

# PSP Library Reference

---

PSP에서 제공하는 라이브러리에 대한 설명서 입니다.

## 라이브러리 목록

- [sd\\_101.php](#) : PBH-101 기본 라이브러리
- [sd\\_104.php](#) : PBH-104 기본 라이브러리
- [sd\\_204.php](#) : PBH-204 기본 라이브러리
- [sd\\_340.php](#) : P4S-341/P4S-342 기본 라이브러리
- [sd\\_spc.php](#) : 스마트 확장보드 기본 라이브러리
- [sn\\_dns.php](#) : DNS 라이브러리
- [sn\\_smtp.php](#) : 이메일(SMTP) 라이브러리
- [sn\\_esmtp.php](#) : 확장 이메일(ESMTP) 라이브러리
- [sn\\_tcp\\_ac.php](#) : TCP 통신 라이브러리
- [sn\\_tcp\\_ws.php](#) : 웹 소켓 라이브러리
- [sn\\_mysql.php](#) : MySQL 라이브러리

## PSP 소스코드 다운로드

- [다운로드 페이지 바로가기](#)

# sd\_101.php

---

sd\_101.php는 PBH-101의 기본 라이브러리입니다.

## 정의된 함수

- led\_setup()
- led\_out()
- led\_in()
- uart\_setup()
- uart\_read()
- uart\_readn()
- uart\_write()
- uart\_txfree()
- st\_ioctl()
- st\_free\_setup()
- st\_free\_get\_count()

## 정의된 상수

### LED 포트

- LED\_A
- LED\_B
- LED\_C
- LED\_D
- LED\_E
- LED\_F
- LED\_G
- LED\_H

### LED 유형

- LOW
- HIGH
- TOGGLE

# sd\_104.php

---

sd\_104.php는 PBH-104의 기본 라이브러리입니다.

## 정의된 함수

- led\_setup()
- led\_out()
- led\_in()
- uart\_setup()
- uart\_read()
- uart\_readn()
- uart\_write()
- uart\_txfree()
- st\_ioctl()
- st\_free\_setup()
- st\_free\_get\_count()

## 정의된 상수

### LED 포트

- LED\_A
- LED\_B
- LED\_C
- LED\_D
- LED\_E
- LED\_F
- LED\_G
- LED\_H

### LED 유형

- LOW
- HIGH
- TOGGLE

# sd\_204.php

---

sd\_204.php는 PBH-204의 기본 라이브러리입니다.

## 정의된 함수

- led\_setup()
- led\_out()
- led\_in()
- dio\_out()
- dio\_in()
- uart\_setup()
- uart\_read()
- uart\_readn()
- uart\_write()
- uart\_txfree()
- st\_ioctl()
- st\_free\_setup()
- st\_free\_get\_count()

## 정의된 상수

### LED 포트

- LED\_A
- LED\_B
- LED\_C
- LED\_D
- LED\_E
- LED\_F
- LED\_G
- LED\_H

### I/O 포트

- DO\_0
- DO\_1
- DO\_2
- DO\_3
- DI\_0
- DI\_1
- DI\_2
- DI\_3

### LED 및 I/O 출력 유형

- LOW
- HIGH
- TOGGLE

# sd\_340.php

---

sd\_340.php는 P4S-341 / P4S-342의 기본 라이브러리입니다.

## 정의된 함수

- uio\_ioctl()
- uio\_setup()
- uio\_out()
- uio\_in()
- adc\_setup()
- adc\_in()
- uart\_setup()
- uart\_read()
- uart\_readn()
- uart\_write()
- uart\_txfree()
- spi\_setup()
- spi\_ioctl()
- spi\_read()
- spi\_write()
- spi\_write\_read()
- i2c\_setup()
- i2c\_ioctl()
- i2c\_scan()
- i2c\_read()
- i2c\_write()
- i2c\_write\_read()
- ht\_ioctl()
- ht\_pwm\_setup()
- ht\_pwm\_width()
- st\_ioctl()
- st\_free\_setup()
- st\_free\_get\_count()
- st\_pwm\_setup()
- st\_pwm\_width()
- um\_read()
- um\_write()
- nm\_read()
- nm\_write()

## 정의된 상수

- LOW
- HIGH
- TOGGLE

---

# sd\_spc.php

---

sd\_spc.php는 PHPoC 스마트 확장보드 제어용 라이브러리입니다.

## 정의된 함수

- [spc\\_reset\(\)](#)
- [spc\\_sync\\_baud\(\)](#)
- [spc\\_scan\(\)](#)
- [spc\\_ioctl\(\)](#)
- [spc\\_request\(\)](#)
- [spc\\_request\\_sys\(\)](#)
- [spc\\_request\\_dev\(\)](#)

---

# sn\_dns.php

---

sn\_dns.php는 DNS (Domain Name System) 라이브러리입니다.

## 정의된 함수

- [dns\\_setup\(\)](#)
- [dns\\_send\\_query\(\)](#)
- [dns\\_loop\(\)](#)
- [dns\\_lookup\(\)](#)

## 정의된 상수

- [RR\\_A](#)
- [RR\\_NS](#)
- [RR\\_MX](#)

---

# sn\_smtp.php

---

sn\_smtp.php는 이메일(Simple Mail Transfer Protocol) 라이브러리입니다.

## 라이브러리 요구사항

- [DNS 라이브러리](#)

## 정의된 함수

- [smtp\\_setup\(\)](#)
- [smtp\\_hostname\(\)](#)
- [smtp\\_account\(\)](#)
- [smtp\\_loop\(\)](#)
- [smtp\\_start\(\)](#)
- [smtp\\_send\(\)](#)



---

# sn\_esmtp.php

---

sn\_esmtp.php는 확장 이메일(Extended Simple Mail Transfer Protocol) 라이브러리입니다.

## 라이브러리 요구사항

- [DNS 라이브러리](#)

## 정의된 함수

- [esmtp\\_setup\(\)](#)
- [esmtp\\_hostname\(\)](#)
- [esmtp\\_account\(\)](#)
- [esmtp\\_auth\(\)](#)
- [esmtp\\_msa\(\)](#)
- [esmtp\\_loop\(\)](#)
- [esmtp\\_start\(\)](#)
- [esmtp\\_send\(\)](#)

---

# sn\_tcp\_ac.php

---

sn\_tcp\_ac.php는 TCP 통신 라이브러리입니다.

## 정의된 함수

- tcp\_client()
- tcp\_server()
- tcp\_read()
- tcp\_readn()
- tcp\_write()
- tcp\_txfree()
- tcp\_state()

---

# sn\_tcp\_ws.php

---

sn\_tcp\_ws.php는 웹 소켓 라이브러리입니다.

## 정의된 함수

- `ws_setup()`
- `ws_read()`
- `ws_readn()`
- `ws_write()`
- `ws_txfree()`
- `ws_state()`

---

# sn\_mysql.php

---

sn\_mysql.php는 MySQL 라이브러리입니다.

## 라이브러리 요구사항

- DNS 라이브러리

## 정의된 함수

- mysql\_close()
- mysql\_connect()
- mysql\_query()
- mysql\_select\_db()
- mysql\_error()
- mysql\_errno()
- mysql\_affected\_rows()
- mysql\_num\_rows()
- mysql\_result()
- mysql\_fetch\_row()
- mysql\_setup()
- mysql\_ping()

# led\_setup()

---

지정된 LED 포트의 유형을 설정

## Description

```
void led_setup(int $pin, string $mode)
```

## Parameters

- \$pin: 유형을 설정할 LED 포트의 번호
- \$mode: LED 포트의 유형

## Return Value

없음

## Example

```
<?php
include "/lib/sd_101.php";
led_setup(LED_A, "led_net0_link"); // LED_A를 네트워크 ACT LED로 설정
led_setup(LED_B, "led_net0_rx"); // LED_B를 네트워크 수신 LED로 설정
?>
```

## See also

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)
- [led\\_out\(\)](#)
- [led\\_in\(\)](#)

# led\_out()

---

지정된 LED 포트에 지정된 논리를 출력

## Description

```
void led_out(int $pin, int $type)
```

## Parameters

- \$pin: 유형을 설정할 LED 포트의 번호
- \$type: 출력할 논리(LOW, HIGH 또는 TOGGLE)

## Return Value

없음

## Example

```
<?php
include "/lib/sd_101.php";
led_setup(LED_A, "out"); // LED_A 출력으로 설정
led_out(LED_A, LOW);    // LED_A에 LOW를 출력(ON)
?>
```

## See also

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)
- [led\\_setup\(\)](#)
- [led\\_in\(\)](#)

# led\_in()

---

지정된 LED 포트의 상태를 반환

## Description

```
void led_in(int $pin)
```

## Parameters

- \$pin: 상태를 확인할 LED 포트의 번호

## Return Value

HIGH일 때 1, LOW일 때 0

## Example

```
<?php
include "/lib/sd_101.php";
led_setup(LED_A, "in"); // LED_A를 입력으로 설정
$state = led_in(LED_A); // LED_A의 상태를 확인
?>
```

## See also

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)
- [led\\_setup\(\)](#)
- [led\\_out\(\)](#)

# dio\_out()

---

지정된 DO 포트에 출력 수행

## Description

```
void dio_out(int $pin, int $type)
```

## Parameters

- \$pin: 출력을 수행할 출력포트 번호
- \$type: 출력 논리 (LOW, HIGH 또는 TOGGLE)

## Return Value

없음

## Example

```
<?php
include "/lib/sd_204.php";
dio_out(DO_0, HIGH); // 0번 출력포트(DO_0)에 HIGH 출력(ON)
sleep(1);
dio_out(DO_0, LOW); // 0번 출력포트에 LOW 출력(OFF)
?>
```

## See also

- [sd\\_204.php](#)
- [dio\\_in\(\)](#)
- [LOW](#)
- [HIGH](#)
- [TOGGLE](#)
- [DO\\_0](#)
- [DO\\_1](#)
- [DO\\_2](#)
- [DO\\_3](#)



# dio\_in()

---

지정된 DI 포트의 입력값을 읽음

## Description

```
int dio_in(int $pin)
```

## Parameters

- \$pin: 입력값을 읽을 입력포트 번호

## Return Values

입력포트의 상태가 HIGH일 때 1, LOW일 때 0

## Example

```
<?php
include "/lib/sd_204.php";
echo dio_in(DI_0), "WrWn"; // 0번 입력포트(DI_0)의 값을 읽어서 출력
echo dio_in(DI_1), "WrWn"; // 1번 입력포트(DI_1)의 값을 읽어서 출력
?>
```

## See also

- [sd\\_204.php](#)
- [dio\\_out\(\)](#)
- [DI\\_0](#)
- [DI\\_1](#)
- [DI\\_2](#)
- [DI\\_3](#)

# uio\_ioctl()

지정된 UIO의 설정 및 사용

## Description

int/string uio\_ioctl(int \$uio\_id, string \$cmd)

## Parameters

- \$uio\_id: 설정 및 사용할 UIO의 아이디
- \$cmd: 설정 및 사용을 위한 명령어 세트

\$cmd에 사용 가능한 명령어의 종류 및 사용법은 [PHPoC Device Programming Guide for P40](#)을 참조하시기 바랍니다.

## Return Value

해당 명령어에 대한 응답 값

## Example

```
<?php
include "/lib/sd_340.php";
uio_ioctl(0, "set 0 mode out"); // UIO0의 0번 핀을 출력포트로 설정
uio_ioctl(0, "set 1-3 mode in"); // UIO0의 1, 2, 3번 핀을 입력포트로 설정
$state_0 = uio_ioctl(0, "get 0 output"); // UIO0의 1번 핀의 출력상태를 확인
$state_1 = uio_ioctl(0, "get 1 input"); // UIO0의 2번 핀의 입력상태를 확인
$state_2 = uio_ioctl(0, "get 2 input"); // UIO0의 3번 핀의 입력상태를 확인
$state_3 = uio_ioctl(0, "get 3 input"); // UIO0의 4번 핀의 입력상태를 확인
?>
```

## See also

- [sd\\_340.php](#)
- [uio\\_setup\(\)](#)
- [uio\\_out\(\)](#)
- [uio\\_in\(\)](#)

# uio\_setup()

지정된 UIO 핀의 유형 설정

## Description

```
void uio_setup(int $uio_id, int $pin, string $mode)
```

## Parameters

- \$uio\_id: 유형을 설정할 UIO 아이디
- \$pin: 유형을 설정할 핀의 번호
- \$mode: 설정할 유형

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
uio_setup(0, 0, "out"); // UIO0의 0번 핀을 출력포트로 설정
uio_setup(0, 1, "out_pp"); // UIO0의 1번 핀을 푸쉬 풀 방식의 출력포트로 설정
uio_setup(0, 2, "in"); // UIO0의 2번 핀을 입력포트로 설정
uio_setup(0, 3, "in_pu"); // UIO0의 3번 핀을 풀업 입력포트로 설정
?>
```

## See also

- [sd\\_340.php](#)
- [uio\\_ioctl\(\)](#)
- [uio\\_out\(\)](#)
- [uio\\_in\(\)](#)

# uio\_out()

---

지정된 UIO 핀의 출력 수행

## Description

```
void uio_out(int $uio_id, int $pin, int $type)
```

## Parameters

- \$uio\_id: 출력을 수행할 UIO 디바이스 아이디
- \$pin: 출력을 수행할 핀의 번호
- \$type: 출력 논리 (LOW, HIGH 또는 TOGGLE)

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
uio_setup(0, 0, "out"); // UIO0의 0번 핀을 출력포트로 설정
uio_out(0, 0, HIGH); // UIO0의 0번 핀에 HIGH 출력
sleep(1);
uio_out(0, 0, LOW); // UIO0의 0번 핀에 LOW 출력
?>
```

## See also

- [sd\\_340.php](#)
- [uio\\_ioctl\(\)](#)
- [uio\\_setup\(\)](#)
- [uio\\_in\(\)](#)
- [LOW](#)
- [HIGH](#)
- [TOGGLE](#)

# uio\_in()

---

지정된 UIO 핀의 입력값을 읽음

## Description

```
int uio_in(int $uio_id, int $pin)
```

## Parameters

- \$uio\_id: 입력값을 읽을 핀의 UIO 디바이스 번호
- \$pin: 입력값을 읽을 핀 번호

## Return Value

입력포트의 상태가 HIGH일 때 1, LOW일 때 0

## Example

```
<?php
include "/lib/sd_340.php";
uio_setup(0, 0, "in_pd"); // UIO0의 0번 핀을 입력포트(풀업)로 설정
echo uio_in(0, 0);       // UIO0의 0번 핀의 값을 읽어서 출력
?>
```

## See also

- [sd\\_340.php](#)
- [uio\\_ioctl\(\)](#)
- [uio\\_setup\(\)](#)
- [uio\\_out\(\)](#)

# uart\_setup()

지정된 UART를 설정

## Description

```
void uart_setup(int $uart_id, int $baud [, string $ctrl = "N81N"])
```

## Parameters

- \$uart\_id: 설정할 UART 아이디
- \$baud: 통신속도(단위: bps)
- \$ctrl: 4바이트 설정. 순서대로 패리티, 데이터 비트, 정지 비트 및 흐름 제어. 기본값은 "N81N".

| 구분    | \$ctrl 사용 가능한 값   |
|-------|---|
| 패리티   | N: 없음, E: Even, O: Odd, M: Mark, S: Space               |
| 데이터비트 | 8 또는 7  |
| 정지비트  | 1 또는 2  |
| 흐름제어  | N: 없음, H: RTS/CTS 사용, S: Xon/Xoff 사용, T: TxDE(RS485) 사용 |

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
uart_setup(0, 9600, "E71N"); // 0번 UART를 9600 bps, Even 패리티, 7 데이터 비트로 설정
uart_setup(1, 115200); // 1번 UART를 115200 bps로 설정
?>
```

## See also

- [sd\\_340.php](#)
- [uart\\_read\(\)](#)
- [uart\\_readn\(\)](#)
- [uart\\_write\(\)](#)
- [uart\\_txfree\(\)](#)

# uart\_read()

지정된 UART의 수신 데이터를 읽음

## Description

```
int uart_read(int $uart_id, int/string &$rbuf [, int $rlen = MAX_STRING_LEN])
```

## Parameters

- \$uart\_id: 데이터를 수신할 UART 아이디
- \$rbuf: 데이터를 읽어서 저장할 변수
- \$rlen: 읽을 데이터 길이, 생략하면 최대 읽을 수 있는 크기만큼 읽음

## Return Value

읽은 데이터 길이(바이트 수)

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$rlen = 0;
uart_setup(0, 9600); // 0번 UART를 9600 bps로 설정
while(1)
{
    uart_read(0, $rbuf); // 0번 UART로 수신된 데이터를 읽음
    echo "$rbuf\r\n";
}
?>
```

## See also

- [sd\\_340.php](#)
- [uart\\_setup\(\)](#)
- [uart\\_readn\(\)](#)
- [uart\\_write\(\)](#)
- [uart\\_txfree\(\)](#)

# uart\_readn()

지정된 UART의 수신 데이터를 지정된 길이만큼 읽음

## Description

```
int uart_readn(int $uart_id, int/string &$rbuf, int $rlen)
```

## Parameters

- \$uart\_id: 데이터를 수신할 UART 아이디
- \$rbuf: 데이터를 읽어서 저장할 변수
- \$rlen: 읽을 데이터 길이

## Return Value

성공 시 읽은 데이터 길이(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
uart_setup(0, 9600); // 0번 UART를 9600 bps로 설정
while(1)
{
    $rlen = uart_readn(0, $rbuf, 5); // 0번 UART로 수신된 데이터를 읽음(5 바이트)
    if($rlen)
    {
        echo "$rbufWrWn";
        break;
    }
}
?>
```

## See also

- [sd\\_340.php](#)
- [uart\\_setup\(\)](#)
- [uart\\_read\(\)](#)
- [uart\\_write\(\)](#)
- [uart\\_txfree\(\)](#)



# uart\_write()

---

지정된 UART로 데이터를 전송

## Description

```
int uart_write(int $uart_id, int/string $wbuf [, int $wlen = MAX_STRING_LEN])
```

## Parameters

- \$uart\_id: 데이터를 수신할 UART 아이디
- \$wbuf: 전송할 데이터가 저장된 변수
- \$wlen: 전송할 데이터 길이

## Return Value

성공 시 전송한 데이터 길이(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
uart_setup(0, 9600); // 0번 UART를 9600 bps로 설정
$slen = uart_write(0, $wbuf); // 0번 UART로 데이터 전송
usleep(500000);
echo "$slen bytes has been sentWrWn";
?>
```

## See also

- [sd\\_340.php](#)
- [uart\\_setup\(\)](#)
- [uart\\_read\(\)](#)
- [uart\\_readn\(\)](#)
- [uart\\_txfree\(\)](#)

# uart\_txfree()

지정된 UART의 송신 버퍼 여유 공간 확인

## Description

```
int uart_txfree(int $uart_id)
```

## Parameters

- \$uart\_id: 송신 버퍼 여유 공간을 확인할 UART 아이디

## Return Value

성공 시 송신 버퍼 여유 공간(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
uart_setup(0, 9600);    // 0번 UART를 9600 bps로 설정
$tx_free = uart_txfree(0); // 송신 버퍼의 여유 공간 확인
if($tx_free >= 5)      // 송신 버퍼의 여유 공간이 5보다 크면
    uart_write(0, $wbuf); // 5 바이트 데이터 전송
?>
```

## See also

- [sd\\_340.php](#)
- [uart\\_setup\(\)](#)
- [uart\\_read\(\)](#)
- [uart\\_readn\(\)](#)
- [uart\\_write\(\)](#)

# spi\_setup()

SPI 통신 설정

## Description

```
void spi_setup(int $spi_id[, int $div = 256[, int $mode = 3]])
```

## Parameters

- \$spi\_id: SPI 아이디
- \$div: 분주 비(2, 4, 8, 16, 32, 64, 128 또는 256)

| 분주 비 | 통신 속도(bps) |
|------|------------|
| 2    | 21M        |
| 4    | 10.5M      |
| 8    | 5.25M      |
| 16   | 약 2.6M     |
| 32   | 약 1.3M     |
| 64   | 약 656K     |
| 128  | 약 328K     |
| 256  | 약 164K     |

- \$mode: SPI 모드 (0 ~ 3)

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
spi_setup(0);          // SPI를 분주 비 256, 모드 3으로 설정
spi_setup(0, 128);    // SPI를 분주 비 128, 모드 3으로 설정
spi_setup(0, 128, 0); // SPI를 분주 비 128, 모드 0으로 설정
?>
```

## See also

- [sd\\_340.php](#)
- [spi\\_ioctl\(\)](#)
- [spi\\_read\(\)](#)
- [spi\\_write\(\)](#)
- [spi\\_write\\_read\(\)](#)

# spi\_ioctl()

지정 된 SPI의 설정 및 사용

## Description

int/string spi\_ioctl(int \$spi\_id, string \$cmd)

## Parameters

- \$spi\_id: SPI 아이디
- \$cmd: 설정 및 사용을 위한 명령어 세트

\$cmd에 사용 가능한 명령어의 종류 및 사용법은 [PHPoC Device Programming Guide for P40](#)을 참조하시기 바랍니다.

## Return Value

해당 명령어에 대한 응답 값

## Example

```
<?php
include "/lib/sd_340.php";
spi_ioctl(0, "set mode 3); // SPI 모드 3으로 설정
spi_ioctl(0, "set div 128); // 분주 비 128로 설정
?>
```

## See also

- [sd\\_340.php](#)
- [spi\\_setup\(\)](#)
- [spi\\_read\(\)](#)
- [spi\\_write\(\)](#)
- [spi\\_write\\_read\(\)](#)

# spi\_read()

SPI의 수신 데이터를 읽음

## Description

```
int spi_read(int $spi_id, string &$rbuf, int $rlen)
```

## Parameters

- \$spi\_id: SPI 아이디
- \$rbuf: 데이터를 읽어서 저장할 변수
- \$rlen: 읽을 데이터 길이(바이트)

## Return Value

성공 시 읽은 데이터 길이(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$rlen = 0;
spi_setup(0);      // SPI 통신 설정
while(1)
{
    $rlen = spi_read(0, $rbuf, 2); // SPI로 수신된 데이터를 2바이트 읽음
    if($rlen == 2)
        echo "$rbufWrWn";
    sleep(1);
}
?>
```

## See also

- [sd\\_340.php](#)
- [spi\\_setup\(\)](#)
- [spi\\_ioctl\(\)](#)
- [spi\\_write\(\)](#)
- [spi\\_write\\_read\(\)](#)

# spi\_write()

SPI로 데이터를 전송

## Description

```
int spi_write(int $spi_id, int/string $wbuf[, int $wlen = MAX_STRING_LEN])
```

## Parameters

- \$spi\_id: SPI 아이디
- \$wbuf: 전송할 데이터가 저장된 변수
- \$wlen: 전송할 데이터 길이

## Return Value

성공 시 전송한 데이터 길이(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "\x1E\x07";
spi_setup(0); // SPI 설정
$slen = spi_write(0, $wbuf, 2); // SPI로 데이터 전송
if($slen == 2)
    echo "$slen bytes has been sent\n";
?>
```

## See also

- [sd\\_340.php](#)
- [spi\\_setup\(\)](#)
- [spi\\_ioctl\(\)](#)
- [spi\\_read\(\)](#)
- [spi\\_write\\_read\(\)](#)

# spi\_write\_read()

SPI로 데이터를 전송하고 수신 데이터를 읽음

## Description

```
int spi_write_read(int $spi_id, string $wbuf, string &$rbuf, int $rlen)
```

## Parameters

- \$spi\_id: SPI 아이디
- \$wbuf: 전송할 데이터가 저장된 변수(문자열 데이터만 가능)
- \$rbuf: 데이터를 읽어서 저장할 변수
- \$rlen: 읽을 데이터 길이

## Return Value

읽거나 쓴 데이터 길이(바이트)

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$wbuf = "\x1E\x07";
$rlen = 0;
spi_setup(0); // SPI 통신 설정
$rlen = spi_write_read(0, $wbuf, $rbuf, 2); // SPI 데이터 쓰기 및 읽기
if($rlen == 2)
    echo "$rbuf\r\n"; // 읽은 데이터 출력
?>
```

## See also

- [sd\\_340.php](#)
- [spi\\_setup\(\)](#)
- [spi\\_ioctl\(\)](#)
- [spi\\_read\(\)](#)
- [spi\\_write\(\)](#)

# i2c\_ioctl()

I2C의 설정 및 사용

## Description

int/string i2c\_ioctl(int \$i2c\_id, string \$cmd)

## Parameters

- \$i2c\_id: I2C 아이디
- \$cmd: 설정 및 사용 명령어

\$cmd에 사용 가능한 명령어의 종류 및 사용법은 [PHPoC Device Programming Guide for P40](#)을 참조하시기 바랍니다.

## Return Value

사용 명령어에 따른 반환 값

## Example

```
<?php
include "/lib/sd_340.php";
i2c_ioctl(0, "set saddr 0x0e"); // I2C 상대 슬레이브 디바이스 주소 설정
i2c_ioctl(0, "set mode sm"); // I2C 모드 설정: 표준 모드
i2c_ioctl(0, "req read 0"); // I2C 데이터 읽기
i2c_ioctl(0, "req write"); // I2C 데이터 쓰기
?>
```

## See also

- [sd\\_340.php](#)
- [i2c\\_setup\(\)](#)
- [i2c\\_scan\(\)](#)
- [i2c\\_read\(\)](#)
- [i2c\\_write\(\)](#)
- [i2c\\_write\\_read\(\)](#)



# i2c\_setup()

I2C 통신 설정

## Description

```
void i2c_setup(int $i2c_id, int $saddr[, string $mode = "sm"])
```

## Parameters

- \$i2c\_id: I2C 아이디
- \$saddr: 상대 슬레이브 디바이스 주소
- \$mode: 모드(표준 - sm, 고속 - fm)

| I2C모드 | 표기 | 속도      |
|-------|----|---------|
| 표준    | sm | 100Kbps |
| 고속    | fm | 400Kbps |

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
i2c_setup(0, 0x1e); // I2C 슬레이브 디바이스 주소(0x0E) 설정
i2c_setup(0, 0x1e, "fm"); // I2C 슬레이브 디바이스 주소(0x0E) 및 모드(고속) 설정
?>
```

## See also

- [sd\\_340.php](#)
- [i2c\\_ioctl\(\)](#)
- [i2c\\_scan\(\)](#)
- [i2c\\_read\(\)](#)
- [i2c\\_write\(\)](#)
- [i2c\\_write\\_read\(\)](#)

# i2c\_scan()

I2C 슬레이브 검색

## Description

```
int i2c_scan(int $i2c_id[, int $rw_bit = 1[, int $len = 0]])
```

## Parameters

- \$i2c\_id: I2C 아이디
- \$rw\_bit: 읽기 혹은 쓰기 비트
- \$len: 읽기 요청 데이터 길이

## Return Value

검색 된 슬레이브 수

## Example

```
<?php
include "/lib/sd_340.php";
$num = i2c_scan(0);           // 슬레이브 검색
echo "$num slave(s) are founded"; // 검색 결과 출력
?>
```

## See also

- [sd\\_340.php](#)
- [i2c\\_ioctl\(\)](#)
- [i2c\\_setup\(\)](#)
- [i2c\\_read\(\)](#)
- [i2c\\_write\(\)](#)
- [i2c\\_write\\_read\(\)](#)

# i2c\_read()

I2C의 수신 데이터를 읽음

## Description

```
int i2c_read(int $i2c_id, string &$rbuf, int $rlen)
```

## Parameters

- \$i2c\_id: I2C 아이디
- \$rbuf: 데이터를 읽어서 저장할 변수
- \$rlen: 읽을 데이터 길이(바이트)

## Return Value

읽은 데이터 길이(바이트 수)

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$rlen = 0;
i2c_setup(0, 0x0E);      // I2C 통신 설정
while(1)
{
    $rlen = i2c_read(0, $rbuf, 2); // I2C로 수신된 데이터를 2바이트 읽음
    if($rlen == 2)
        echo "$rbufWrWn";      // 읽은 데이터 출력
    sleep(1);
}
?>
```

## See also

- [sd\\_340.php](#)
- [i2c\\_ioctl\(\)](#)
- [i2c\\_setup\(\)](#)
- [i2c\\_scan\(\)](#)
- [i2c\\_write\(\)](#)
- [i2c\\_write\\_read\(\)](#)

# i2c\_write()

I2C로 데이터를 전송

## Description

```
int i2c_write(int $i2c_id, int/string $wbuf[, int $wlen = MAX_STRING_LEN])
```

## Parameters

- \$i2c\_id: I2C 아이디
- \$wbuf: 전송할 데이터가 저장된 변수
- \$wlen: 전송할 데이터 길이

## Return Value

성공 시 전송한 데이터 길이(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "\x05\x03";
i2c_setup(0, 0x1e);           // I2C 설정
$slen = i2c_write(0, $wbuf, 2); // I2C로 데이터 전송
if($slen == 2)
    echo "$slen bytes have been sent\n";
?>
```

## See also

- [sd\\_340.php](#)
- [i2c\\_ioctl\(\)](#)
- [i2c\\_setup\(\)](#)
- [i2c\\_scan\(\)](#)
- [i2c\\_read\(\)](#)
- [i2c\\_write\\_read\(\)](#)

# i2c\_write\_read()

I2C로 데이터를 전송하고 수신 데이터를 읽음

## Description

```
int i2c_write_read(int $i2c_id, string $wbuf, string &$rbuf, int $rlen)
```

## Parameters

- `$i2c_id`: I2C 아이디
- `$wbuf`: 전송할 데이터가 저장된 변수(문자열 데이터만 가능)
- `$rbuf`: 데이터를 읽어서 저장할 변수
- `$rlen`: 읽을 데이터 길이

## Return Value

성공 시 읽거나 쓴 길이(바이트), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$wbuf = "\x05\x03";
$rlen = 0;
i2c_setup(0, 0x1e);           // I2C 통신 설정
$rlen = i2c_write_read(0, $wbuf, $rbuf, 2); // I2C 데이터 쓰기 및 읽기
if($rlen == 2)
    echo "$rbuf";           // 읽은 데이터 출력
?>
```

## See also

- [sd\\_340.php](#)
- [i2c\\_ioctl\(\)](#)
- [i2c\\_setup\(\)](#)
- [i2c\\_scan\(\)](#)
- [i2c\\_read\(\)](#)
- [i2c\\_write\(\)](#)

# st\_ioctl()

지정된 ST(소프트웨어 타이머)의 설정 및 사용

## Description

```
void st_ioctl(int $st_id, string $cmd)
```

## Parameters

- \$st\_id: 설정 및 사용할 ST 아이디
- \$cmd: 설정 및 사용 명령어

\$cmd에 사용 가능한 명령어의 종류 및 사용법은 [PHPoC Device Programming Guide for P40](#)을 참조하시기 바랍니다.

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
st_ioctl(0, "set div us"); // 0번 ST의 단위를 마이크로초로 설정
st_ioctl(0, "set mode free"); // 0번 ST를 프리모드(카운터모드)로 설정
st_ioctl(0, "set dir up"); // 0번 ST를 업카운터로 설정
st_ioctl(0, "start"); // 0번 ST 가동
sleep(1);
st_ioctl(0, "stop"); // 0번 ST 정지
?>
```

## See also

- [sd\\_340.php](#)
- [st\\_free\\_setup\(\)](#)
- [st\\_free\\_get\\_count\(\)](#)
- [st\\_pwm\\_setup\(\)](#)
- [st\\_pwm\\_width\(\)](#)

# st\_free\_setup()

---

지정된 ST(소프트웨어 타이머)를 프리모드로 설정하고 가동

## Description

```
void st_free_setup(int $st_id [, string $div = "ms"])
```

## Parameters

- \$st\_id: 프리모드로 사용할 ST 아이디
- \$div: 단위 설정: 초(sec), 밀리초(ms), 마이크로초(us) 중 하나, 기본값은 밀리초

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
st_free_setup(0); // 0번 ST를 프리모드로 설정하고 가동
st_free_setup(1, "sec"); // 1번 ST를 프리모드(초 단위)로 설정하고 가동
?>
```

## See also

- [sd\\_340.php](#)
- [st\\_ioctl\(\)](#)
- [st\\_free\\_get\\_count\(\)](#)
- [st\\_pwm\\_setup\(\)](#)
- [st\\_pwm\\_width\(\)](#)

# st\_free\_get\_count()

---

지정된 프리모드 ST(소프트웨어 타이머)의 카운트 값을 읽음

## Description

```
int st_free_get_count(int $st_id)
```

## Parameters

- \$st\_id: 카운트 값을 읽을 ST 아이디

## Return Value

해당 ST의 카운트 값, 단위는 설정 단위를 따름

## Example

```
<?php
include "/lib/sd_340.php";
st_free_setup(0); // 0번 ST를 프리모드로 설정하고 가동
while(1)
{
    $count = st_free_get_count(0); // 0번 ST의 카운트 값을 읽음
    echo "$count\r\n"; // 값을 출력
}
?>
```

## See also

- [sd\\_340.php](#)
- [st\\_ioctl\(\)](#)
- [st\\_free\\_setup\(\)](#)
- [st\\_pwm\\_setup\(\)](#)
- [st\\_pwm\\_width\(\)](#)



# st\_pwm\_setup()

지정된 ST(소프트웨어 타이머)의 PWM 모드로 설정 및 사용

## Description

```
void st_pwm_setup(int $st_id, int $pin, int $width, int $period [, string $div = "ms"])
```

## Parameters

- \$st\_id: PWM 모드로 사용할 ST 아이디
- \$pin: 신호를 출력할 핀 번호
- \$width: 펄스의 HIGH 구간 지속시간
- \$period: 펄스의 주기
- \$div: ST의 단위, 초("sec"), 밀리초("ms") 또는 마이크로초("us")로 설정 가능하며 기본값은 밀리초

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
// 0번 ST를 UIO0의 5번 핀으로 주기가 200밀리초이고 듀티 사이클이 50%인 PWM 모드로 설정 및 가동
st_pwm_setup(0, 5, 100, 200);
// 1번 ST를 UIO0의 9번 핀으로 주기가 100마이크로초이고 듀티 사이클이 10%인 PWM 모드로 설정 및 가동
st_pwm_setup(1, 9, 10, 100, "us");
?>
```

## See also

- [sd\\_340.php](#)
- [st\\_ioctl\(\)](#)
- [st\\_free\\_setup\(\)](#)
- [st\\_free\\_get\\_count\(\)](#)
- [st\\_pwm\\_width\(\)](#)

# st\_pwm\_width()

지정된 PWM 모드 ST(소프트웨어 타이머)의 듀티 사이클을 조정

## Description

```
void st_pwm_width(int $st_id, int $width, int $period)
```

## Parameters

- \$st\_id: PWM 모드로 사용할 ST 아이디
- \$width: 펄스의 HIGH 구간 지속시간
- \$period: 펄스의 주기

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
// 0번 ST를 주기가 100밀리초이고 듀티 사이클이 1%인 PWM 모드로 설정 및 가동
st_pwm_setup(0, 0, 1, 100);
for($i = 0; $i < 99; $i++)
{
    st_pwm_width(0, 1+$i, 100); // PWM 출력의 듀티 사이클을 1씩 증가
}
?>
```

## See also

- [sd\\_340.php](#)
- [st\\_ioctl\(\)](#)
- [st\\_free\\_setup\(\)](#)
- [st\\_free\\_get\\_count\(\)](#)
- [st\\_pwm\\_setup\(\)](#)

# ht\_ioctl()

지정된 HT(하드웨어 타이머)의 설정 및 사용

## Description

```
int ht_ioctl(int $ht_id, string $cmd)
```

## Parameters

- \$ht\_id: 설정 및 사용할 HT 아이디
- \$cmd: 설정 및 사용 명령어

\$cmd에 사용 가능한 명령어의 종류 및 사용법은 [PHPoC Device Programming Guide for P40](#) 을 참조하시기 바랍니다.

## Return Value

사용 명령어에 따른 반환 값

## Example

```
<?php
include "/lib/sd_340.php";
ht_ioctl(0, "set div us"); // 0번 HT의 단위를 마이크로초로 설정
ht_ioctl(0, "set mode output pulse"); // 0번 HT를 펄스출력모드로 설정
ht_ioctl(0, "set count 100 100"); // 펄스의 주기를 200마이크로초, 듀티 사이클을 50%로 설정
ht_ioctl(0, "set repc 2"); // 펄스의 출력횟수를 2로 설정
ht_ioctl(0, "start"); // 0번 HT 가동
sleep(1);
ht_ioctl(0, "stop"); // 0번 HT 정지
?>
```

## See also

- [sd\\_340.php](#)
- [ht\\_pwm\\_setup\(\)](#)
- [ht\\_pwm\\_width\(\)](#)

# ht\_pwm\_setup()

지정된 HT(하드웨어 타이머)의 PWM 모드로 설정 및 사용

## Description

```
void ht_pwm_setup(int $ht_id, int $width, int $period [, string $div = "us"])
```

## Parameters

- \$ht\_id: PWM 모드로 사용할 HT 아이디
- \$width: 펄스의 HIGH 구간 지속시간
- \$period: 펄스의 주기
- \$div: HT의 단위, 밀리초("ms") 또는 마이크로초("us")로 설정 가능하며 기본값은 마이크로초

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
// 0번 HT를 주기가 200마이크로초이고 듀티 사이클이 50%인 PWM 모드로 설정 및 가동
ht_pwm_setup(0, 100, 200);
// 1번 HT를 주기가 100밀리초이고 듀티 사이클이 10%인 PWM 모드로 설정 및 가동
ht_pwm_setup(1, 10, 100);
?>
```

## See also

- [sd\\_340.php](#)
- [ht\\_ioctl\(\)](#)
- [ht\\_pwm\\_width\(\)](#)

# ht\_pwm\_width()

---

지정된 PWM모드 HT(하드웨어 타이머)의 듀티 사이클을 조정

## Description

```
void ht_pwm_width(int $ht_id, int $width, int $period)
```

## Parameters

- \$ht\_id: PWM 모드 HT 아이디
- \$width: 펄스의 HIGH 구간 지속시간
- \$period: 펄스의 주기

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
// 0번 HT를 주기가 100마이크로초이고 듀티 사이클이 1%인 PWM 모드로 설정 및 가동
ht_pwm_setup(0, 1, 100);
for($i = 0; $i < 99; $i++)
{
    ht_pwm_width(0, 1+$i, 100); // PWM 출력의 듀티 사이클을 1씩 증가
}
?>
```

## See also

- [sd\\_340.php](#)
- [ht\\_ioctl\(\)](#)
- [ht\\_pwm\\_setup\(\)](#)

# adc\_setup()

---

지정된 ADC의 채널 설정

## Description

```
void adc_setup(int $adc_id, int $ch)
```

## Parameters

- \$adc\_id: 채널을 설정할 ADC 아이디
- \$ch: ADC 채널 번호(0 ~ 5)

## Return Value

없음

## Example

```
<?php
include "/lib/sd_340.php";
adc_setup(0, 0); // 0번 ADC를 0번 채널로 설정
adc_setup(1, 5); // 1번 ADC를 5번 채널로 설정
?>
```

## See also

- [sd\\_340.php](#)
- [adc\\_in\(\)](#)

# adc\_in()

---

지정된 ADC의 값을 읽음

## Description

```
int adc_in(int $adc_id [, int $sc = 1])
```

## Parameters

- \$adc\_id: 입력 값을 읽을 ADC 아이디
- \$sc: 아날로그 입력값의 평균을 계산하는 데 사용할 샘플링 카운트(수), 생략하면 기본값은 1

## Return Value

해당 ADC의 입력 값(0~4095)

## Example

```
<?php
include "/lib/sd_340.php";
$in_value = 0;
adc_setup(0, 0);      // 0번 ADC를 0번 채널로 설정
adc_setup(1, 1);     // 1번 ADC를 1번 채널로 설정
$value_0 = adc_in(0, 1); // 0번 ADC 값을 읽음
$value_1 = adc_in(1, 50); // 1번 ADC 값을 읽음(50번 샘플링한 후 평균값)
?>
```

## See also

- [sd\\_340.php](#)
- [adc\\_setup\(\)](#)

# um\_read()

---

지정된 UM(휘발성 사용자 메모리)의 데이터 읽기

## Description

```
int um_read(int $um_id, int $offset, string &$rbuf, int $rlen)
```

## Parameters

- `$um_id`: 데이터를 읽을 UM 아이디
- `$offset`: 데이터를 읽을 시작 위치
- `$rbuf`: 데이터를 읽어서 저장할 버퍼
- `$rlen`: 읽을 데이터의 크기(바이트)

## Return Value

읽은 데이터 크기(바이트)

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
um_read(0, 0, $rbuf, 10); // 0번 UM의 0번지부터 10바이트를 읽음
hexdump($rbuf); // 읽은 데이터를 출력
?>
```

## See also

- [sd\\_340.php](#)
- [um\\_write\(\)](#)



# um\_write()

지정된 UM(휘발성 사용자 메모리)에 데이터 저장

## Description

```
int um_write(int $um_id, int $offset, int/string $wbuf [, int $wlen = MAX_STRING_LEN])
```

## Parameters

- \$um\_id: 데이터를 저장할 UM 아이디
- \$offset: 데이터를 저장할 시작 위치
- \$wbuf: 저장할 데이터
- \$rlen: 저장할 데이터의 크기(바이트)

## Return Value

성공 시 저장한 데이터 크기, 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
$rbuf = "";
um_write(0, 0, $wbuf, 5); // 0번 UM의 0번지부터 5바이트를 저장
um_read(0, 0, $rbuf, 5); // 0번 UM의 0번지부터 5바이트를 읽음
hexdump($rbuf); // 읽은 데이터를 출력
?>
```

## See also

- [sd\\_340.php](#)
- [um\\_read\(\)](#)

# nm\_read()

지정된 NM(비휘발성 사용자 메모리)의 데이터 읽기

## Description

```
int nm_read(int $nm_id, int $offset, string &$rbuf, int $rlen)
```

## Parameters

- \$nm\_id: 데이터를 읽을 NM 아이디
- \$offset: 데이터를 읽을 시작 위치
- \$rbuf: 데이터를 읽어서 저장할 버퍼
- \$rlen: 읽을 데이터의 크기(바이트)

## Return Value

읽은 데이터 크기(바이트)

## Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
nm_read(0, 0, $rbuf, 10); // 0번 NM의 0번지부터 10바이트를 읽음
hexdump($rbuf); // 읽은 데이터를 출력
?>
```

## See also

- [sd\\_340.php](#)
- [nm\\_write\(\)](#)

# nm\_write()

지정된 NM(비휘발성 사용자 메모리)에 데이터 저장

## Description

```
int nm_write(int $nm_id, int $offset, int/string $wbuf [, int $wlen = MAX_STRING_LEN])
```

## Parameters

- \$nm\_id: 데이터를 저장할 NM 아이디
- \$offset: 데이터를 저장할 시작 위치
- \$wbuf: 저장할 데이터
- \$rlen: 저장할 데이터의 크기(바이트)

## Return Value

성공 시 저장한 데이터 크기(바이트), 실패 시 0

## Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
$rbuf = "";
nm_write(0, 0, $wbuf, 5); // 0번 NM의 0번지부터 5바이트를 저장
nm_read(0, 0, $rbuf, 5); // 0번 NM의 0번지부터 5바이트를 읽음
hexdump($rbuf); // 읽은 데이터를 출력
?>
```

## See also

- [sd\\_340.php](#)
- [nm\\_read\(\)](#)

# dns\_setup()

DNS 쿼리를 위한 설정

## Description

```
void dns_setup(int $udp_id [, string $server_addr = "", bool $ip6 = false])
```

## Parameters

- `$udp_id`: DNS 쿼리를 위해 사용할 UDP 아이디
- `$server_addr`: DNS 서버의 IP주소, 생략하면 시스템 환경 값의 DNS 서버주소를 사용
- `$ip6`: IPv6 활성화 또는 비활성화

## Return Value

없음

## Example

```
<?php
include "/lib/sn_dns.php";
dns_setup(0, "192.168.0.1"); // 0번 UDP를 이용, DNS 서버주소는 192.168.0.1로 설정
dns_setup(1); // 1번 UDP를 이용, DNS 서버주소는 시스템 환경 값의 DNS 서버주소 사용
dns_setup(0, "", true); // IPv6 활성화
?>
```

## See also

- [sn\\_dns.php](#)
- [dns\\_send\\_query\(\)](#)
- [dns\\_loop\(\)](#)
- [dns\\_lookup\(\)](#)

# dns\_send\_query()

지정된 도메인 이름으로 DNS 쿼리를 전송

## Description

```
int dns_send_query(string $name, int $type)
```

## Parameters

- \$name: 도메인 이름
- \$type: 도메인 유형 (RR\_A, RR\_NS 또는 RR\_MX)

## Return Value

성공 시 전송한 쿼리의 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_dns.php";
$rr = "";
$name = "www.phpoc.com";
dns_setup(0); // 0번 UDP를 이용
dns_send_query($name, RR_A); // www.phpoc.com의 IP주소를 질의
while(1)
{
    $rr = dns_loop(); // DNS 서버의 응답을 수신
    if($rr === false)
        usleep(1000);
    elseif($rr == "")
    {
        echo "$nameWrWn"; // 도메인 이름 출력
        break;
    }
    else
    {
        echo "$rrWrWn"; // DNS 서버의 응답을 출력
        break;
    }
}
?>
```

## See also

- [sn\\_dns.php](#)
- [dns\\_setup\(\)](#)
- [dns\\_loop\(\)](#)

- dns\_lookup()
- RR\_A
- RR\_NS
- RR\_MX

# dns\_loop()

DNS 쿼리에 대한 응답을 받아 결과를 반환

## Description

string dns\_loop(void)

## Parameters

없음

## Return Value

성공 시 IP주소, 실패 시 빈 문자열("") 또는 DNS 응답을 못 받았을 때 false

## Example

```
<?php
include "/lib/sn_dns.php";
$rr = "";
$name = "www.phpoc.com";
dns_setup(0); // 0번 UDP를 이용
dns_send_query($name, RR_A); // www.phpoc.com의 IP주소를 질의
while(1)
{
    $rr = dns_loop(); // DNS 서버의 응답을 수신
    if($rr === false)
        usleep(1000);
    elseif($rr == "")
    {
        echo "$nameWrWn"; // 도메인 이름 출력
        break;
    }
    else
    {
        echo "$rrWrWn"; // DNS 서버의 응답을 출력
        break;
    }
}
?>
```

## See also

- [sn\\_dns.php](#)
- [dns\\_setup\(\)](#)
- [dns\\_send\\_query\(\)](#)
- [dns\\_lookup\(\)](#)

- RR\_A
- RR\_NS
- RR\_MX



# dns\_lookup()

지정된 도메인 이름으로 DNS 쿼리를 전송하고 응답을 반환

## Description

string dns\_lookup(string \$name, int \$type)

## Parameters

- \$name: 도메인 이름
- \$type: 도메인 유형 (RR\_A, RR\_NS 또는 RR\_MX)

## Return Value

성공 시 IP주소, 실패 시 입력한 도메인 이름

## Example

```
<?php
include "/lib/sn_dns.php";
dns_setup(0); // 0번 UDP를 이용
$host_name = "www.phpoc.com";
$host_addr = dns_lookup($host_name, RR_A); // www.phpoc.com의 IP주소를 질의
if($host_addr == $host_name)
    echo "$host_name: Not FoundWrWn"; // 실패 시 에러메시지 출력
else
    echo "$host_name: $host_addrWrWn"; // IP주소를 출력
?>
```

## See also

- [sn\\_dns.php](#)
- [dns\\_setup\(\)](#)
- [dns\\_send\\_query\(\)](#)
- [dns\\_loop\(\)](#)
- [RR\\_A](#)
- [RR\\_NS](#)
- [RR\\_MX](#)

# smtp\_setup()

SMTP로 이메일 전송을 위한 TCP 및 UDP 아이디 설정

## Description

```
void smtp_setup(int $udp_id, int $tcp_id [, string $dns_server = "", bool $ip6 = false])
```

## Parameters

- `$udp_id`: DNS 쿼리를 위해 사용할 UDP 아이디
- `$tcp_id`: 이메일을 전송하기 위한 TCP 아이디
- `$dns_server`: DNS 서버의 IP주소, 생략하면 시스템 환경 값의 DNS 서버주소를 사용
- `$ip6`: IPv6 활성화 또는 비활성화

## Return Value

없음

## Example

```
<?php
include "/lib/sn_dns.php"; // DNS 라이브러리 추가
include "/lib/sn_smtp.php"; // SMTP 라이브러리 추가
// DNS를 위해 0번 UDP, 메일 전송을 위해 1번 TCP 및 DNS 서버주소를 192.168.0.1로 설정
smtp_setup(0, 1, "192.168.0.1");

// DNS와 SMTP를 위해 각각 2번 UDP와 3번 TCP를 설정,
smtp_setup(2, 3);

// IPv6 활성화
smtp_setup(0, 1, "", true);
?>
```

## See also

- [sn\\_smtp.php](#)
- [sn\\_dns.php](#)
- [smtp\\_hostname\(\)](#)
- [smtp\\_account\(\)](#)
- [smtp\\_loop\(\)](#)
- [smtp\\_start\(\)](#)
- [smtp\\_send\(\)](#)

# smtp\_hostname()

SMTP 이메일 송신자 호스트 이름 설정

## Description

```
void smtp_hostname(string $hostname)
```

## Parameters

- \$hostname: 이메일 송신자 호스트 이름

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_smtp.php"; // SMTP 라이브러리 추가
smtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
smtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
$msg = smtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successful\r\n";
else
    echo "send mail failed\r\n";
?>
```

## See also

- [sn\\_smtp.php](#)
- [sn\\_dns.php](#)
- [smtp\\_setup\(\)](#)
- [smtp\\_account\(\)](#)
- [smtp\\_loop\(\)](#)
- [smtp\\_start\(\)](#)
- [smtp\\_send\(\)](#)

# smtp\_account()

SMTP 송신자 이메일 주소 및 이름 설정

## Description

```
void smtp_account(string $email, string $name)
```

## Parameters

- \$email: 송신자 이메일 주소
- \$name: 송신자 이름

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_smtp.php"; // SMTP 라이브러리 추가
smtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
smtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
$subject = "email test from PHPoC";
$message = "Hi PHPoCWrWnThis is PHPoC test emailWrWnGood byeWrWn";
$msg = smtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successfulWrWn";
else
    echo "send mail failedWrWn";
?>
```

## See also

- [sn\\_smtp.php](#)
- [sn\\_dns.php](#)
- [smtp\\_setup\(\)](#)
- [smtp\\_hostname\(\)](#)
- [smtp\\_loop\(\)](#)
- [smtp\\_start\(\)](#)
- [smtp\\_send\(\)](#)

# smtp\_loop()

SMTP 메일서버의 응답을 받아 SMTP 메일 전송 과정을 진행

## Description

false/string smtp\_loop(void)

## Parameters

없음

## Return Value

성공 시 서버의 응답메시지, 실패 시 빈 문자열("") 또는 응답을 못 받았을 때 false

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_smtp.php"; // SMTP 라이브러리 추가
smtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
smtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
smtp_start("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송 준비
while(1)
{
    $msg = smtp_loop(); // 서버의 응답메시지 수신
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msg\r\n"; // 서버의 응답메시지 출력
}
?>
```

## See also

- [sn\\_smtp.php](#)
- [sn\\_dns.php](#)
- [smtp\\_setup\(\)](#)
- [smtp\\_hostname\(\)](#)
- [smtp\\_account\(\)](#)
- [smtp\\_start\(\)](#)
- [smtp\\_send\(\)](#)

# smtp\_send()

SMTP로 메일을 전송

## Description

string smtp\_send(string \$rcpt\_email, string \$rcpt\_name, string \$subject, string \$body)

## Parameters

- \$rcpt\_email: 수신자 메일 주소
- \$rcpt\_name: 수신자 이름
- \$subject: 메일 제목
- \$body: 메일 내용

## Return Value

성공 시 서버의 응답메시지, 실패 시 빈 문자열("") 또는 응답을 못 받았을 때 false

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_smtp.php"; // SMTP 라이브러리 추가
smtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
smtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
$subject = "email test from PHPoC";
$message = "Hi PHPoCWrWnThis is PHPoC test emailWrWnGood byeWrWn";
$msg = smtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successfulWrWn";
else
    echo "send mail failedWrWn";
?>
```

## See also

- [sn\\_smtp.php](#)
- [sn\\_dns.php](#)
- [smtp\\_setup\(\)](#)
- [smtp\\_hostname\(\)](#)
- [smtp\\_account\(\)](#)
- [smtp\\_start\(\)](#)
- [smtp\\_loop\(\)](#)

# smtp\_start()

SMTP로 메일을 보내는 데 필요한 항목들을 설정하고 준비

## Description

```
void smtp_start(string $rcpt_email, string $rcpt_name, string $subject, string $body)
```

## Parameters

- \$rcpt\_email: 수신자 메일 주소
- \$rcpt\_name: 수신자 이름
- \$subject: 메일 제목
- \$body: 메일 내용

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_smtp.php"; // SMTP 라이브러리 추가
smtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
smtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
smtp_start("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송 준비
while(1)
{
    $msg = smtp_loop(); // 서버의 응답메시지 수신
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msg\r\n"; // 서버의 응답메시지 출력
}
?>
```

## See also

- [sn\\_smtp.php](#)
- [sn\\_dns.php](#)
- [smtp\\_setup\(\)](#)
- [smtp\\_hostname\(\)](#)

- [smtp\\_account\(\)](#)
- [smtp\\_send\(\)](#)
- [smtp\\_loop\(\)](#)



# esmtp\_setup()

ESMTP로 이메일 전송을 위한 TCP 및 UDP 아이디 설정

## Description

```
void esmtp_setup(int $udp_id, int $tcp_id [, string $dns_server = ""[, bool $ip6 = false]])
```

## Parameters

- `$udp_id`: DNS 쿼리를 위해 사용할 UDP 아이디
- `$tcp_id`: 이메일을 전송하기 위한 TCP 아이디
- `$dns_server`: DNS 서버의 IP주소, 생략하면 시스템 환경 값의 DNS 서버주소를 사용
- `$ip6`: IPv6 활성화 또는 비활성화

## Return Value

없음

## Example

```
<?php
include "/lib/sn_dns.php"; // DNS 라이브러리 추가
include "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
// DNS를 위해 0번 UDP, 메일 전송을 위해 1번 TCP 및 DNS 서버주소를 192.168.0.1로 설정
esmtp_setup(0, 1, "192.168.0.1");

// DNS와 ESMTP를 위해 각각 2번 UDP와 3번 TCP를 설정,
// DNS 서버주소는 시스템 환경 값의 DNS 서버주소 사용
esmtp_setup(2, 3);

// IPv6 활성화
esmtp_setup(0, 1, "", true);

?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_hostname\(\)](#)
- [esmtp\\_account\(\)](#)
- [esmtp\\_auth\(\)](#)
- [esmtp\\_msa\(\)](#)
- [esmtp\\_loop\(\)](#)
- [esmtp\\_start\(\)](#)
- [esmtp\\_send\(\)](#)

# esmtp\_hostname()

ESMTP 이메일 송신자 호스트 이름 설정

## Description

void esmtp\_hostname(string \$hostname)

## Parameters

- \$hostname: 이메일 송신자 호스트 이름

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
esmtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
esmtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
esmtp_auth("msa_id", "msa_password"); // 보내는 메일서버 계정의 아이디와 비밀번호 설정
esmtp_msa("smtp.msa_domain", 465); // 보내는 메일서버와 포트 번호 설정
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successfulWrWn";
else
    echo "send mail failedWrWn";
?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_setup\(\)](#)
- [esmtp\\_account\(\)](#)
- [esmtp\\_auth\(\)](#)
- [esmtp\\_msa\(\)](#)
- [esmtp\\_loop\(\)](#)
- [esmtp\\_start\(\)](#)
- [esmtp\\_send\(\)](#)

# esmtp\_account()

ESMTP 송신자 이메일 주소 및 이름 설정

## Description

```
void esmtp_account(string $email, string $name)
```

## Parameters

- \$email: 송신자 이메일 주소
- \$name: 송신자 이름

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
esmtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
esmtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
esmtp_auth("msa_id", "msa_password"); // 보내는 메일서버 계정의 아이디와 비밀번호 설정
esmtp_msa("smtp.msa_domain", 465); // 보내는 메일서버와 포트 번호 설정
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successfulWrWn";
else
    echo "send mail failedWrWn";
?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_setup\(\)](#)
- [esmtp\\_hostname\(\)](#)
- [esmtp\\_auth\(\)](#)
- [esmtp\\_msa\(\)](#)
- [esmtp\\_loop\(\)](#)
- [esmtp\\_start\(\)](#)
- [esmtp\\_send\(\)](#)

# esmtp\_auth()

보내는 메일서버 계정의 아이디와 비밀번호 설정

## Description

```
void esmtp_auth(string $auth_id, string $auth_pwd)
```

## Parameters

- \$auth\_id: 보내는 메일서버 계정의 아이디
- \$auth\_pwd: 보내는 메일서버 계정의 비밀번호

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
esmtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
esmtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
esmtp_auth("msa_id", "msa_password"); // 보내는 메일서버 계정의 아이디와 비밀번호 설정
esmtp_msa("smtp.msa_domain", 465); // 보내는 메일서버와 포트 번호 설정
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successfulWrWn";
else
    echo "send mail failedWrWn";
?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_setup\(\)](#)
- [esmtp\\_hostname\(\)](#)
- [esmtp\\_account\(\)](#)
- [esmtp\\_msa\(\)](#)
- [esmtp\\_loop\(\)](#)
- [esmtp\\_start\(\)](#)
- [esmtp\\_send\(\)](#)

# esmtp\_msa()

보내는 메일서버의 호스트 이름과 포트 번호를 설정

## Description

```
void esmtp_msa(string $msa_name, int $msa_port)
```

## Parameters

- \$msa\_name: 보내는 메일서버의 호스트 이름
- \$msa\_port: 보내는 메일서버의 통신포트

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
esmtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
esmtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
esmtp_auth("msa_id", "msa_password"); // 보내는 메일서버 계정의 아이디와 비밀번호 설정
esmtp_msa("smtp.msa_domain", 465); // 보내는 메일서버와 포트 번호 설정
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successfulWrWn";
else
    echo "send mail failedWrWn";
?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_setup\(\)](#)
- [esmtp\\_hostname\(\)](#)
- [esmtp\\_account\(\)](#)
- [esmtp\\_auth\(\)](#)
- [esmtp\\_loop\(\)](#)
- [esmtp\\_start\(\)](#)
- [esmtp\\_send\(\)](#)

# esmtp\_loop()

메일서버의 응답을 받아 ESMTP 메일 전송과정을 진행

## Description

false/string esmtp\_loop(void)

## Parameters

없음

## Return Value

성공 시 서버의 응답메시지, 실패 시 빈 문자열("") 또는 응답을 못 받았을 때 false

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
esmtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
esmtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
esmtp_auth("msa_id", "msa_password"); // 보내는 메일서버 계정의 아이디와 비밀번호 설정
esmtp_msa("smtp.msa_domain", 465); // 보내는 메일서버와 포트 번호 설정
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
esmtp_start("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송 준비
while(1)
{
    $msg = esmtp_loop(); // 서버의 응답메시지 수신
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msgWrWn"; // 서버의 응답메시지 출력
}
?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_setup\(\)](#)
- [esmtp\\_hostname\(\)](#)
- [esmtp\\_account\(\)](#)
- [esmtp\\_auth\(\)](#)

- esmtp\_msa()
- esmtp\_start()
- esmtp\_send()

# esmtp\_start()

ESMTP로 메일을 보내는 데 필요한 항목들을 설정하고 준비

## Description

```
void esmtp_start(string $rcpt_email, string $rcpt_name, string $subject, string $body)
```

## Parameters

- \$rcpt\_email: 수신자 메일 주소
- \$rcpt\_name: 수신자 이름
- \$subject: 메일 제목
- \$body: 메일 내용

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
esmtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
esmtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
esmtp_auth("msa_id", "msa_password"); // 보내는 메일서버 계정의 아이디와 비밀번호 설정
esmtp_msa("smtp.msa_domain", 465); // 보내는 메일서버와 포트 번호 설정
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
esmtp_start("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송 준비
while(1)
{
    $msg = esmtp_loop(); // 서버의 응답메시지 수신
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msgWrWn"; // 서버의 응답메시지 출력
}
?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_setup\(\)](#)



- esmtp\_hostname()
- esmtp\_account()
- esmtp\_auth()
- esmtp\_msa()
- esmtp\_loop()
- esmtp\_send()

# esmtp\_send()

ESMTP로 메일을 전송

## Description

string esmtp\_send(string \$rcpt\_email, string \$rcpt\_name, string \$subject, string \$body)

## Parameters

- \$rcpt\_email: 수신자 메일 주소
- \$rcpt\_name: 수신자 이름
- \$subject: 메일 제목
- \$body: 메일 내용

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리 추가
include_once "/lib/sn_esmtp.php"; // ESMTP 라이브러리 추가
esmtp_hostname("from_domain.com"); // 송신자 호스트 이름 설정
esmtp_account("from_id@from_domain.com", "from_name"); // 송신자 이메일 및 이름 설정
esmtp_auth("msa_id", "msa_password"); // 보내는 메일서버 계정의 아이디와 비밀번호 설정
esmtp_msa("smtp.msa_domain", 465); // 보내는 메일서버와 포트 번호 설정
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message); // 메일 전송
if($msg == "221")
    echo "send mail successfulWrWn";
else
    echo "send mail failedWrWn";
?>
```

## See also

- [sn\\_esmtp.php](#)
- [sn\\_dns.php](#)
- [esmtp\\_setup\(\)](#)
- [esmtp\\_hostname\(\)](#)
- [esmtp\\_account\(\)](#)
- [esmtp\\_auth\(\)](#)
- [esmtp\\_msa\(\)](#)
- [esmtp\\_loop\(\)](#)

- [esmtplib.start\(\)](#)

# tcp\_client()

---

클라이언트로써 TCP 접속 시도

## Description

```
void tcp_client(int $tcp_id, string $addr, int $port)
```

## Parameters

- \$tcp\_id: TCP 접속에 사용할 TCP 아이디
- \$addr: 접속할 TCP 서버의 IP주소
- \$port: 접속할 TCP 포트 번호

## Return Value

없음

## Example

```
<?php
include "/lib/sn_tcp_ac.php";
$addr = "192.168.0.100"; // 접속할 TCP 서버의 IP주소
$port = 1470; // 접속할 TCP 포트 번호
tcp_client(0, $addr, $port); // 0번 TCP로 TCP 접속 시도
sleep(15);
?>
```

## See also

- [sn\\_tcp\\_ac.php](#)
- [tcp\\_server\(\)](#)
- [tcp\\_read\(\)](#)
- [tcp\\_readn\(\)](#)
- [tcp\\_write\(\)](#)
- [tcp\\_txfree\(\)](#)
- [tcp\\_state\(\)](#)

# tcp\_server()

---

서버로써 TCP 접속 대기

## Description

```
void tcp_server(int $tcp_id, int $port)
```

## Parameters

- \$tcp\_id: TCP 접속에 사용할 TCP 아이디
- \$port: 접속 대기할 TCP 포트 번호

## Return Value

없음

## Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470; // 접속을 대기할 TCP 포트 번호
tcp_server(0, $port); // 0번 TCP로 접속 대기
sleep(15);
?>
```

## See also

- [sn\\_tcp\\_ac.php](#)
- [tcp\\_client\(\)](#)
- [tcp\\_read\(\)](#)
- [tcp\\_readn\(\)](#)
- [tcp\\_write\(\)](#)
- [tcp\\_txfree\(\)](#)
- [tcp\\_state\(\)](#)

# tcp\_read()

지정한 TCP 세션의 수신 데이터를 읽음

## Description

```
int tcp_read(int $tcp_id, string &$rbuf [, int $rlen = MAX_STRING_LEN])
```

## Parameters

- \$tcp\_id: 데이터를 수신할 TCP 세션의 아이디
- \$rbuf: 데이터를 수신하여 저장할 버퍼
- \$rlen: 수신할 데이터의 크기, 생략하면 최대 길이로 설정됨

## Return Value

성공 시 읽은 데이터 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port); // 0번 TCP로 접속 대기
$rbuf = "";
while(1)
{
    $rlen = tcp_read(0, $rbuf); // TCP 수신 데이터를 읽음
    if($rlen > 0)
        echo "$rbufWrWn"; // 데이터 출력
}
?>
```

## See also

- [sn\\_tcp\\_ac.php](#)
- [tcp\\_client\(\)](#)
- [tcp\\_server\(\)](#)
- [tcp\\_readn\(\)](#)
- [tcp\\_write\(\)](#)
- [tcp\\_txfree\(\)](#)
- [tcp\\_state\(\)](#)

# tcp\_readn()

지정한 TCP 세션의 수신 데이터를 지정된 크기만큼 읽음

## Description

```
int tcp_readn(int $tcp_id, string &$rbuf, int $rlen)
```

## Parameters

- \$tcp\_id: 데이터를 수신할 TCP 세션의 아이디
- \$rbuf: 데이터를 수신하여 저장할 버퍼
- \$rlen: 수신할 데이터의 크기, 이 크기 이상의 데이터가 수신되면 읽음

## Return Value

성공 시 읽은 데이터 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port); // 0번 TCP로 접속 대기
$rbuf = "";
while(1)
{
    $rlen = tcp_readn(0, $rbuf, 100); // TCP 수신 데이터가 100바이트 이상이면 읽음
    if($rlen > 0)
        echo "$rbuf\r\n"; // 데이터 출력
}
?>
```

## See also

- [sn\\_tcp\\_ac.php](#)
- [tcp\\_client\(\)](#)
- [tcp\\_server\(\)](#)
- [tcp\\_read\(\)](#)
- [tcp\\_write\(\)](#)
- [tcp\\_txfree\(\)](#)
- [tcp\\_state\(\)](#)

# tcp\_write()

지정한 TCP 세션으로 데이터를 송신

## Description

```
int tcp_write(int $tcp_id, int/string $wbuf [, int $wlen = MAX_STRING_LEN])
```

## Parameters

- \$tcp\_id: 데이터를 송신할 TCP 세션의 아이디
- \$wbuf: 송신할 데이터
- \$wlen: 송신할 데이터의 크기, 생략하면 최대 길이로 설정됨

## Return Value

성공 시 송신한 데이터 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port); // 0번 TCP로 접속 대기
$rdbuf = "";
while(1)
{
    $rwlen = tcp_read(0, $rdbuf); // TCP 수신 데이터를 읽음
    if($rwlen > 0)
        tcp_write(0, $rdbuf); // 수신한 데이터를 그대로 송신
}
?>
```

## See also

- [sn\\_tcp\\_ac.php](#)
- [tcp\\_client\(\)](#)
- [tcp\\_server\(\)](#)
- [tcp\\_read\(\)](#)
- [tcp\\_readn\(\)](#)
- [tcp\\_txfree\(\)](#)
- [tcp\\_state\(\)](#)



# tcp\_txfree()

지정한 TCP 세션의 송신 버퍼 여유 공간을 반환

## Description

```
int tcp_txfree(int $tcp_id)
```

## Parameters

- \$tcp\_id: 송신 버퍼 여유 공간을 확인할 TCP 세션의 아이디

## Return Value

성공 시 송신 버퍼 여유 공간(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port); // 0번 TCP로 접속 대기
$rbuf = "";
while(1)
{
    $rlen = tcp_read(0, $rbuf, tcp_txfree(0)); // TCP 송신 버퍼 여유 공간만큼 수신 데이터를 읽음
    if($rlen > 0)
        echo "$rbufWrWn"; // 데이터 출력
}
?>
```

## See also

- [sn\\_tcp\\_ac.php](#)
- [tcp\\_client\(\)](#)
- [tcp\\_server\(\)](#)
- [tcp\\_read\(\)](#)
- [tcp\\_readn\(\)](#)
- [tcp\\_write\(\)](#)
- [tcp\\_state\(\)](#)

# tcp\_state()

---

지정한 TCP 세션의 상태를 반환

## Description

```
int tcp_state(int $tcp_id)
```

## Parameters

- \$tcp\_id: 상태를 확인할 TCP 세션의 아이디

## Return Value

TCP 세션 상태를 나타내는 정수 값(TCP\_CLOSED, TCP\_CONNECTED 또는 TCP\_LISTEN)

## Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port); // 0번 TCP로 접속 대기
$rbuf = "";
while(tcp_state(0) != TCP_CONNECTED)
;
echo "TCP connected!\WrWn";
?>
```

## See also

- [sn\\_tcp\\_ac.php](#)
- [tcp\\_client\(\)](#)
- [tcp\\_server\(\)](#)
- [tcp\\_read\(\)](#)
- [tcp\\_readn\(\)](#)
- [tcp\\_write\(\)](#)
- [tcp\\_txfree\(\)](#)

# ws\_setup()

웹 소켓 서버 설정 및 접속 대기

## Description

```
void ws_setup(int $tcp_id, string $path, string $proto [, $port = 0])
```

## Parameters

- \$tcp\_id: 접속에 사용할 TCP 아이디
- \$path: 접속 경로
- \$proto: 통신 프로토콜
- \$port: 웹 소켓 포트, 생략하면 기본값은 0

## Return Value

없음

## Example

```
<?php
include "/lib/sn_tcp_ws.php";
ws_setup(0, "my_path", "my_proto"); // 웹 소켓 서버 설정 및 접속 대기
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        ws_write(0, "hello, world!\r\n"); // 메시지 전송
        sleep(1);
    }
}
?>
```

## See also

- [sn\\_tcp\\_ws.php](#)
- [ws\\_read\(\)](#)
- [ws\\_readn\(\)](#)
- [ws\\_read\\_line\(\)](#)
- [ws\\_write\(\)](#)
- [ws\\_txfree\(\)](#)
- [ws\\_state\(\)](#)

# ws\_read()

웹 소켓 세션의 수신 데이터를 읽음

## Description

```
int ws_read(int $tcp_id, string &$rbuf [, int $rlen = MAX_STRING_LEN])
```

## Parameters

- \$tcp\_id: 데이터를 수신할 웹 소켓 세션의 TCP 아이디
- \$rbuf: 데이터를 수신하여 저장할 버퍼
- \$rlen: 수신할 데이터의 크기, 생략하면 최대 길이로 설정됨

## Return Value

성공 시 읽은 데이터 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ws.php";
ws_setup(0, "my_path", "my_proto"); // 웹 소켓 서버 설정 및 접속 대기
$rbuf = "";
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        ws_read(0, $rbuf); // 수신 데이터 읽기
        if(strlen($rbuf) > 0)
        {
            hexdump($rbuf); // 수신 데이터 출력
            $rbuf = "";
        }
        sleep(1);
    }
}
?>
```

## See also

- [sn\\_tcp\\_ws.php](#)
- [ws\\_setup\(\)](#)
- [ws\\_readn\(\)](#)
- [ws\\_read\\_line\(\)](#)
- [ws\\_write\(\)](#)
- [ws\\_txfree\(\)](#)

- [ws\\_state\(\)](#)

# ws\_readn()

웹 소켓 세션의 수신 데이터를 지정된 크기만큼 읽음

## Description

```
int ws_readn(int $tcp_id, string &$rbuf, int $rlen)
```

## Parameters

- \$tcp\_id: 데이터를 수신할 웹 소켓 세션의 TCP 아이디
- \$rbuf: 데이터를 수신하여 저장할 버퍼
- \$rlen: 수신할 데이터의 크기, 이 크기 이상의 데이터가 수신되면 읽음

## Return Value

성공 시 읽은 데이터 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ws.php";
ws_setup(0, "my_path", "my_proto"); // 웹 소켓 서버 설정 및 접속 대기
$rbuf = "";
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        ws_readn(0, $rbuf, 10); // 수신 데이터가 10바이트 이상 되면 읽기
        if(strlen($rbuf) > 0)
        {
            hexdump($rbuf); // 데이터 출력
            $rbuf = "";
        }
        sleep(1);
    }
}
?>
```

## See also

- [sn\\_tcp\\_ws.php](#)
- [ws\\_setup\(\)](#)
- [ws\\_read\(\)](#)
- [ws\\_read\\_line\(\)](#)
- [ws\\_write\(\)](#)
- [ws\\_txfree\(\)](#)

- [ws\\_state\(\)](#)

# ws\_read\_line()

웹 소켓 세션의 수신 데이터를 줄 단위(CR + LF로 구분)로 읽음

## Description

```
int ws_read_line(int $tcp_id, string &$rbuf)
```

## Parameters

- \$tcp\_id: 데이터를 수신할 웹 소켓 세션의 TCP 아이디
- \$rbuf: 데이터를 수신하여 저장할 버퍼

## Return Value

성공 시 읽은 데이터 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ws.php";
ws_setup(0, "my_path", "my_proto"); // 웹 소켓 서버 설정 및 접속 대기
$rbuf = "";
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        ws_read_line(0, $rbuf); // 수신 데이터를 한 줄 읽음
        if(strlen($rbuf) > 0)
        {
            hexdump($rbuf); // 수신 데이터 출력
            $rbuf = "";
        }
        sleep(1);
    }
}
?>
```

## See also

- [sn\\_tcp\\_ws.php](#)
- [ws\\_setup\(\)](#)
- [ws\\_read\(\)](#)
- [ws\\_readn\(\)](#)
- [ws\\_write\(\)](#)
- [ws\\_txfree\(\)](#)
- [ws\\_state\(\)](#)



# ws\_write()

웹 소켓 세션으로 데이터를 송신

## Description

```
int ws_write(int $tcp_id, int/string $wbuf [, int $wlen = MAX_STRING_LEN])
```

## Parameters

- \$tcp\_id: 데이터를 송신할 웹 소켓 세션의 TCP 아이디
- \$wbuf: 송신할 데이터
- \$wlen: 송신할 데이터의 크기, 생략하면 최대 길이로 설정됨

## Return Value

성공 시 송신한 데이터 크기(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ws.php";
$rdbuf = "";
ws_setup(0, "my_path", "my_proto"); // 웹 소켓 서버 설정 및 접속 대기
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        $rwlen = ws_read(0, $rdbuf); // 수신 데이터를 읽기
        if($rwlen > 0)
            ws_write(0, $rdbuf); // 수신한 데이터를 그대로 송신
    }
    sleep(1);
}
?>
```

## See also

- [sn\\_tcp\\_ws.php](#)
- [ws\\_setup\(\)](#)
- [ws\\_read\(\)](#)
- [ws\\_readn\(\)](#)
- [ws\\_read\\_line\(\)](#)
- [ws\\_txfree\(\)](#)
- [ws\\_state\(\)](#)

# ws\_txfree()

웹 소켓 세션의 송신 버퍼 여유 공간을 반환

## Description

```
int ws_txfree(int $tcp_id)
```

## Parameters

- \$tcp\_id: 송신 버퍼 여유 공간을 확인할 웹 소켓 세션의 TCP 아이디

## Return Value

성공 시 송신 버퍼 여유 공간(바이트 수), 실패 시 0

## Example

```
<?php
include "/lib/sn_tcp_ws.php";
$rdbuf = "";
ws_setup(0, "my_path", "my_proto"); // 웹 소켓 서버 설정 및 접속 대기
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        $rflen = ws_read(0, $rdbuf, ws_txfree(0)); // 송신 버퍼 여유 공간만큼 수신 데이터를 읽기
        if($rflen > 0)
            ws_write(0, $rdbuf); // 수신한 데이터를 그대로 송신
    }
    sleep(1);
}
?>
```

## See also

- [sn\\_tcp\\_ws.php](#)
- [ws\\_setup\(\)](#)
- [ws\\_read\(\)](#)
- [ws\\_readn\(\)](#)
- [ws\\_read\\_line\(\)](#)
- [ws\\_write\(\)](#)
- [ws\\_state\(\)](#)

# ws\_state()

---

웹 소켓 세션의 상태를 반환

## Description

```
int ws_state(int $tcp_id)
```

## Parameters

- \$tcp\_id: 상태를 확인할 웹 소켓 세션의 TCP 아이디

## Return Value

TCP 세션 상태를 나타내는 정수 값(TCP\_CLOSED, TCP\_CONNECTED 또는 TCP\_LISTEN)

## Example

```
<?php
include "/lib/sn_tcp_ws.php";
ws_setup(0, "my_path", "my_proto"); // 웹 소켓 서버 설정 및 접속 대기
while(ws_state(0) != TCP_CONNECTED)
;
echo "Web Socket connected!\r\n";
?>
```

## See also

- [sn\\_tcp\\_ws.php](#)
- [ws\\_setup\(\)](#)
- [ws\\_read\(\)](#)
- [ws\\_readn\(\)](#)
- [ws\\_read\\_line\(\)](#)
- [ws\\_write\(\)](#)
- [ws\\_txfree\(\)](#)

# mysql\_close()

MySQL 접속을 종료

## Description

bool mysql\_close()

## Parameters

없음

## Return Value

성공 시 true, 실패 시 false

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$server_addr = "192.168.0.100"; // MySQL 서버의 IP주소
$user_name = "user_id"; // MySQL 계정의 아이디
$password = "password"; // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error()); // 에러메시지 출력 및 종료

mysql_close(); // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_connect()

주어진 계정 정보를 이용하여 MySQL DB 서버로 접속 시도

## Description

```
bool mysql_connect(string $server, string $user_name, string $raw_pwd)
```

## Parameters

- \$server: MySQL 서버의 IP주소 또는 호스트네임
- \$user\_name: MySQL 계정의 아이디
- \$raw\_pwd: MySQL 계정의 비밀번호

## Return Value

성공 시 true, 실패 시 false

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$server_addr = "192.168.0.100"; // MySQL 서버의 IP주소
$user_name = "user_id"; // MySQL 계정의 아이디
$password = "password"; // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error()); // 에러메시지 출력 및 종료

mysql_close(); // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)

- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_query()

SQL 쿼리 문자열 전송

## Description

bool/string mysql\_query(string \$query)

## Parameters

- \$query: SQL 쿼리 문자열

## Return Value

- INSERT, UPDATE, DELETE등: 성공 시 true, 실패 시 false
- SELECT, SHOW등: 성공 시 응답데이터, 실패 시 false

## Example

```
<?php
include_once "/lib/sn_dns.php";           // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$query_create_db = "CREATE DATABASE testdb;"; // 데이터베이스 생성 쿼리 문자열
$query_drop_db = "DROP DATABASE testdb;";    // 데이터베이스 제거 쿼리 문자열

$server_addr = "192.168.0.100";           // MySQL 서버의 IP주소
$user_name = "user_id";                   // MySQL 계정의 아이디
$password = "password";                   // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                  // 에러메시지 출력 및 종료

if(mysql_query($query_create_db) === true) // 쿼리 전송
    echo "create db success!\r\n";
else
    exit(mysql_error());                  // 에러메시지 출력 및 종료

if(mysql_query($query_drop_db) === true)  // 쿼리 전송
    echo "drop db success!\r\n";
else
    exit(mysql_error());                  // 에러메시지 출력 및 종료

mysql_close();                             // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)



# mysql\_ping()

서버로 접속 상태 확인을 요청

## Description

bool mysql\_ping(void)

## Parameters

없음

## Return Value

성공 시 true, 실패 시 false

## Example

```
<?php
include_once "/lib/sn_mysql.php";

$server_addr = "192.168.0.100";           // MySQL 서버의 IP주소
$user_name = "user_id";                  // MySQL 계정의 아이디
$password = "password";                  // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_ping() === true)                // 접속 상태 확인 요청
    echo "connection is alive\r\n";
else
    echo "connection is dead\r\n";

mysql_close();                           // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)

- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_select\_db()

MySQL DB 선택 쿼리 전송

## Description

bool/string mysql\_select\_db(string \$db\_name)

## Parameters

- \$db\_name: 선택할 DB 이름

## Return Value

성공 시 true, 실패 시 false 또는 응답데이터

## Example

```
<?php
include_once "/lib/sn_dns.php";           // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$db_name = "testdb";                     // 선택할 DB 이름
$query_create_db = "CREATE DATABASE $db_name;"; // 데이터베이스 생성 쿼리 문자열
$query_drop_db = "DROP DATABASE $db_name;";   // 데이터베이스 제거 쿼리 문자열

$server_addr = "192.168.0.100";         // MySQL 서버의 IP주소
$user_name = "user_id";                 // MySQL 계정의 아이디
$password = "password";                 // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_create_db) === false) // 쿼리 전송
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_select_db($db_name) === true)    // DB 선택 쿼리 전송
    echo "Database $db_name is selected!\r\n";
else
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_drop_db) === false) // 쿼리 전송
    exit(mysql_error());                 // 에러메시지 출력 및 종료

mysql_close();                           // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_error()

MySQL 서버의 에러 응답에 대한 메시지 반환

## Description

string mysql\_error()

## Parameters

없음

## Return Value

에러가 있는 경우 에러 메시지, 없는 경우 빈 문자열

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$server_addr = "192.168.0.100"; // MySQL 서버의 IP주소
$user_name = "user_id"; // MySQL 계정의 아이디
$password = "password"; // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error()); // 에러메시지 출력 및 종료

mysql_close(); // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_errno()

MySQL 서버의 에러 응답에 대한 에러 코드 반환

## Description

```
int mysql_errno()
```

## Parameters

없음

## Return Value

에러가 있는 경우 에러 코드, 없는 경우 0

## Example

```
<?php
include_once "/lib/sn_dns.php";    // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$server_addr = "192.168.0.100";    // MySQL 서버의 IP주소
$user_name = "user_id";           // MySQL 계정의 아이디
$password = "password";           // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_errno());           // 에러코드 출력 및 종료

mysql_close();                     // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_affected\_rows()

SQL 쿼리에 의해 변경된 레코드 수 반환

## Description

```
int mysql_affected_rows(string $result)
```

## Parameters

- \$result: DB 서버의 응답메시지

## Return Value

성공 시 변경된 레코드 수, 실패 시 -1

## Example

```
<?php
include_once "/lib/sn_dns.php";           // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$db_name = "testdb";                     // 선택할 DB 이름

// 데이터베이스 생성, 테이블 생성, 레코드 삽입 및 추가 등의 쿼리 문자열
$query_create_db = "CREATE DATABASE testdb;";
$query_create_table = "CREATE TABLE test_table (id INTEGER NOT NULL, age INTEGER);";
$query_insert_1 = "INSERT INTO test_table (id, age) VALUES ('1', '10');";
$query_insert_2 = "INSERT INTO test_table (id, age) VALUES ('2', '10');";
$query_update = "UPDATE test_table SET age=age+1 WHERE age=10;";
$query_drop_table = "DROP TABLE test_table;";
$query_drop_db = "DROP DATABASE testdb;";

$server_addr = "192.168.0.100";          // MySQL 서버의 IP주소
$user_name = "user_id";                 // MySQL 계정의 아이디
$password = "password";                 // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_create_db) === false) // 쿼리 전송(DB 생성)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_select_db($db_name) === false)   // DB 선택 쿼리 전송
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_create_table) === false) // 쿼리 전송(테이블 생성)
    exit(mysql_error());                 // 에러메시지 출력 및 종료
```

```

if(mysql_query($query_insert_1) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

if(mysql_query($query_insert_2) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

$result = mysql_query($query_update);         // 쿼리 전송(레코드 변경)
if($result === false)
    exit(mysql_error());                      // 에러메시지 출력 및 종료
else
{
    $affected_rows = mysql_affected_rows($result); // 변경 된 레코드 수 확인
    if($affected_rows > 0)
        echo "$affected_rows record(s) have(has) been updated!";
}

if(mysql_query($query_drop_table) === false)  // 쿼리 전송(테이블 삭제)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

if(mysql_query($query_drop_db) === false)    // 쿼리 전송(DB 삭제)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

mysql_close();                               // MySQL 접속 종료
?>

```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)



# mysql\_num\_rows()

SQL 쿼리의 결과 레코드 수 반환

## Description

```
int mysql_num_rows(string $result)
```

## Parameters

- \$result: DB서버의 응답메시지

## Return Value

성공 시 결과 레코드 수, 실패 시 -1

## Example

```
<?php
include_once "/lib/sn_dns.php";           // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$db_name = "testdb";                     // 선택할 DB 이름

// 데이터베이스 생성, 테이블 생성, 레코드 삽입 및 추가 등의 쿼리 문자열
$query_create_db = "CREATE DATABASE testdb;";
$query_create_table = "CREATE TABLE test_table (id INTEGER NOT NULL, age INTEGER);";
$query_insert_1 = "INSERT INTO test_table (id, age) VALUES ('1', '10');";
$query_insert_2 = "INSERT INTO test_table (id, age) VALUES ('2', '10');";
$query_select = "SELECT * FROM test_table";
$query_drop_table = "DROP TABLE test_table;";
$query_drop_db = "DROP DATABASE testdb;";

$server_addr = "192.168.0.100";          // MySQL 서버의 IP주소
$user_name = "user_id";                 // MySQL 계정의 아이디
$password = "password";                 // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_create_db) === false) // 쿼리 전송(DB 생성)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_select_db($db_name) === false)   // DB 선택 쿼리 전송
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_create_table) === false) // 쿼리 전송(테이블 생성)
    exit(mysql_error());                 // 에러메시지 출력 및 종료
```

```

if(mysql_query($query_insert_1) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

if(mysql_query($query_insert_2) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

$result = mysql_query($query_select);         // SELECT 쿼리 전송
if($result === false)
    exit(mysql_error());                      // 에러메시지 출력 및 종료
else
{
    $num_rows = mysql_num_rows($result);      // 결과 레코드 수 확인
    if($num_rows > 0)
        echo "$num_rows records has been detected!\n\n";
}

if(mysql_query($query_drop_table) === false) // 쿼리 전송(테이블 삭제)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

if(mysql_query($query_drop_db) === false)    // 쿼리 전송(DB 삭제)
    exit(mysql_error());                      // 에러메시지 출력 및 종료

mysql_close();                               // MySQL 접속 종료
?>

```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_result()

DB 테이블의 특정행, 열에 해당하는 결과 레코드를 반환

## Description

```
string mysql_result(string $result, int $row [, int $col = 0])
```

## Parameters

- \$result: DB서버의 응답메시지
- \$row: 행 번호
- \$col: 열 번호, 생략하면 가장 첫 번째 열로 선택

## Return Value

성공 시 해당 데이터값, 실패 시 false

## Example

```
<?php
include_once "/lib/sn_dns.php";           // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$db_name = "testdb";                     // 선택할 DB 이름

// 데이터베이스 생성, 테이블 생성, 레코드 삽입 및 추가 등의 쿼리 문자열
$query_create_db = "CREATE DATABASE testdb;";
$query_create_table = "CREATE TABLE test_table (id INTEGER NOT NULL, age INTEGER);";
$query_insert_1 = "INSERT INTO test_table (id, age) VALUES ('1', '10');";
$query_insert_2 = "INSERT INTO test_table (id, age) VALUES ('2', '10');";
$query_select = "SELECT * FROM test_table";
$query_drop_table = "DROP TABLE test_table;";
$query_drop_db = "DROP DATABASE testdb;";

$server_addr = "192.168.0.100";          // MySQL 서버의 IP주소
$user_name = "user_id";                 // MySQL 계정의 아이디
$password = "password";                 // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_create_db) === false) // 쿼리 전송(DB 생성)
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_select_db($db_name) === false) // DB 선택 쿼리 전송
    exit(mysql_error());                 // 에러메시지 출력 및 종료

if(mysql_query($query_create_table) === false) // 쿼리 전송(테이블 생성)
```

```

exit(mysql_error());          // 에러메시지 출력 및 종료

if(mysql_query($query_insert_1) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());          // 에러메시지 출력 및 종료

if(mysql_query($query_insert_2) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());          // 에러메시지 출력 및 종료

$result = mysql_query($query_select);        // SELECT 쿼리 전송
if($result === false)
    exit(mysql_error());          // 에러메시지 출력 및 종료
else
{
    $value = mysql_result($result, 1, 1);    // 1행 1열의 데이터 확인
    if($value)
        echo "$value\r\n";                // 1행 1열의 데이터 출력
    else
        echo "No data has been detected!\r\n";
}

if(mysql_query($query_drop_table) === false) // 쿼리 전송(테이블 삭제)
    exit(mysql_error());          // 에러메시지 출력 및 종료

if(mysql_query($query_drop_db) === false)    // 쿼리 전송(DB 삭제)
    exit(mysql_error());          // 에러메시지 출력 및 종료

mysql_close();                    // MySQL 접속 종료
?>

```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_fetch\_row()

DB 테이블의 각 레코드를 순서대로 반환

## Description

array mysql\_fetch\_row(string \$result)

## Parameters

- \$result: DB 서버의 응답메시지

## Return Value

성공 시 배열형태의 단위 레코드, 실패 시 false

## Example

```
<?php
include_once "/lib/sn_dns.php";           // DNS 라이브러리
include_once "/lib/sn_mysql.php";

$db_name = "testdb";                     // 선택할 DB 이름

// 데이터베이스 생성, 테이블 생성, 레코드 삽입 및 추가 등의 쿼리 문자열
$query_create_db = "CREATE DATABASE testdb;";
$query_create_table = "CREATE TABLE test_table (id INTEGER NOT NULL, age INTEGER);";
$query_insert_1 = "INSERT INTO test_table (id, age) VALUES ('1', '10');";
$query_insert_2 = "INSERT INTO test_table (id, age) VALUES ('2', '10');";
$query_select = "SELECT * FROM test_table";
$query_drop_table = "DROP TABLE test_table;";
$query_drop_db = "DROP DATABASE testdb;";

$server_addr = "192.168.0.100";           // MySQL 서버의 IP주소
$user_name = "user_id";                  // MySQL 계정의 아이디
$password = "password";                   // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                  // 에러메시지 출력 및 종료

if(mysql_query($query_create_db) === false) // 쿼리 전송(DB 생성)
    exit(mysql_error());                  // 에러메시지 출력 및 종료

if(mysql_select_db($db_name) === false)   // DB 선택 쿼리 전송
    exit(mysql_error());                  // 에러메시지 출력 및 종료

if(mysql_query($query_create_table) === false) // 쿼리 전송(테이블 생성)
    exit(mysql_error());                  // 에러메시지 출력 및 종료
```

```

if(mysql_query($query_insert_1) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());                    // 에러메시지 출력 및 종료

if(mysql_query($query_insert_2) === false)    // 쿼리 전송(레코드 삽입)
    exit(mysql_error());                    // 에러메시지 출력 및 종료

$result = mysql_query($query_select);        // SELECT 쿼리 전송
if($result === false)
    exit(mysql_error());                    // 에러메시지 출력 및 종료
else
{
    while(1)
    {
        $arr = mysql_fetch_row($result);    // 단위 레코드 반환
        if($arr === false)
            break;
        $arr_len = count($arr);            // 레코드의 행 개수
        for($i = 0; $i < $arr_len; $i++)
        {
            if($arr[$i] !== "")
                printf("%s ", $arr[$i]);    // 레코드 출력
            else
                break;
        }
        echo "WrWn";
    }
}

if(mysql_query($query_drop_table) === false) // 쿼리 전송(테이블 삭제)
    exit(mysql_error());                    // 에러메시지 출력 및 종료

if(mysql_query($query_drop_db) === false)    // 쿼리 전송(DB 삭제)
    exit(mysql_error());                    // 에러메시지 출력 및 종료

mysql_close();                             // MySQL 접속 종료
?>

```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)
- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_setup\(\)](#)

# mysql\_setup()

MySQL 사용을 위한 설정

## Description

```
void mysql_setup(int $udp_id = 4, int $tcp_id = 4[, string $dns_server = ""[, bool $ip6 = false]])
```

## Parameters

- \$udp\_id: DNS용 UDP 디바이스 번호
- \$tcp\_id: MySQL용 TCP 디바이스 번호
- \$dns\_server: DNS쿼리를 위한 DNS 서버 주소
- \$ip6: IPv6 활성화 또는 비활성화

## Return Value

없음

## Example

```
<?php
include_once "/lib/sn_dns.php"; // DNS 라이브러리
include_once "/lib/sn_mysql.php";

mysql_setup(4, 4, "", false); // MySQL 사용을 위한 설정

$server_addr = "192.168.0.100"; // MySQL 서버의 IP주소
$user_name = "user_id"; // MySQL 계정의 아이디
$password = "password"; // MySQL 계정의 비밀번호

// MySQL 접속 시도
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error()); // 에러메시지 출력 및 종료

mysql_close(); // MySQL 접속 종료
?>
```

## See also

- [sn\\_mysql.php](#)
- [sn\\_dns.php](#)
- [mysql\\_close\(\)](#)
- [mysql\\_connect\(\)](#)
- [mysql\\_query\(\)](#)
- [mysql\\_select\\_db\(\)](#)
- [mysql\\_error\(\)](#)

- [mysql\\_errno\(\)](#)
- [mysql\\_affected\\_rows\(\)](#)
- [mysql\\_num\\_rows\(\)](#)
- [mysql\\_result\(\)](#)
- [mysql\\_fetch\\_row\(\)](#)



# spc\_reset()

---

스마트 확장보드를 사용하기 위한 초기화

## Description

```
void spc_reset(int $t1 = 10, int $t2 = 500)
```

## Parameters

- \$t1 : 리셋 신호 출력 시간(밀리초)
- \$t2 : 리셋 대기시간(밀리초)

## Return Value

없음

## Example

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

## See also

- [sd\\_spc.php](#)
- [spc\\_sync\\_baud\(\)](#)
- [spc\\_scan\(\)](#)
- [spc\\_ioctl\(\)](#)
- [spc\\_request\(\)](#)
- [spc\\_request\\_sys\(\)](#)
- [spc\\_request\\_dev\(\)](#)

# spc\_sync\_baud()

스마트 확장보드와의 통신속도 설정

## Description

```
void spc_sync_baud(int $baud = 115200, int $t1 = 1, int $t2 = 100)
```

## Parameters

- \$baud : 통신 속도 (bps)
- \$t1 : 동기화 신호 출력 시간 (밀리초)
- \$t2 : 동기화 대기시간 (밀리초)

## Return Value

없음

## Example

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

## See also

- [sd\\_spc.php](#)
- [spc\\_reset\(\)](#)
- [spc\\_scan\(\)](#)
- [spc\\_ioctl\(\)](#)
- [spc\\_request\(\)](#)
- [spc\\_request\\_sys\(\)](#)
- [spc\\_request\\_dev\(\)](#)

# spc\_scan()

---

연결 된 스마트 확장보드 검색

## Description

int spc\_scan(int \$start = 1, int \$end = 14, bool \$verbose = false)

## Parameters

- \$start: 시작 아이디
- \$end: 종료 아이디
- \$verbose: 검색결과 출력 여부(true: 출력, false: 출력 안 함)

## Return Value

성공적으로 검색한 스마트 확장보드 개수

## Example

```
<?php
include "/lib/sd_spc.php";
$count = 0;
spc_reset();
spc_sync_baud();
$count = spc_scan(1, 14, true);
echo "$count smart expansion board(s) has(have) been detected!\r\n";
?>
```

## See also

- [sd\\_spc.php](#)
- [spc\\_reset\(\)](#)
- [spc\\_sync\\_baud\(\)](#)
- [spc\\_ioctl\(\)](#)
- [spc\\_request\(\)](#)
- [spc\\_request\\_sys\(\)](#)
- [spc\\_request\\_dev\(\)](#)

# spc\_ioctl()

---

SPC 디바이스의 설정 및 사용

## Description

int/string spc\_ioctl(string \$cmd)

## Parameters

- \$cmd: SPC 설정 및 사용을 위한 명령어 문자열

## Return Value

해당 명령어에 대한 반환 값

## Example

```
<?php
include "/lib/sd_spc.php";
$cmd = "set baud 115200";
spc_ioctl($cmd);
?>
```

## See also

- [sd\\_spc.php](#)
- [spc\\_reset\(\)](#)
- [spc\\_sync\\_baud\(\)](#)
- [spc\\_scan\(\)](#)
- [spc\\_request\(\)](#)
- [spc\\_request\\_sys\(\)](#)
- [spc\\_request\\_dev\(\)](#)

# spc\_request()

스마트 확장보드로 명령어 전달을 요청

## Description

string/bool spc\_request(int \$sid, int \$addr, string \$msg, string \$opt = "")

## Parameters

- \$sid: 슬레이브 아이디
- \$addr: 명령 주소
- \$msg: 명령어
- \$opt: 명령어에 따른 옵션

## Return Value

성공 시 명령어에 따른 스마트 확장보드의 응답(CSV형태의 문자열), 실패 시 false

※ CSV : Comma-Separated Values

## Example

```
<?php
include_once "/lib/sd_spc.php";

spc_reset();
spc_sync_baud(115200);

spc_scan();

$sid = 1;

function eio_in($sid, $pin)
{
    $resp = spc_request($sid, 4, "get $pin input");

    if($resp === false)
        return "";

    $resp = explode(",", $resp);

    return $resp[1];
}

while(1)
{
    echo eio_in($sid, 0);
    echo eio_in($sid, 1);
    echo eio_in($sid, 2);
```

```
echo eio_in($sid, 3);  
echo "WrWn";  
  
sleep(1);  
}  
?>
```

## See also

- [sd\\_spc.php](#)
- [spc\\_reset\(\)](#)
- [spc\\_sync\\_baud\(\)](#)
- [spc\\_scan\(\)](#)
- [spc\\_ioctl\(\)](#)
- [spc\\_request\\_sys\(\)](#)
- [spc\\_request\\_dev\(\)](#)

# spc\_request\_sys()

스마트 확장보드로 시스템 공통 명령어 전달을 요청

## Description

string spc\_request\_sys(int \$sid, string \$cmd)

## Parameters

- \$sid: 슬레이브 아이디
- \$cmd: 명령어

## Return Value

명령어에 따른 스마트 확장보드의 응답, 문자열 형태

## Example

```
<?php
include_once "/lib/sd_spc.php";

spc_reset();
spc_sync_baud(115200);
spc_scan();

$sid = 1;

echo spc_request_sys($sid, "get did");
echo spc_request_sys($sid, "get uid");
?>
```

## See also

- [sd\\_spc.php](#)
- [spc\\_reset\(\)](#)
- [spc\\_sync\\_baud\(\)](#)
- [spc\\_scan\(\)](#)
- [spc\\_ioctl\(\)](#)
- [spc\\_request\(\)](#)
- [spc\\_request\\_dev\(\)](#)

# spc\_request\_dev()

스마트 확장보드로 디바이스 명령어 전달을 요청

## Description

string spc\_request\_dev(int \$sid, string \$cmd)

## Parameters

- \$sid: 슬레이브 아이디
- \$cmd: 명령어

## Return Value

명령어에 따른 스마트 확장보드의 응답, 문자열 형태

## Example

```
<?php
/* This example is for PES-2403 only */
include_once "/lib/sd_spc.php";

spc_reset();
spc_sync_baud(460800);

$sid = 1;
spc_request_dev($sid, "set mode full");
spc_request_dev($sid, "set vref stop 2");
spc_request_dev($sid, "set vref drive 8");
spc_request_dev($sid, "set speed 400");
spc_request_dev($sid, "set accel 800");
spc_request_dev($sid, "set rsnc 120 250");

$state = 0;
spc_request_dev($sid, "move 400");
while($state = (int)spc_request_dev($sid, "get state"))
{
    echo "state: $stateWrWn";
    usleep(200000);
}
echo "state: $stateWrWn";
?>
```

## See also

- [sd\\_spc.php](#)
- [spc\\_reset\(\)](#)



- `spc_sync_baud()`
- `spc_scan()`
- `spc_ioctl()`
- `spc_request()`
- `spc_request_sys()`

# LOW

---

LOW는 논리 0을 나타내기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)
- [sd\\_340.php](#)

# HIGH

---

HIGH는 논리 1을 나타내기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)
- [sd\\_340.php](#)

# TOGGLE

---

TOGGLE은 논리를 반전시키기 위해(0은 1로, 1은 0으로) 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)
- [sd\\_340.php](#)

---

# LED\_A

---

LED\_A는 PHPoC 외장형 제품의 A LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)

---

# LED\_B

---

LED\_B는 PHPoC 외장형 제품의 B LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)

---

# LED\_C

---

LED\_C는 PHPoC 외장형 제품의 C LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)

---

# LED\_D

---

LED\_D는 PHPoC 외장형 제품의 D LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)



---

# LED\_E

---

LED\_E는 PHPoC 외장형 제품의 E LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)

---

# LED\_F

---

LED\_F는 PHPoC 외장형 제품의 F LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)

---

# LED\_G

---

LED\_G는 PHPoC 외장형 제품의 G LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)

---

# LED\_H

---

LED\_H는 PHPoC 외장형 제품의 H LED를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_101.php](#)
- [sd\\_104.php](#)
- [sd\\_204.php](#)

# DO\_0

---

DO\_0은 PHPoC 외장형 제품의 0번 출력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)

# DO\_1

---

DO\_1은 PHPoC 외장형 제품의 1번 출력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)

# DO\_2

---

DO\_2은 PHPoC 외장형 제품의 2번 출력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)

# DO\_3

---

DO\_3은 PHPoC 외장형 제품의 3번 출력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)



---

# DI\_0

---

DI\_0은 PHPoC 외장형 제품의 0번 입력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)

---

# DI\_1

---

DI\_1은 PHPoC 외장형 제품의 1번 입력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)

---

# DI\_2

---

DI\_2은 PHPoC 외장형 제품의 2번 입력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)

---

# DI\_3

---

DI\_3은 PHPoC 외장형 제품의 3번 입력포트를 맵핑하기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sd\\_204.php](#)

---

# RR\_A

---

RR\_A는 DNS의 유형 중 호스트 주소(host address)를 나타내기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sn\\_dns.php](#)

# RR\_AAAA

---

RR\_AAAA는 DNS의 유형 중 IPv6 주소(IPv6 address)를 나타내기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sn\\_dns.php](#)

---

# RR\_NS

---

RR\_NS는 DNS의 유형 중 네임서버(authoritative name server)를 나타내기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sn\\_dns.php](#)

---

# RR\_MX

---

RR\_MX는 DNS의 유형 중 메일서버(mail exchange)를 나타내기 위해 정의한 상수입니다.

## 사용된 라이브러리

- [sn\\_dns.php](#)