
?>
```

저역통과필터 설정 및 모니터링

엔코더 모니터링을 위한 저역통과필터(low-pass filter) 설정 및 모니터링 명령어는 "lpf"입니다.
관련된 명령어로는 컷오프 주파수 설정, 노이즈 카운터 설정 및 노이즈 카운터 모니터링이 있습니다.

저역통과필터 컷오프 주파수 설정

저역통과필터 컷오프 주파수를 설정하는 명령어는 set freq입니다.

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

frequency에 컷오프 주파수를 지정합니다.

저역통과필터 컷오프 주파수를 설정하면 엔코더 모니터링 시 해당 주파수보다 높은 주파수에 해당하는 신호는 카운트하지 않습니다.

- 저역통과필터 컷오프 주파수 설정 예

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

저역통과필터 노이즈 카운터값 설정

저역통과필터 노이즈 카운터값을 설정하는 명령어는 set pnc입니다.

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

value에 노이즈 카운터값을 지정합니다.

- 저역통과필터 노이즈 카운터값 설정 예

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

저역통과필터 노이즈 카운터값 모니터링

저역통과필터 노이즈 카운터값 모니터링을 위한 명령어는 get pnc입니다.

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

- 저역통과필터 노이즈 카운터값 모니터링 예

```
<?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);
}
?>
```