
PSP Library Reference

This manual is for libraries provided in PSP.

List of Libraries

- [sd_101.php](#) : basic library of PBH-101
- [sd_104.php](#) : basic library of PBH-104
- [sd_204.php](#) : basic library of PBH-204
- [sd_340.php](#) : basic library of P4S-341 and P4S-342
- [sd_spc.php](#) : library for smart expansion boards
- [sn_dns.php](#) : DNS library
- [sn_smtp.php](#) : e-mail(SMTP) library
- [sn_esmtp.php](#) : extended e-mail(ESMTP) library
- [sn_tcp_ac.php](#) : TCP communication library
- [sn_tcp_ws.php](#) : Web socket library
- [sn_mysql.php](#) : MySQL library

Download Source Code of PSP

- [Go to the download page](#)

sd_101.php

sd_101.php is a basic library of PBH-101.

Functions

- 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()

Constants

LED Ports

- LED_A
- LED_B
- LED_C
- LED_D
- LED_E
- LED_F
- LED_G
- LED_H

LED Types

- LOW
- HIGH
- TOGGLE

sd_104.php

sd_104.php is a basic library of PBH-104.

Functions

- `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()`

Constants

LED Ports

- `LED_A`
- `LED_B`
- `LED_C`
- `LED_D`
- `LED_E`
- `LED_F`
- `LED_G`
- `LED_H`

LED Types

- `LOW`
- `HIGH`
- `TOGGLE`

sd_204.php

sd_204.php is a basic library of PBH-204.

Functions

- 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()

Constants

LED Ports

- LED_A
- LED_B
- LED_C
- LED_D
- LED_E
- LED_F
- LED_G
- LED_H

I/O Ports

- DO_0
- DO_1
- DO_2
- DO_3
- DI_0
- DI_1
- DI_2
- DI_3

LED and I/O Types

- LOW
- HIGH
- TOGGLE

sd_340.php

sd_340.php is a basic library of P4S-341 and P4S-342.

Functions

- 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()

Constants

- LOW
- HIGH
- TOGGLE

sd_spc.php

sd_spc.php is a library of smart expansion boards.

Functions

- [spc_reset\(\)](#)
- [spc_sync_baud\(\)](#)
- [spc_scan\(\)](#)
- [spc_ioctl\(\)](#)
- [spc_request\(\)](#)
- [spc_request_sys\(\)](#)
- [spc_request_dev\(\)](#)

sn_dns.php

sn_dns.php is a library for DNS (Domain Name System).

Functions

- [dns_setup\(\)](#)
- [dns_send_query\(\)](#)
- [dns_loop\(\)](#)
- [dns_lookup\(\)](#)

Constants

- [RR_A](#)
- [RR_NS](#)
- [RR_MX](#)

sn_smtp.php

sn_smtp.php is an E-mail (Simple Mail Transfer Protocol) library.

Functions

- `smtp_setup()`
- `smtp_hostname()`
- `smtp_account()`
- `smtp_loop()`
- `smtp_start()`
- `smtp_send()`

sn_esmtp.php

sn_esmtp.php is an extended E-mail (Extended Simple Mail Transfer Protocol) library.

Functions

- [esmtp_setup\(\)](#)
- [esmtp_hostname\(\)](#)
- [esmtp_account\(\)](#)
- [esmtp_auth\(\)](#)
- [esmtp_msa\(\)](#)
- [esmtp_loop\(\)](#)
- [esmtp_start\(\)](#)
- [esmtp_send\(\)](#)

sn_tcp_ac.php

sn_tcp_ac.php is a TCP Communication library.

Functions

- [tcp_client\(\)](#)
- [tcp_server\(\)](#)
- [tcp_read\(\)](#)
- [tcp_readn\(\)](#)
- [tcp_write\(\)](#)
- [tcp_txfree\(\)](#)
- [tcp_state\(\)](#)

sn_tcp_ws.php

sn_tcp_ws.php is Web Socket library.

Functions

- [ws_setup\(\)](#)
- [ws_read\(\)](#)
- [ws_readn\(\)](#)
- [ws_write\(\)](#)
- [ws_txfree\(\)](#)
- [ws_state\(\)](#)

sn_mysql.php

sn_mysql.php is a MySQL library.

Requirements

- [DNS library](#)

Functions

- [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()

Set the type of a designated LED

Description

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

Parameters

- \$pin: a LED port
- \$mode: a Type of LED

Return Value

None

Example

```
<?php
include "/lib/sd_101.php";
led_setup(LED_A, "led_net0_link"); // Setting the LED_A to network ACT LED
led_setup(LED_B, "led_net0_rx");   // Setting the LED_B to network receive LED
?>
```

See also

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)
- [led_out\(\)](#)
- [led_in\(\)](#)

led_out()

Outputs HIGH or LOW signal to a designated LED port

Description

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

Parameters

- \$pin: a port number
- \$type: output logic(LOW, HIGH or TOGGLE)

Return Value

none

Example

```
<?php
include "/lib/sd_101.php";
led_setup(LED_A, "out"); // Setting the LED_A to input
led_out(LED_A, "LOW");  // Output LOW to LED_A
?>
```

See also

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)
- [led_setup\(\)](#)
- [led_in\(\)](#)

led_in()

Reading the value of designated LED port

Description

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

Parameters

- \$pin: a LED port number

Return Value

Returns 1 when the port state is HIGH, 0 when the port state is LOW

Example

```
<?php
include "/lib/sd_101.php";
led_setup(LED_A, "in"); // Setting the LED_A to input
$state = led_in(LED_A); // Reading the LED_A
?>
```

See also

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)
- [led_setup\(\)](#)
- [led_out\(\)](#)

dio_out()

Output a signal to given port.

Description

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

Parameters

- \$pin: port number
- \$type: output signal (LOW, HIGH or TOGGLE)

Return Value

none

Example

```
<?php
include "/lib/sd_204.php";
dio_out(DO_0, HIGH); // Output HIGH to DO 0 port
sleep(1);
dio_out(DO_0, LOW); // Output LOW to DO 0 port
?>
```

See also

- [sd_204.php](#)
- [LOW](#)
- [HIGH](#)
- [TOGGLE](#)
- [DO_0](#)
- [DO_1](#)
- [DO_2](#)
- [DO_3](#)
- [dio_in\(\)](#)

dio_in()

Reading the designated port value

Description

int dio_in(int \$pin)

Parameters

- \$pin: the number of DI port

Return Values

state is HIGH, 0 when the port state is LOW

Example

```
<?php
include "/lib/sd_204.php";
echo dio_in(DI_0), "\r\n"; // Reading the port 0 of the DI
echo dio_in(DI_1), "\r\n"; // Reading the port 1 of the DI
?>
```

See also

- [sd_204.php](#)
- [dio_out\(\)](#)
- [DI_0](#)
- [DI_1](#)
- [DI_2](#)
- [DI_3](#)

uio_ioctl()

Configuring and Using the designated UIO

Description

int/string uio_ioctl(int \$uio_id, string \$cmd)

Parameters

- \$uio_id: UIO ID
- \$cmd: command set

refer to the document named [PHPoC Device Programming Guide for P40](#) for the details about commands available on \$cmd

Return Value

return value for the command set

Example

```
<?php
include "/lib/sd_340.php";
uio_ioctl(0, "set 0 mode out"); // setting the pin 0 of the UIO0 to output port
uio_ioctl(0, "set 1-3 mode in"); // setting the pin 1, 2, and 3 of the UIO0 to input ports
$state_0 = uio_ioctl(0, "get 0 output"); // getting the pin 0's state of the UIO0
$state_1 = uio_ioctl(0, "get 1 input"); // getting the pin 1's state of the UIO0
$state_2 = uio_ioctl(0, "get 2 input"); // getting the pin 2's state of the UIO0
$state_3 = uio_ioctl(0, "get 3 input"); // getting the pin 3's state of the UIO0
?>
```

See also

- [sd_340.php](#)
- [uio_setup\(\)](#)
- [uio_out\(\)](#)
- [uio_in\(\)](#)

uio_setup()

Configuring the designated UIO pin

Description

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

Parameters

- `$uio_id`: UIO ID for configuration
- `$pin`: pin number to configure
- `$mode`: pin mode

Example

```
<?php
include "/lib/sd_340.php";
uio_setup(0, 0, "out"); // configuring pin 0 of the UIO 0 to output port
uio_setup(0, 1, "out_pp"); // configuring pin 1 of the UIO 0 to push-pull output port
uio_setup(0, 2, "in"); // configuring pin 2 of the UIO 0 to input port
uio_setup(0, 3, "in_pu"); // configuring pin 3 of the UIO 0 to input with pull-up
?>
```

See also

- [sd_340.php](#)
- [uio_ioctl\(\)](#)
- [uio_out\(\)](#)
- [uio_in\(\)](#)

uio_out()

Outputs HIGH or LOW signal to the output port

Description

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

Parameters

- `$uio_id`: UIO ID
- `$pin`: pin number to output
- `$type`: output data (LOW, HIGH, or TOGGLE)

Example

```
<?php
include "/lib/sd_340.php";
uio_setup(0, 0, "out"); // configuring pin 0 of the UIO 0 to output port
uio_out(0, 0, HIGH);   // outputs HIGH to pin 0 of the UIO 0
sleep(1);
uio_out(0, 0, LOW);    // outputs LOW to pin 0 of the UIO 0
?>
```

See also

- [sd_340.php](#)
- [uio_ioctl\(\)](#)
- [uio_setup\(\)](#)
- [uio_in\(\)](#)

uio_in()

Reading the designated pin value

Description

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

Parameters

- `$uio_id`: UIO ID
- `$pin`: pin number to read

Return Value

Returns 1 when the port state is HIGH, 0 when the port state is LOW

Example

```
<?php
$in_value = 0;
include "/lib/sd_340.php";
uio_setup(0, 0, "in_pu"); // Setting the pin 0 of the UIO 0 to input with pull-up
echo $in_value = uio_in(0, 0); // Reading and printing the pin 0 of the UIO 0
?>
```

See also

- [sd_340.php](#)
- [uio_ioctl\(\)](#)
- [uio_setup\(\)](#)
- [uio_out\(\)](#)

uart_setup()

Configuring the designated UART port

Description

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

Parameters

- `$uart_id`: UART ID to configure
- `$baud`: Baudrate(unit: bps)
- `$ctrl`: parity, data bits, stop bit, and flowcontrol in a row. default value is "N81N".

ctrl	available values
parity	N: None, E: Even, O: Odd, M: Mark, S: Space
data bit	8 or 7
stop bit	1 or 2
flow control	N: None, H: RTS/CTS, S: Xon/Xoff, T: TxDE(RS485) enable

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
uart_setup(0, 9600, "E71N"); // set UART0 to 9600 bps, Even parity, 7 data bits, 1 stop bit, No-
flowcontrol
uart_setup(1, 115200);      // set UART1 to 115200bps
?>
```

See also

- [sd_340.php](#)
- [uart_read\(\)](#)
- [uart_readn\(\)](#)
- [uart_write\(\)](#)
- [uart_txfree\(\)](#)

uart_read()

Reading received data from the designated UART port

Description

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

Parameters

\$uart_id: UART ID to read \$rbuf: variable to save the received data \$rlen: maximum data length to read

Return Value

Returns the number of bytes read

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$rlen = 0;
uart_setup(0, 9600); // Configuring UART0 to 9600bps
while(1)
{
    uart_read(0, $rbuf); // Reading data from UART0 into $rbuf
    echo "$rbuf\r\n";
}
?>
```

See also

- [sd_340.php](#)
- [uart_setup\(\)](#)
- [uart_readn\(\)](#)
- [uart_write\(\)](#)
- [uart_txfree\(\)](#)

uart_readn()

Reading designated number of bytes from the UART port

Description

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

Parameters

- `$uart_id`: UART ID
- `$rbuf`: variable to save the received data
- `$rlen`: number of data to read

Return Value

On success, returns the number of read data. On fail, returns 0.(If the number of received data is less than the `$rlen` it is fail.)

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
uart_setup(0, 9600);           // Configuring UART0 to 9600 bps
while(1)
{
    $rlen = uart_readn(0, $rbuf, 5); // Reading 5 bytes of data from the UART0
    if($rlen)
    {
        echo "$rbuf\r\n";
        break;
    }
}
?>
```

See also

- [sd_340.php](#)
- [uart_setup\(\)](#)
- [uart_read\(\)](#)
- [uart_write\(\)](#)
- [uart_txfree\(\)](#)

uart_write()

Sending data to the designated UART port

Description

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

Parameters

- `$uart_id`: UART ID
- `$wbuf`: source buffer to send
- `$wlen`: byte to send

Return Value

On success, the number of bytes sent is returned. otherwise: 0

Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
uart_setup(0, 9600); // Configuring UART0 to 9600 bps
$slen = uart_write(0, $wbuf); // Sending $wbuf data to UART0
usleep(500000);
echo "$slen bytes has been sentWrWn";
?>
```

See also

- [sd_340.php](#)
- [uart_setup\(\)](#)
- [uart_read\(\)](#)
- [uart_readn\(\)](#)
- [uart_txfree\(\)](#)

uart_txfree()

Getting free buffer size of the designated UART port

Description

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

Parameters

\$uart_id: UART ID

Return Value

On success, it returns free TX buffer length of the UART, otherwise 0

Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
uart_setup(0, 9600);      // Configuring UART0 to 9600 bps
$tx_free = uart_txfree(0); // Getting free TX buffer space
if($tx_free >= 5)        // If free TX space is equal to 5 or larger than 5
    uart_write(0, $wbuf); // Sending the 5 bytes of data
?>
```

See also

- [sd_340.php](#)
- [uart_setup\(\)](#)
- [uart_read\(\)](#)
- [uart_readn\(\)](#)
- [uart_write\(\)](#)

spi_setup()

Configuring SPI

Description

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

Parameters

- `$spi_id`: SPI ID
- `$div`: clock divider for SPI (2, 4, 8, 16, 32, 64, 128 or 256)

divider	communication rate(bps)
2	21M
4	10.5M
8	5.25M
16	about 2.6M
32	about 1.3M
64	about 656K
128	about 328K
256	about 164K

- `$mode`: SPI mode (0 ~ 3)

Example

```
<?php
include "/lib/sd_340.php";
spi_setup(0);          // Configuring SPI0 with divider 256 and mode 3
spi_setup(0, 128);    // Configuring SPI0 with divider 128 and mode 3
spi_setup(0, 128, 0); // Configuring SPI0 with divider 128 and mode 0
?>
```

See also

- [sd_340.php](#)
- [spi_ioctl\(\)](#)
- [spi_read\(\)](#)
- [spi_write\(\)](#)
- [spi_write_read\(\)](#)

spi_ioctl()

Configuring or using SPI

Description

int/string spi_ioctl(int \$spi_id, string \$cmd)

Parameters

- \$spi_id: SPI ID
- \$cmd: command for configuring or using

refer to the document named [PHPoC Device Programming Guide for P40](#) for the details about commands available on \$cmd

Return Value

return value for each command

Example

```
<?php
include "/lib/sd_340.php";
spi_ioctl(0, "set mode 3); // Setting the SPI mode to 3
spi_ioctl(0, "set div 128); // Setting the division to 128
?>
```

See also

- [sd_340.php](#)
- [spi_setup\(\)](#)
- [spi_read\(\)](#)
- [spi_write\(\)](#)
- [spi_write_read\(\)](#)

spi_read()

Reading data from the SPI

Description

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

Parameters

- `$spi_id`: SPI ID
- `$rbuf`: variable to save the received data
- `$rlen`: data length to read (byte)

Return Value

data length (byte) on success, 0 on fail

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$rlen = 0;
spi_setup(0);           // configuring SPI
while(1)
{
    $rlen = spi_read(0, $rbuf, 2); // Reading 2 bytes data from the SPI
    if($rlen == 2)
        echo "$rbufWrWn";
    sleep(1);
}
?>
```

See also

- [sd_340.php](#)
- [spi_setup\(\)](#)
- [spi_ioctl\(\)](#)
- [spi_write\(\)](#)
- [spi_write_read\(\)](#)

spi_write()

Transmitting data to SPI

Description

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

Parameters

- `$spi_id`: SPI ID
- `$wbuf`: variable to transmit
- `$wlen`: data length to transmit

Return Value

data length (byte) on success, 0 on fail

Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "\x1E\x07";
spi_setup(0); // configuring SPI
$slen = spi_write(0, $wbuf, 2); // transmitting data to the SPI
if($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()

Transmitting data to the SPI and reading data from the SPI simultaneously

Description

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

Parameters

- `$spi_id`: SPI ID
- `$wbuf`: variable to transmit (string)
- `$rbuf`: variable to save the received data
- `$rlen`: data length to read

Return Value

data length to read/write (byte)

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$wbuf = "\x1E\x07";
$rlen = 0;
spi_setup(0); // configuring SPI
$rlen = spi_write_read(0, $wbuf, $rbuf, 2); // writing and reading to/from the SPI
if($rlen == 2)
    echo "$rbuf\r\n"; // outputting the received data
?>
```

See also

- [sd_340.php](#)
- [spi_setup\(\)](#)
- [spi_ioctl\(\)](#)
- [spi_read\(\)](#)
- [spi_write\(\)](#)

i2c_ioctl()

Configuring or using I2C

Description

int/string i2c_ioctl(int \$i2c_id, string \$cmd)

Parameters

- \$i2c_id: I2C ID
- \$cmd: command for configuring or using

refer to the document named [PHPoC Device Programming Guide for P40](#) for the details about commands available on \$cmd

Return Value

return value for each command

Example

```
<?php
include "/lib/sd_340.php";
i2c_ioctl(0, "set saddr 0x0e"); // configuring I2C slave's device address
i2c_ioctl(0, "set mode sm"); // configuring I2C mode (standard mode)
i2c_ioctl(0, "req read 0"); // reading data from the I2C
i2c_ioctl(0, "req write"); // writing data to the I2C
?>
```

See also

- [sd_340.php](#)
- [i2c_setup\(\)](#)
- [i2c_scan\(\)](#)
- [i2c_read\(\)](#)
- [i2c_write\(\)](#)
- [i2c_write_read\(\)](#)

i2c_setup()

Configuring I2C

Description

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

Parameters

- `$i2c_id`: SPI ID
- `$saddr`: slave's address
- `$mode`: SPI mode(standard mode - sm, fast mode - fm)

I2C mode	notation	clock speed
standard	sm	100Kbps
fast	fm	400Kbps

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
i2c_setup(0, 0x1e);          // configuring I2C slave address (0x0E)
i2c_setup(0, 0x1e, "fm");   // configuring I2C slave address (0x0E) and fast mode
?>
```

See also

- [sd_340.php](#)
- [i2c_ioctl\(\)](#)
- [i2c_scan\(\)](#)
- [i2c_read\(\)](#)
- [i2c_write\(\)](#)
- [i2c_write_read\(\)](#)

i2c_scan()

scanning I2C slaves

Description

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

Parameters

- `$i2c_id`: I2C ID
- `$rw_bit`: read or write bit for scanning operation
- `$len`: data length of read request

Return Value

the number of slaves which have been found

Example

```
<?php
include "/lib/sd_340.php";
$num = i2c_scan(0);           // scanning slaves
echo "$num slave(s) are founded"; // outputting scan result
?>
```

See also

- [sd_340.php](#)
- [i2c_ioctl\(\)](#)
- [i2c_setup\(\)](#)
- [i2c_read\(\)](#)
- [i2c_write\(\)](#)
- [i2c_write_read\(\)](#)

i2c_read()

Receiving data form the I2C

Description

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

Parameters

- `$i2c_id`: I2C ID
- `$rbuf`: variable to save the received data
- `$rlen`: data length to read (byte)

Return Value

data length (byte) on success

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$rlen = 0;
i2c_setup(0, 0x0E);      // configuring I2C
while(1)
{
    $rlen = i2c_read(0, $rbuf, 2); // reading 2 bytes data from the I2C
    if($rlen == 2)
        echo "$rbuf\r\n";      // Outputting the received data
    sleep(1);
}
?>
```

See also

- [sd_340.php](#)
- [i2c_ioctl\(\)](#)
- [i2c_setup\(\)](#)
- [i2c_scan\(\)](#)
- [i2c_write\(\)](#)
- [i2c_write_read\(\)](#)

i2c_write()

Transmitting data to I2C

Description

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

Parameters

- `$i2c_id`: I2C ID
- `$wbuf`: variable to transmit
- `$wlen`: data length to transmit

Return Value

data length (byte) on success, 0 on fail

Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "\x05\x03";
i2c_setup(0, 0x1e);           // configuring I2C
$slen = i2c_write(0, $wbuf, 2); // transmitting data to the I2C
if($slen == 2)
    echo "$slen bytes have been sent\r\n";
?>
```

See also

- [sd_340.php](#)
- [i2c_ioctl\(\)](#)
- [i2c_setup\(\)](#)
- [i2c_scan\(\)](#)
- [i2c_read\(\)](#)
- [i2c_write_read\(\)](#)

i2c_write_read()

Transmitting data to the I2C and reading data from the I2C simultaneously

Description

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

Parameters

- `$i2c_id`: I2C ID
- `$wbuf`: variable to transmit (string)
- `$rbuf`: variable to save the received data
- `$rlen`: data length to read

Return Value

data length to read/write on success, 0 on fail

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
$wbuf = "\x05\x03";
$rlen = 0;
i2c_setup(0, 0x1e); // configuring I2C
$rlen = i2c_write_read(0, $wbuf, $rbuf, 2); // writing and reading to/from the I2C
if($rlen == 2)
    echo "$rbuf"; // outputting the received data
?>
```

See also

- [sd_340.php](#)
- [i2c_ioctl\(\)](#)
- [i2c_setup\(\)](#)
- [i2c_scan\(\)](#)
- [i2c_read\(\)](#)
- [i2c_write\(\)](#)

st_ioctl()

Configuring and Operating the designaed software timer(ST)

Description

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

Parameters

- `$st_id`: Software timer(ST) ID
- `$cmd`: command for configuring or operation

refer to the document named [PHPoC Device Programming Guide for P40](#) for the details about commands available on `$cmd`

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
st_ioctl(0, "set div us"); // Configuring the unit of ST0 to micro second
st_ioctl(0, "set mode free"); // free mode
st_ioctl(0, "set dir up"); // up count
st_ioctl(0, "start"); // start the ST0
sleep(1);
st_ioctl(0, "stop"); // stop the ST0
?>
```

See also

- [sd_340.php](#)
- [st_free_setup\(\)](#)
- [st_free_get_count\(\)](#)
- [st_pwm_setup\(\)](#)
- [st_pwm_width\(\)](#)

st_free_setup()

Run the designated Software Timer(ST) after configuring it to the free mode

Description

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

Parameters

- `$st_id`: Software Timer(ST) ID
- `$div`: unit - sec (second), ms (milli second, default unit), us (micro second)

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
st_free_setup(0); // Configuring ST0 as a free mode and start the timer
st_free_setup(1, "sec"); // Configuring ST1 as a free mode and start the timer
?>
```

See also

- [sd_340.php](#)
- [st_ioctl\(\)](#)
- [st_free_get_count\(\)](#)
- [st_pwm_setup\(\)](#)
- [st_pwm_width\(\)](#)

st_free_get_count()

Reading the designated Software Timer's counter value

Description

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

Parameters

- `$st_id`: Software Timer ID

Return Value

Returns the counter value of the designated Software Timer

Example

```
<?php
include "/lib/sd_340.php";
st_free_setup(0);           // Configuring ST0 as a free mode and start the timer
while(1)
{
    $count = st_free_get_count(0); // Reading the ST0's counter value
    echo "$count\r\n";           // outputting the value
}
?>
```

See also

- [sd_340.php](#)
- [st_ioctl\(\)](#)
- [st_free_setup\(\)](#)
- [st_pwm_setup\(\)](#)
- [st_pwm_width\(\)](#)

st_pwm_setup()

Run the designated Software Timer(ST) after configuring it to the PWM mode

Description

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

Parameters

- `$st_id`: Software Timer ID
- `$pin`: pin number to output signal
- `$width`: pulse HIGH duration
- `$period`: a pulse period
- `$div`: unit - sec (second), ms (milli second, default unit), us (micro second)

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
// Configuring ST0 as period 200us with 50% duty PWM with output signal pin 5 and start it
st_pwm_setup(0, 5, 100, 200);
// Configuring ST1 as period 100us with 10% duty PWM with output signal pin 9 and start it
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()

Changing designated Software Timer's PWM signal

Description

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

Parameters

- `$st_id`: Software Timer ID
- `$width`: pulse HIGH duration
- `$period`: a pulse period

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
// Changing ST0 as period 100us with 1% duty PWM
st_pwm_setup(0, 0, 1, 100);
for($i = 0; $i < 99; $i++)
{
    st_pwm_width(0, 1+$i, 100); // increasing duty cycle by 1%
}
?>
```

See also

- [sd_340.php](#)
- [st_ioctl\(\)](#)
- [st_free_setup\(\)](#)
- [st_free_get_count\(\)](#)
- [st_pwm_setup\(\)](#)

ht_ioctl()

Configuring and Operating the designaed hardware timer(HT)

Description

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

Parameters

- \$ht_id: Hardware Timer ID
- \$cmd: command for configuring and operating

refer to the document named [PHPoC Device Programming Guide for P40](#) for the details about commands available on \$cmd

Return Value

Reterns return value for each command

Example

```
<?php
include "/lib/sd_340.php";
ht_ioctl(0, "set div us");          // unit: micro second
ht_ioctl(0, "set mode output pulse"); // setting the HT0's mode to output pulse mode
ht_ioctl(0, "set count 100 100");   // pulse period: width - 200 us, duty cycle - 50%
ht_ioctl(0, "set repc 2");          // number of pulse outputs - 2
ht_ioctl(0, "start");               // start the HT0
sleep(1);
ht_ioctl(0, "stop");                // stop the HT0
?>
```

See also

- [sd_340.php](#)
- [ht_pwm_width\(\)](#)
- [ht_pwm_setup\(\)](#)

ht_pwm_setup()

Run the designated Hardware Timer(HT) after configuring it to the PWM mode

Description

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

Parameters

- \$ht_id: Hardware Timer ID
- \$width: pulse HIGH duration
- \$period: a pulse period
- \$div: unit - sec (second), ms (milli second), us (micro second, default unit)

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
// Configuring HT0 as period 200us with 50% duty PWM and start it
ht_pwm_setup(0, 100, 200);
// Configuring HT1 as period 200us with 10% duty PWM and start it
ht_pwm_setup(1, 10, 100);
?>
```

See also

- [sd_340.php](#)
- [ht_ioctl\(\)](#)
- [ht_pwm_width\(\)](#)

ht_pwm_width()

Changing designated Hardware Timer's PWM signal

Description

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

Parameters

- `$ht_id`: Hardware Timer ID
- `$width`: pulse HIGH duration
- `$period`: a pulse period

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
ht_pwm_setup(0, 1, 100); // Changing HTO as period 100us with 1% duty PWM
for($i = 0; $i < 99; $i++)
{
    ht_pwm_width(0, 1+$i, 100); // increasing duty cycle by 1%
}
?>
```

See also

- [sd_340.php](#)
- [ht_ioctl\(\)](#)
- [ht_pwm_setup\(\)](#)

adc_setup()

Configuring designated ADC channel

Description

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

Parameters

- `$adc_id`: ADC ID
- `$ch`: ADC channel number

Return Value

none

Example

```
<?php
include "/lib/sd_340.php";
adc_setup(0, 0); // Setting ADC0 to channel 0
adc_setup(1, 5); // Setting ADC1 to channel 5
?>
```

See also

- [sd_340.php](#)
- [adc_in\(\)](#)

adc_in()

Reading the designated ADC

Description

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

Parameters

- \$adc_id: ADC ID
- \$sc: sampling count (default: 1), it samples the number of \$sc and calculates an average and returns it

Return Value

average value after reading the number of the sample count

Example

```
<?php
$in_value = 0;
include "/lib/sd_340.php";
adc_setup(0, 0); // Setting ADC0 to channel 0
adc_setup(1, 1); // Setting ADC1 to channel 1
$value_0 = adc_in(0, 1); // Reading from ADC0
$value_1 = adc_in(1, 50); // Reading from ADC1 (average value after reading 50 times)
?>
```

See also

- [sd_340.php](#)
- [adc_setup\(\)](#)

um_read()

Reading data from the designated UM(volatile memory)

Description

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

Parameters

- `$um_id`: UM ID
- `$offset`: starting position to read
- `$rbuf`: buffer to save data
- `$rlen`: number of bytes to read

Return Value

number of bytes to have been read

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
um_read(0, 0, $rbuf, 10); // Reading 10 bytes from address 0 of the UM0
hexdump($rbuf);          // Outputting the data
?>
```

See also

- [sd_340.php](#)
- [um_write\(\)](#)

um_write()

Writing data to designated UM(volatile memory)

Description

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

Parameters

- \$um_id: UM ID
- \$offset: starting position to write
- \$wbuf: data to write
- \$rlen: number of bytes to write

Return Value

On success, number of bytes to have been written. On fail, 0

Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
$rbuf = "";
um_write(0, 0, $wbuf, 5); // Writing 5 bytes of data to the address 0 of the UM0
um_read(0, 0, $rbuf, 5); // Reading 5 bytes of data from the address 0 of the UM0
hexdump($rbuf);          // Outputting the data
?>
```

See also

- [sd_340.php](#)
- [um_read\(\)](#)

nm_read()

Reading data from the designated NM(Non-volatile Memory)

Description

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

Parameters

- \$nm_id: NM ID
- \$offset: starting position to read
- \$rbuf: buffer to save data
- \$rlen: number of bytes to read

Return Value

number of bytes to have been read

Example

```
<?php
include "/lib/sd_340.php";
$rbuf = "";
nm_read(0, 0, $rbuf, 10); // Reading 10 bytes from address 0 of the NM0
hexdump($rbuf);          // Outputting the data
?>
```

See also

- [sd_340.php](#)
- [nm_write\(\)](#)

nm_write()

Writing data to the designated NM(Non-volatile Memory)

Description

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

Parameters

- `$nm_id`: NM ID
- `$offset`: starting position to write
- `$wbuf`: data to write
- `$rlen`: number of bytes to write

Return Value

On success, number of bytes to have been written. On fail, 0

Example

```
<?php
include "/lib/sd_340.php";
$wbuf = "abcde";
$rbuf = "";
nm_write(0, 0, $wbuf, 5); // Writing 5 bytes of data to the address 0 of the NM0
nm_read(0, 0, $rbuf, 5); // Reading 5 bytes of data from the address 0 of the NM0
hexdump($rbuf);         // Outputting the data
?>
```

See also

- [sd_340.php](#)
- [nm_read\(\)](#)

dns_setup()

Configuring for DNS query

Description

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

Parameters

- `$udp_id`: UDP ID for DNS query
- `$server_addr`: IP address of the DNS server, if it is omitted system's DNS IP address will be used
- `$ip6`: enable or disable IPv6

Return Value

none

Example

```
<?php
include "/lib/sn_dns.php";
dns_setup(0, "192.168.0.1"); // UDP ID 0, DNS server address is 192.168.0.1
dns_setup(1);             // UDP ID 1, system DNS address will be used for the DNS server IP
address
dns_setup(0, "", true);   // enable IPv6
?>
```

See also

- [sn_dns.php](#)
- [dns_send_query\(\)](#)
- [dns_loop\(\)](#)
- [dns_lookup\(\)](#)

dns_send_query()

Sending DNS query to designated DSN server IP address

Description

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

Parameters

- `$name`: host name to find
- `$type`: domain type (RR_A, RR_NS or RR_MX)

Return Value

the length of query on success, 0 on fail

Example

```
<?php
include "/lib/sn_dns.php";
$rr = "";
$name = "www.phpoc.com";
dns_setup(0);           // UDP ID 0, and system DSN server IP addresss
dns_send_query($name, RR_A); // sending query for www.phpoc.com
while(1)
{
    $rr = dns_loop();    // receiving response from the DNS server
    if($rr === false)
        usleep(1000);
    elseif($rr == "")
    {
        echo $name, "rWrWn"; // outputting host name
        break;
    }
    else
    {
        echo "$rrWrWn";     // outputting DNS server's response
        break;
    }
}
?>
```

See also

- [sn_dns.php](#)
- [dns_setup\(\)](#)
- [dns_loop\(\)](#)

- [dns_lookup\(\)](#)
- [RR_A](#)
- [RR_NS](#)
- [RR_MX](#)

dns_loop()

Returns a response from the DNS server

Description

string dns_loop(void)

Parameters

none

Return Value

On success IP address, On fail null string(""), when didn't receive DNS response returns false

Example

```
<?php
include "/lib/sn_dns.php";
$rr = "";
$name = "www.phpoc.com";
dns_setup(0);          // UDP ID 0, and system DSN server IP addresss
dns_send_query($name, RR_A); // sending query for www.phpoc.com
while(1)
{
    $rr = dns_loop();    // receiving response from the DNS server
    if($rr === false)
        usleep(1000);
    elseif($rr == "")
    {
        echo $name, "rWrWn"; // outputting domain name
        break;
    }
    else
    {
        echo "$rrrWrWn";    // outputting DNS server's response
        break;
    }
}
?>
```

See also

- [sn_dns.php](#)
- [dns_setup\(\)](#)
- [dns_send_query\(\)](#)
- [dns_lookup\(\)](#)

- RR_A
- RR_NS
- RR_MX

dns_lookup()

Returns a response from the DNS server after sending DNS query to designated DSN server IP address

Description

string dns_lookup(string \$name, int \$type)

Parameters

- \$name: host name
- \$type: domain type (RR_A, RR_NS or RR_MX)

Return Value

On success IP address, otherwise input host name

Example

```
<?php
include "/lib/sn_dns.php";
dns_setup(0); // UDP 0
$host_name = "www.phpoc.com";
$host_addr = dns_lookup($host_name, RR_A); // sending query for www.phpoc.com
if($host_addr == $host_name)
    echo "$host_name: Not Found\r\n"; // on fail
else
    echo "$host_name: $host_addr\r\n"; // on success, output host's IP address
?>
```

See also

- [sn_dns.php](#)
- [dns_setup\(\)](#)
- [dns_send_query\(\)](#)
- [dns_loop\(\)](#)
- [RR_A](#)
- [RR_NS](#)
- [RR_MX](#)

smtp_setup()

Configuring TCP ID, UDP ID, and DNS server to send emails by using SMTP

Description

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

Parameters

- `$udp_id`: UDP ID for DNS query
- `$tcp_id`: TCP ID for sending email
- `$dns_server`: IP address of the DNS server, if it is omitted system's DNS IP address will be used
- `$ip6`: enable or disable IPv6

Return Value

none

Example

```
<?php
include "/lib/sn_dns.php";    // including DNS library
include "/lib/sn_smtp.php";  // including SMTP library
// UDP 0 for DNS, TCP 1 for sending email, DSN server IP: 192.168.0.1
smtp_setup(0, 1, "192.168.0.1");

// UDP 2 for DNS, TCP 3 for sending email, DSN server IP: system DNS server IP
smtp_setup(2, 3);

// enable IPv6
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()

Setting the SMTP email sender's host name

Description

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

Parameters

- `$hostname`: host name of email sender

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php"; // including DNS library
include_once "/lib/sn_smtp.php"; // including SMTP library
smtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
smtp_account("from_id@from_domain.com", "from_name");
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
// transmitting the email
$msg = smtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($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()

Setting SMTP email sender's email address and name

Description

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

Parameters

- `$email`: sender's email address
- `$name`: sender's name

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php"; // including DNS library
include_once "/lib/sn_smtp.php"; // including SMTP library
smtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
smtp_account("from_id@from_domain.com", "from_name");
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
// transmitting the email
$msg = smtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($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_hostname\(\)](#)
- [smtp_loop\(\)](#)
- [smtp_start\(\)](#)
- [smtp_send\(\)](#)

smtp_loop()

Transmitting email by talking to the SMTP email server

Description

false/string smtp_loop(void)

Parameters

none

Return Value

On success the server's response message, on fail null string(""), false if there's no response

Example

```
<?php
include_once "/lib/sn_dns.php"; // including DNS library
include_once "/lib/sn_smtp.php"; // including SMTP library
smtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
smtp_account("from_id@from_domain.com", "from_name");
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
// preparing to send an email
smtp_start("to_id@to_domain.com", "to_name", $subject, $message);
while(1)
{
    $msg = smtp_loop();
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msg\r\n";
}
?>
```

See also

- [sn_smtp.php](#)
- [sn_dns.php](#)
- [smtp_setup\(\)](#)
- [smtp_hostname\(\)](#)
- [smtp_account\(\)](#)
- [smtp_start\(\)](#)

- [smtp_send\(\)](#)

smtp_send()

Sending an email by SMTP

Description

string smtp_send(string \$rcpt_email, string \$rcpt_name, string \$subject, string \$body)

Parameters

- \$rcpt_email: receiver's email address
- \$rcpt_name: receiver's name
- \$subject: email subject
- \$body: email content

Return Value

response message from the server on success, empty string("") or false on fail

Example

```
<?php
include_once "/lib/sn_dns.php"; // including DNS library
include_once "/lib/sn_smtp.php"; // including SMTP library
smtp_hostname("from_domain.com"); // sender's host name
// sender's email address and name
smtp_account("from_id@from_domain.com", "from_name");
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
// sending an email
$msg = smtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($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_hostname\(\)](#)
- [smtp_account\(\)](#)
- [smtp_start\(\)](#)
- [smtp_loop\(\)](#)

smtp_start()

Preparing to sending email

Description

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

Parameters

- \$rcpt_email: receiver's email address
- \$rcpt_name: receiver's name
- \$subject: email subject
- \$body: email content

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php"; // including DNS library
include_once "/lib/sn_smtp.php"; // including SMTP library
smtp_hostname("from_domain.com"); // sender's host name
// sender's email address and name
smtp_account("from_id@from_domain.com", "from_name");
$subject = "email test from PHPoC";
$message = "Hi PHPoC\r\nThis is PHPoC test email\r\nGood bye\r\n";
// preparing to send
smtp_start("to_id@to_domain.com", "to_name", $subject, $message);
while(1)
{
    $msg = smtp_loop();
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msg\r\n";
}
?>
```

See also

- [sn_smtp.php](#)
- [sn_dns.php](#)
- [smtp_setup\(\)](#)

- `smtp_hostname()`
- `smtp_account()`
- `smtp_send()`
- `smtp_loop()`

esmtp_setup()

Configuring TCP ID, UDP ID, and DNS server to send emails by using ESMTP

Description

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

Parameters

- `$udp_id`: UDP ID for DNS query
- `$tcp_id`: TCP ID for sending email
- `$dns_server`: IP address of the DNS server, if it is omitted system's DNS IP address will be used
- `$ip6`: enable or disable IPv6

Return Value

none

Example

```
<?php
include "/lib/sn_dns.php"; // including DNS library
include "/lib/sn_esmtp.php"; // including ESMTP library
// UDP 0 for DNS, TCP 1 for sending email, DSN server IP: 192.168.0.1
esmtp_setup(0, 1, "192.168.0.1");

// UDP 2 for DNS, TCP 3 for sending email, DSN server IP: system DNS server IP
esmtp_setup(2, 3);

// enable IPv6
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()

Setting the ESMTP email sender's host name

Description

```
void esmtp_hostname(string $hostname)
```

Parameters

- `$hostname`: host name of email sender

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php";    // including DNS library
include_once "/lib/sn_esmtp.php"; // including ESMTP library
esmtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
esmtp_account("from_id@from_domain.com", "from_name");
esmtp_auth("msa_id", "msa_password"); // setting email account and password of the email server
esmtp_msa("smtp.msa_domain", 465); // setting email server host name and port number
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
// Transmitting an email
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($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()

Setting ESMTP email sender's email address and name

Description

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

Parameters

- `$email`: sender's email address
- `$name`: sender's name

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php";    // including DNS library
include_once "/lib/sn_esmtp.php"; // including ESMTP library
esmtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
esmtp_account("from_id@from_domain.com", "from_name");
esmtp_auth("msa_id", "msa_password"); // setting email account and password of the email server
esmtp_msa("smtp.msa_domain", 465); // setting email server host name and port number
$subject = "msa test";
$message = "Hi PHPoC\r\nThis is PHPoC msa test email\r\nGood bye\r\n";
// Transmitting an email
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($msg == "221")
    echo "send mail successful\r\n";
else
    echo "send mail failed\r\n";
?>
```

See also

- [sn_esmtp.php](#)
- [sn_dns.php](#)
- [esmtp_setup\(\)](#)
- [esmtp_hostname\(\)](#)
- [esmtp_auth\(\)](#)
- [esmtp_msa\(\)](#)
- [esmtp_loop\(\)](#)
- [esmtp_start\(\)](#)

- [esmtp_send\(\)](#)

esmtp_auth()

Setting an account ID and password of the outgoing email server

Description

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

Parameters

- `$auth_id`: account of the outgoing email server
- `$auth_pwd`: password of the outgoing email server

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php";      // including DNS library
include_once "/lib/sn_esmtp.php";    // including ESMTP library
esmtp_hostname("from_domain.com");   // setting email sender's host name
// setting email sender's email address and name
esmtp_account("from_id@from_domain.com", "from_name");
esmtp_auth("msa_id", "msa_password"); // setting email account and password of the outgoing
email server
esmtp_msa("smtp.msa_domain", 465);   // setting email server host name and port number
$subject = "msa test";
$message = "Hi PHPoC\r\nThis is PHPoC msa test email\r\nGood bye\r\n";
// Transmitting an email
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($msg == "221")
    echo "send mail successful\r\n";
else
    echo "send mail failed\r\n";
?>
```

See also

- [sn_esmtp.php](#)
- [sn_dns.php](#)
- [esmtp_setup\(\)](#)
- [esmtp_hostname\(\)](#)
- [esmtp_account\(\)](#)
- [esmtp_msa\(\)](#)
- [esmtp_loop\(\)](#)

- [esmtp_start\(\)](#)
- [esmtp_send\(\)](#)

esmtp_msa()

Setting host name and port number of the outgoing email server

Description

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

Parameters

- `$msa_name`: outgoing email server's host name
- `$msa_port`: outgoing email server's port number

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php";      // including DNS library
include_once "/lib/sn_esmtp.php";    // including ESMTP library
esmtp_hostname("from_domain.com");   // setting email sender's host name
// setting email sender's email address and name
esmtp_account("from_id@from_domain.com", "from_name");
esmtp_auth("msa_id", "msa_password"); // setting email account and password of the outgoing
email server
esmtp_msa("smtp.msa_domain", 465);   // setting email server host name and port number
$subject = "msa test";
$message = "Hi PHPoC\r\nThis is PHPoC msa test email\r\nGood bye\r\n";
// Transmitting an email
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($msg == "221")
    echo "send mail successful\r\n";
else
    echo "send mail failed\r\n";
?>
```

See also

- [sn_esmtp.php](#)
- [sn_dns.php](#)
- [esmtp_setup\(\)](#)
- [esmtp_hostname\(\)](#)
- [esmtp_account\(\)](#)
- [esmtp_auth\(\)](#)
- [esmtp_loop\(\)](#)

- `esmtp_start()`
- `esmtp_send()`

esmtp_loop()

Transmitting email by talking to the ESMTP email server

Description

false/string esmtp_loop(void)

Parameters

none

Return Value

On success the server's response message, on fail null string(""), false if there's no response

Example

```
<?php
include_once "/lib/sn_dns.php";    // including DNS library
include_once "/lib/sn_esmtp.php";  // including ESMTP library
esmtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
esmtp_account("from_id@from_domain.com", "from_name");
esmtp_auth("msa_id", "msa_password"); // setting email account and password of the email server
esmtp_msa("smtp.msa_domain", 465);  // setting email server host name and port number
$subject = "msa test";
$message = "Hi PHPoCWrWnThis is PHPoC msa test emailWrWnGood byeWrWn";
// preparing to send
esmtp_start("to_id@to_domain.com", "to_name", $subject, $message);
while(1)
{
    $msg = esmtp_loop();
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msgWrWn";
}
?>
```

See also

- [sn_esmtp.php](#)
- [sn_dns.php](#)
- [esmtp_setup\(\)](#)
- [esmtp_hostname\(\)](#)

- [esmtp_account\(\)](#)
- [esmtp_auth\(\)](#)
- [esmtp_msa\(\)](#)
- [esmtp_start\(\)](#)
- [esmtp_send\(\)](#)

esmtp_start()

Preparing to sending email

Description

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

Parameters

- `$rcpt_email`: receiver's email address
- `$rcpt_name`: receiver's name
- `$subject`: email subject
- `$body`: email content

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php";    // including DNS library
include_once "/lib/sn_esmtp.php"; // including ESMTP library
esmtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
esmtp_account("from_id@from_domain.com", "from_name");
esmtp_auth("msa_id", "msa_password"); // setting email account and password of the email server
esmtp_msa("smtp.msa_domain", 465); // setting email server host name and port number
$subject = "msa test";
$message = "Hi PHPoC\r\nThis is PHPoC msa test email\r\nGood bye\r\n";
// preparing to send
esmtp_start("to_id@to_domain.com", "to_name", $subject, $message);
while(1)
{
    $msg = esmtp_loop();
    if($msg === false)
        usleep(1000);
    elseif($msg == "")
        ;
    else
        echo "$msg\r\n";
}
?>
```

See also

- [sn_esmtp.php](#)
- [sn_dns.php](#)
- [esmtp_setup\(\)](#)
- [esmtp_hostname\(\)](#)
- [esmtp_account\(\)](#)
- [esmtp_auth\(\)](#)
- [esmtp_msa\(\)](#)
- [esmtp_loop\(\)](#)
- [esmtp_send\(\)](#)

esmtp_send()

Sending an email by ESMTP

Description

string esmtp_send(string \$rcpt_email, string \$rcpt_name, string \$subject, string \$body)

Parameters

- \$rcpt_email: receiver's email address
- \$rcpt_name: receiver's name
- \$subject: email subject
- \$body: email content

Return Value

none

Example

```
<?php
include_once "/lib/sn_dns.php";    // including DNS library
include_once "/lib/sn_esmtp.php";  // including ESMTP library
esmtp_hostname("from_domain.com"); // setting email sender's host name
// setting email sender's email address and name
esmtp_account("from_id@from_domain.com", "from_name");
esmtp_auth("msa_id", "msa_password"); // setting email account and password of the outgoing
email server
esmtp_msa("smtp.msa_domain", 465); // setting email server host name and port number
$subject = "msa test";
$message = "Hi PHPoC\r\nThis is PHPoC msa test email\r\nGood bye\r\n";
// Transmitting an email
$msg = esmtp_send("to_id@to_domain.com", "to_name", $subject, $message);
if($msg == "221")
    echo "send mail successful\r\n";
else
    echo "send mail failed\r\n";
?>
```

See also

- [sn_esmtp.php](#)
- [sn_dns.php](#)
- [esmtp_setup\(\)](#)
- [esmtp_hostname\(\)](#)
- [esmtp_account\(\)](#)

- [esmtp_auth\(\)](#)
- [esmtp_msa\(\)](#)
- [esmtp_loop\(\)](#)
- [esmtp_start\(\)](#)

tcp_client()

Attempting to connect to a server

Description

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

Parameters

- `$tcp_id`: TCP ID
- `$addr`: server's IP address to connect
- `$port`: server's port number to connect

Return Value

none

Example

```
<?php
include "/lib/sn_tcp_ac.php";
$addr = "192.168.0.100"; // IP address to connect
$port = 1470; // TCP port number to connect
tcp_client(0, $addr, $port); // Attempting to connect (TCP ID: 0)
sleep(15);
?>
```

See also

- [sn_tcp_ac.php](#)
- [tcp_server\(\)](#)
- [tcp_read\(\)](#)
- [tcp_readn\(\)](#)
- [tcp_write\(\)](#)
- [tcp_txfree\(\)](#)
- [tcp_state\(\)](#)

tcp_server()

Listening a port as a TCP server

Description

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

Parameters

- `$tcp_id`: TCP ID
- `$port`: TCP port number to accept a TCP connection

Return Value

none

Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470; // TCP port number to accept
tcp_server(0, $port); // listening the port for a TCP connection (TCP ID: 0)
sleep(15);
?>
```

See also

- [sn_tcp_ac.php](#)
- [tcp_client\(\)](#)
- [tcp_read\(\)](#)
- [tcp_readn\(\)](#)
- [tcp_write\(\)](#)
- [tcp_txfree\(\)](#)
- [tcp_state\(\)](#)

tcp_read()

Reading data from the designated TCP ID(session)

Description

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

Parameters

- \$tcp_id: TCP session ID
- \$rbuf: buffer to save data
- \$rlen: maximum length to save data, maximum length if it is omitted

Return Value

the number of byte to read

Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port);      // listening the port for a TCP connection (TCP ID: 0)
$rbuf = "";
while(1)
{
    $rlen = tcp_read(0, $rbuf); // Reading TCP receiving buffer
    if($rlen > 0)
        echo "$rbufWrWn";
}
?>
```

See also

- [sn_tcp_ac.php](#)
- [tcp_client\(\)](#)
- [tcp_server\(\)](#)
- [tcp_readn\(\)](#)
- [tcp_write\(\)](#)
- [tcp_txfree\(\)](#)
- [tcp_state\(\)](#)

tcp_readn()

Reading the designated number of byte from the TCP session

Description

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

Parameters

- `$tcp_id`: TCP session ID
- `$rbuf`: buffer to save data
- `$rlen`: the number data byte to read

Return Value

If there are data equal to or more than the `$rlen` in the TCP receiving buffer, it returns the `$rlen`, otherwise returns 0

Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port);          // listening as a server
$rbuf = "";
while(1)
{
    $rlen = tcp_readn(0, $rbuf, 100); // If there are data equal to or more than 100, it copies 100 bytes
of data to the $rbuf and returns 100
    if($rlen > 0)
        echo "$rbuf\r\n";
}
?>
```

See also

- [sn_tcp_ac.php](#)
- [tcp_client\(\)](#)
- [tcp_server\(\)](#)
- [tcp_read\(\)](#)
- [tcp_write\(\)](#)
- [tcp_txfree\(\)](#)
- [tcp_state\(\)](#)

tcp_write()

Trasmitting data to the designated TCP session(ID)

Description

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

Parameters

- `$tcp_id`: TCP session ID to transmit
- `$wbuf`: data to transmit
- `$wlen`: the number of byte to transmit (if it is omitted maximum length of data will be transmitted)

Return Value

On success returns the transmitted bytes, otherwise 0

Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port);      // listening the port for a TCP connection (TCP ID: 0)
$rdbuf = "";
while(1)
{
    $rwlen = tcp_read(0, $rdbuf); // Reading TCP receiving buffer
    if($rwlen > 0)
        tcp_write(0, $rdbuf);    // transmitting the received data
}
?>
```

See also

- [sn_tcp_ac.php](#)
- [tcp_client\(\)](#)
- [tcp_server\(\)](#)
- [tcp_read\(\)](#)
- [tcp_readn\(\)](#)
- [tcp_txfree\(\)](#)
- [tcp_state\(\)](#)

tcp_txfree()

Getting free buffer space of the designated TCP sessions TX buffer

Description

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

Parameters

- \$tcp_id: TCP session ID

Return Value

On success, it returns free TX buffer length of the TCP session, otherwise 0

Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port); // listening the port for a TCP connection (TCP ID: 0)
$rbuf = "";
while(1)
{
    // Reading the TCP data as many as the TCP TX buffer free or less
    $rlen = tcp_read(0, $rbuf, tcp_txfree(0));
    if($rlen > 0)
        echo "$rbuf\r\n";
}
?>
```

See also

- [sn_tcp_ac.php](#)
- [tcp_client\(\)](#)
- [tcp_server\(\)](#)
- [tcp_read\(\)](#)
- [tcp_readn\(\)](#)
- [tcp_write\(\)](#)
- [tcp_state\(\)](#)

tcp_state()

Getting current TCP state of the designated TCP session

Description

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

Parameters

- \$tcp_id: TCP session ID

Return Value

integer number indicating TCP session state (TCP_CLOSED, TCP_CONNECTED or TCP_LISTEN)

Example

```
<?php
include "/lib/sn_tcp_ac.php";
$port = 1470;
tcp_server(0, $port); // listening the port for a TCP connection (TCP ID: 0)
$rbuf = "";
while(tcp_state(0) != TCP_CONNECTED)
;
echo "TCP connected!\r\n";
?>
```

See also

- [sn_tcp_ac.php](#)
- [tcp_client\(\)](#)
- [tcp_server\(\)](#)
- [tcp_read\(\)](#)
- [tcp_readn\(\)](#)
- [tcp_write\(\)](#)
- [tcp_txfree\(\)](#)

ws_setup()

Configuring a websocket and waiting for a connection

Description

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

Parameters

- `$tcp_id`: TCP ID for the websocket
- `$path`: connection path
- `$proto`: protocol
- `$port`: websocket port, 0 if it is omitted

Return Value

none

Example

```
<?php
include "/lib/sn_tcp_ws.php";
ws_setup(0, "my_path", "my_proto"); // Configuring a websocket and waiting for a connection
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        ws_write(0, "hello, world!\r\n"); // Sending a message to the websocket
        sleep(1);
    }
}
?>
```

See also

- [sn_tcp_ws.php](#)
- [ws_read\(\)](#)
- [ws_readn\(\)](#)
- [ws_read_line\(\)](#)
- [ws_write\(\)](#)
- [ws_txfree\(\)](#)
- [ws_state\(\)](#)

ws_read()

Reading data from the designated websocket session

Description

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

Parameters

- `$tcp_id`: TCP ID of the websocket
- `$rbuf`: buffer to save data
- `$rlen`: maximum length to save data, maximum length if it is omitted

Return Value

the number of byte to read

Example

```
<?php
include "/lib/sn_tcp_ws.php";
// configuring a websocket and waiting for a connection
ws_setup(0, "my_path", "my_proto");
$rbuf = "";
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        ws_read(0, $rbuf); // reading data from the socket
        if(strlen($rbuf) > 0)
        {
            hexdump($rbuf);
            $rbuf = "";
        }
        sleep(1);
    }
}
?>
```

See also

- [sn_tcp_ws.php](#)
- [ws_setup\(\)](#)
- [ws_readn\(\)](#)
- [ws_read_line\(\)](#)
- [ws_write\(\)](#)

- [ws_txfree\(\)](#)
- [ws_state\(\)](#)

ws_readn()

Reading the designated number of byte from the websocket session

Description

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

Parameters

- `$tcp_id`: websocket TCP session ID
- `$rbuf`: buffer to save data
- `$rlen`: the number data byte to read

Return Value

If there are data equal to or more than the `$rlen` in the websocket receiving buffer, it returns the `$rlen`, otherwise returns 0

Example

```
<?php
include "/lib/sn_tcp_ws.php";
// configuring a websocket and waiting for a connection
ws_setup(0, "my_path", "my_proto");
$rbuf = "";
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        /* If there are data equal to or more than 100,
        it copies 100 bytes of data to the $rbuf and returns 100 */
        ws_readn(0, $rbuf, 10);
        if(strlen($rbuf) > 0)
        {
            hexdump($rbuf);
            $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()

Reading data every lines (ending with CR+LF) from the websocket session

Description

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

Parameters

- `$tcp_id`: webscoket TCP session ID
- `$rbuf`: buffer to save data

Return Value

the number of byte to read

Example

```
<?php
<?php
include "/lib/sn_tcp_ws.php";
// configuring a websocket and waiting for a connection
ws_setup(0, "my_path", "my_proto");
$rbuf = "";
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        ws_read_line(0, $rbuf); // Read one line of data
        if(strlen($rbuf) > 0)
        {
            hexdump($rbuf);
            $rbuf = "";
        }
        sleep(1);
    }
}
?>
```

See also

- [sn_tcp_ws.php](#)
- [ws_setup\(\)](#)
- [ws_read\(\)](#)
- [ws_readn\(\)](#)
- [ws_write\(\)](#)

- [ws_txfree\(\)](#)
- [ws_state\(\)](#)

ws_write()

Transmitting data to the designated websocket TCP session(ID)

Description

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

Parameters

- `$tcp_id`: websocket TCP session ID
- `$wbuf`: data to transmit
- `$wlen`: the number of byte to transmit (if it is omitted maximum length of data will be transmitted)

Return Value

On success returns the transmitted bytes, otherwise 0

Example

```
<?php
include "/lib/sn_tcp_ws.php";
$rdbuf = "";
// configuring a websocket and waiting for a connection
ws_setup(0, "my_path", "my_proto");
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        $rwlen = ws_read(0, $rdbuf); // reading data from the websocket
        if($rwlen > 0)
            ws_write(0, $rdbuf);    // transmitting the received data
    }
    sleep(1);
}
?>
```

See also

- [sn_tcp_ws.php](#)
- [ws_setup\(\)](#)
- [ws_read\(\)](#)
- [ws_readn\(\)](#)
- [ws_read_line\(\)](#)
- [ws_txfree\(\)](#)
- [ws_state\(\)](#)

ws_txfree()

Getting free buffer space of the designated websocket TCP session's TX buffer

Description

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

Parameters

- \$tcp_id: websocket TCP session ID

Return Value

On success, it returns free TX buffer length of the websocket TCP session, otherwise 0

Example

```
<?php
include "/lib/sn_tcp_ws.php";
$rdbuf = "";
// configuring a websocket and waiting for a connection
ws_setup(0, "my_path", "my_proto");
while(1)
{
    if(ws_state(0) == TCP_CONNECTED)
    {
        // Reading the data as many as the websocket TX buffer free or less
        $rflen = ws_read(0, $rdbuf, ws_txfree(0));
        if($rflen > 0)
            ws_write(0, $rdbuf); // transmitting the received data
    }
    sleep(1);
}
?>
```

See also

- [sn_tcp_ws.php](#)
- [ws_setup\(\)](#)
- [ws_read\(\)](#)
- [ws_readn\(\)](#)
- [ws_read_line\(\)](#)
- [ws_write\(\)](#)
- [ws_state\(\)](#)

ws_state()

Getting current TCP state of the designated websocket TCP session

Description

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

Parameters

- `$tcp_id`: websocket TCP session ID

Return Value

integer number indicating TCP session state (TCP_CLOSED, TCP_CONNECTED or TCP_LISTEN)

Example

```
<?php
include "/lib/sn_tcp_ws.php";
// configuring a websocket and waiting for a connection
ws_setup(0, "my_path", "my_proto");
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()

Closing the MySQL connection

Description

bool mysql_close()

Parameters

none

Return Value

On success true, otherwise false

Example

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

$server_addr = "192.168.0.100"; // IP address of MySQL server
$user_name = "user_id"; // MySQL ID
$password = "password"; // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error()); // print the error message and quit

mysql_close(); // disconnecting from the MySQL server
?>
```

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()

Attempting to connect to the MySQL server with designated information

Description

false/int mysql_connect(string \$server, string \$user_name, string \$raw_pwd)

Parameters

- \$server: MySQL server's IP address
- \$user_name: MySQL account name
- \$raw_pwd: MySQL account password

Return Value

true on success, false on fail

Example

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

$server_addr = "192.168.0.100"; // IP address of MySQL server
$user_name = "user_id"; // MySQL ID
$password = "password"; // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error()); // print the error message and quit

mysql_close(); // disconnecting from the MySQL server
?>
```

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()

Transmitting a SQL query string

Description

bool/string mysql_query(string \$query)

Parameters

- \$query: SQL query string

Return Value

- INSERT, UPDATE, DELETE: true on success, otherwise false
- SELECT, SHOW: response data on success, otherwise false

Example

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

$query_create_db = "CREATE DATABASE testdb;"; // SQL sentence to make a database
$query_drop_db = "DROP DATABASE testdb;";    // SQL sentence to delete a database

$server_addr = "192.168.0.100";           // IP address of MySQL server
$user_name = "user_id";                   // MySQL ID
$password = "password";                   // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                  // print the error message and quit

if(mysql_query($query_create_db) === true) // send a query to make a database
    echo "create db success!\r\n";
else
    exit(mysql_error());                  // print the error message and quit

if(mysql_query($query_drop_db) === true) // send a query to delete the database
    echo "drop db success!\r\n";
else
    exit(mysql_error());                  // print the error message and quit

mysql_close();                            // disconnecting from the MySQL server
?>
```

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()

Request a connection state to the server

Description

bool mysql_ping(void)

Parameters

none

Return Value

On success true, otherwise false

Example

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

$server_addr = "192.168.0.100";    // IP address of MySQL server
$user_name = "user_id";          // MySQL ID
$password = "password";          // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());          // print the error message and quit

if(mysql_ping() === true)
    echo "connection is alive\r\n";
else
    echo "connection is dead\r\n";

mysql_close();                    // disconnecting from the MySQL server
?>
```

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()

Transmitting a MySQL SELECT query

Description

bool/string mysql_select_db(string \$db_name)

Parameters

- \$db_name: DB name to select

Return Value

true on success, otherwise false or response data

Example

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

$db_name = "testdb";                     // database's name to select
$query_create_db = "CREATE DATABASE $db_name;"; // SQL sentence to make a database
$query_drop_db = "DROP DATABASE $db_name;"; // SQL sentence to delete a database

$server_addr = "192.168.0.100";          // IP address of MySQL server
$user_name = "user_id";                  // MySQL ID
$password = "password";                  // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                  // print the error message and quit

if(mysql_query($query_create_db) === false) // send a query to make a database
    exit(mysql_error());                  // print the error message and quit

if(mysql_select_db($db_name) === true)    // select the database
    echo "Database $db_name is selected!\r\n";
else
    exit(mysql_error());                  // print the error message and quit

if(mysql_query($query_drop_db) === false) // send a query to delete the database
    exit(mysql_error());                  // print the error message and quit

mysql_close();                            // disconnecting from the MySQL server
?>
```


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()

Returning error message from the MySQL server

Description

string mysql_error()

Parameters

none

Return Value

If there is error message, it returns error message, otherwise null string

Example

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

$server_addr = "192.168.0.100";    // IP address of MySQL server
$user_name = "user_id";           // MySQL ID
$password = "password";           // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());           // print the error message and quit

mysql_close();                     // disconnecting from the MySQL server
?>
```

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()

Returning error number according to the error message from the MySQL server

Description

```
int mysql_errno()
```

Parameters

none

Return Value

If there is error message, it returns error code, otherwise 0

Example

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

$server_addr = "192.168.0.100";    // IP address of MySQL server
$user_name = "user_id";           // MySQL ID
$password = "password";           // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_errno());           // printing error code and quit

mysql_close();                     // disconnecting from the MySQL server
?>
```

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()

Returning the number of the affected records by the SQL query

Description

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

Parameters

- `$result`: the response message from the DB server

Return Value

the number of the affected record on success, otherwise -1

Example

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

$db_name = "testdb";                     // database's name to select

// query strings creating or deleting DB, inserting or updating records, and so on.
$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";          // IP address of MySQL server
$user_name = "user_id";                 // MySQL ID
$password = "password";                 // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // print the error message and quit

if(mysql_query($query_create_db) === false) // send a query to make a database
    exit(mysql_error());                 // print the error message and quit

if(mysql_select_db($db_name) === false)   // select the database
    exit(mysql_error());                 // print the error message and quit

if(mysql_query($query_create_table) === false) // send a query to make a table
    exit(mysql_error());                 // print the error message and quit
```

```

if(mysql_query($query_insert_1) === false)    // send a query to insert a record
    exit(mysql_error());                    // print the error message and quit

if(mysql_query($query_insert_2) === false)    // send a query to insert a record
    exit(mysql_error());                    // print the error message and quit

$result = mysql_query($query_update);        // send a query to update a record
if($result === false)
    exit(mysql_error());                    // print the error message and quit
else
{
    $affected_rows = mysql_affected_rows($result); // get the number of affected rows
    if($affected_rows > 0)
        echo "$affected_rows record(s) have(has) been updated!";
}

if(mysql_query($query_drop_table) === false) // send a query to delete the table
    exit(mysql_error());                    // print the error message and quit

if(mysql_query($query_drop_db) === false)    // send a query to delete the database
    exit(mysql_error());                    // print the error message and quit

mysql_close();                             // disconnecting from the MySQL server
?>

```

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()

getting the number of rows in result

Description

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

Parameters

- `$result`: The result message from the server

Return Value

number of rows on success, otherwise -1

Example

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

$db_name = "testdb";                     // database's name to select

// query strings creating or deleting DB, inserting or updating records, and so on.
$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";          // IP address of MySQL server
$user_name = "user_id";                 // MySQL ID
$password = "password";                 // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // print the error message and quit

if(mysql_query($query_create_db) === false) // send a query to make a database
    exit(mysql_error());                 // print the error message and quit

if(mysql_select_db($db_name) === false)   // select the database
    exit(mysql_error());                 // print the error message and quit

if(mysql_query($query_create_table) === false) // send a query to make a table
    exit(mysql_error());                 // print the error message and quit
```

```

if(mysql_query($query_insert_1) === false)    // send a query to insert a record
    exit(mysql_error());                      // print the error message and quit

if(mysql_query($query_insert_2) === false)    // send a query to insert a record
    exit(mysql_error());                      // print the error message and quit

$result = mysql_query($query_select);         // send a query to select records
if($result === false)
    exit(mysql_error());                      // print the error message and quit
else
{
    $num_rows = mysql_num_rows($result);      // get the number of rows in result
    if($num_rows > 0)
        echo "$num_rows records has been detected!\n\n";
}

if(mysql_query($query_drop_table) === false) // send a query to delete the table
    exit(mysql_error());                      // print the error message and quit

if(mysql_query($query_drop_db) === false)    // send a query to delete the database
    exit(mysql_error());                      // print the error message and quit

mysql_close();                               // disconnecting from the MySQL server
?>

```

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()

getting a data in the result

Description

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

Parameters

- \$result: result message from the DB server
- \$row: row number
- \$col: colum number, 0 if it is omitted

Return Value

result value on success, otherwise false

Example

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

$db_name = "testdb";                     // database's name to select

// query strings creating or deleting DB, inserting or updating records, and so on.
$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";          // IP address of MySQL server
$user_name = "user_id";                 // MySQL ID
$password = "password";                  // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                 // print the error message and quit

if(mysql_query($query_create_db) === false) // send a query to make a database
    exit(mysql_error());                 // print the error message and quit

if(mysql_select_db($db_name) === false)   // select the database
    exit(mysql_error());                 // print the error message and quit

if(mysql_query($query_create_table) === false) // send a query to make a table
```



```

exit(mysql_error());           // print the error message and quit

if(mysql_query($query_insert_1) === false) // send a query to insert a record
    exit(mysql_error());       // print the error message and quit

if(mysql_query($query_insert_2) === false) // send a query to insert a record
    exit(mysql_error());       // print the error message and quit

$result = mysql_query($query_select);      // send a query to select records
if($result === false)
    exit(mysql_error());                   // print the error message and quit
else
{
    $value = mysql_result($result, 1, 1); // get a data in 1st row in 1st column
    if($value)
        echo "$value\r\n";              // print the data
    else
        echo "No data has been detected!\r\n";
}

if(mysql_query($query_drop_table) === false) // send a query to delete the table
    exit(mysql_error());                   // print the error message and quit

if(mysql_query($query_drop_db) === false) // send a query to delete the database
    exit(mysql_error());                   // print the error message and quit

mysql_close();                          // disconnecting from the MySQL server
?>

```

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()

getting a result row as an enumerated array

Description

array mysql_fetch_row(string \$result)

Parameters

- \$result: the result message from the DB

Return Value

an array string on success, otherwise false

Example

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

$db_name = "testdb";                     // database's name to select

// query strings creating or deleting DB, inserting or updating records, and so on.
$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";          // IP address of MySQL server
$user_name = "user_id";                  // MySQL ID
$password = "password";                  // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error());                  // print the error message and quit

if(mysql_query($query_create_db) === false) // send a query to make a database
    exit(mysql_error());                  // print the error message and quit

if(mysql_select_db($db_name) === false)   // select the database
    exit(mysql_error());                  // print the error message and quit

if(mysql_query($query_create_table) === false) // send a query to make a table
    exit(mysql_error());                  // print the error message and quit
```

```

if(mysql_query($query_insert_1) === false)    // send a query to insert a record
    exit(mysql_error());                    // print the error message and quit

if(mysql_query($query_insert_2) === false)    // send a query to insert a record
    exit(mysql_error());                    // print the error message and quit

$result = mysql_query($query_select);        // send a query to select records
if($result === false)
    exit(mysql_error());                    // print the error message and quit
else
{
    while(1)
    {
        $arr = mysql_fetch_row($result);    // get a record
        if($arr === false)
            break;
        $arr_len = count($arr);            // the number of column
        for($i = 0; $i < $arr_len; $i++)
        {
            if($arr[$i] !== "")
                printf("%s ", $arr[$i]);    // print a data
            else
                break;
        }
        echo "WrWn";
    }
}

if(mysql_query($query_drop_table) === false) // send a query to delete the table
    exit(mysql_error());                    // print the error message and quit

if(mysql_query($query_drop_db) === false)    // send a query to delete the database
    exit(mysql_error());                    // print the error message and quit

mysql_close();                             // disconnecting from the MySQL server
?>

```

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()

configuring for using MySQL

Description

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

Parameters

- `$udp_id`: UDP ID number for DNS
- `$tcp_id`: TCP ID number for MySQL
- `$dns_server`: DNS server IP address to quering DNS
- `$ip6`: enable or disable IPv6

Return Value

none

Example

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

mysql_setup(4, 4, "", false); // configuring TCP/IP ID for MySQL and DNS

$server_addr = "192.168.0.100"; // IP address of MySQL server
$user_name = "user_id"; // MySQL ID
$password = "password"; // MySQL password

// connect to a MySQL server
if(mysql_connect($server_addr, $user_name, $password) === false)
    exit(mysql_error()); // print the error message and quit

mysql_close(); // disconnecting from the MySQL server
?>
```

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()

Initialize SPC

Description

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

Parameters

- \$t1 : output duration of reset signal (millisecond)
- \$t2 : waiting time for reset (millisecond)

Return Value

none

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()

Synchronize buadrate with smart expansion boards

Description

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

Parameters

- \$baud : baudrate (bps)
- \$t1 : output duration of synchronization signal (millisecond)
- \$t2 : waiting time for synchronization (millisecond)

Return Value

none

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()

Scan smart expansion boards connected

Description

```
int spc_scan(int $start = 1, int $end = 14, bool $verbose = false)
```

Parameters

- \$start: start id
- \$end: end id
- \$verbose: print result of scanning boards or not (true: print, false: do not print)

Return Value

the number of smart expansion boards detected

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!\n\n";
?>
```

See also

- [sd_spc.php](#)
- [spc_reset\(\)](#)
- [spc_sync_baud\(\)](#)
- [spc_ioctl\(\)](#)
- [spc_request\(\)](#)
- [spc_request_sys\(\)](#)
- [spc_request_dev\(\)](#)

spc_ioctl()

Configuring or using SPC

Description

int/string spc_ioctl(string \$cmd)

Parameters

- \$cmd: commnad for configuring or using SPC

Return Value

return value for each command

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()

request sending a command to smart expansion boards

Description

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

Parameters

- \$sid: slave id
- \$addr: command address
- \$msg: a command string
- \$opt: option depends on \$msg

Return Value

a response string in CSV format from smart expansion boards on success, false on failure

※ CSV : Comma-Separated Values

Example

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

spc_reset();
spc_sync_baud();

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()

request sending a common system command to smart expansion boards

Description

string spc_request_sys(int \$sid, string \$cmd)

Parameters

- \$sid: slave id
- \$cmd: a command string

Return Value

a response string according to each command

Example

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

spc_reset();
spc_sync_baud();
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()

request sending a device command to smart expansion boards

Description

string spc_request_dev(int \$sid, string \$cmd)

Parameters

- \$sid: slave id
- \$cmd: a command string

Return Value

a response string according to each command

Example

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

spc_reset();
spc_sync_baud();

$sid = 1;
spc_request_dev($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

LOW is a constant to show the logical 0.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)
- [sd_340.php](#)

HIGH

HIGH is a constant to show the logical 1.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)
- [sd_340.php](#)

TOGGLE

TOGGLE is a constant to toggle a digital signal level.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)
- [sd_340.php](#)

LED_A

LED_A is a constant to define A LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

LED_B

LED_B is a constant to define B LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

LED_C

LED_C is a constant to define C LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

LED_D

LED_D is a constant to define D LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

LED_E

LED_E is a constant to define E LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

LED_F

LED_F is a constant to define F LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

LED_G

LED_G is a constant to define G LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

LED_H

LED_H is a constant to define H LED on external products.

Related Library

- [sd_101.php](#)
- [sd_104.php](#)
- [sd_204.php](#)

DO_0

DO_0 is a constant to define the number 0 of digital output ports

Related Library

- [sd_204.php](#)

DO_1

DO_1 is a constant to define the number 1 of digital output ports

Related Library

- [sd_204.php](#)

DO_2

DO_2 is a constant to define the number 2 of digital output ports

Related Library

- [sd_204.php](#)

DO_3

DO_3 is a constant to define the number 3 of digital output ports.

Related Library

- [sd_204.php](#)

DI_0

DI_0 is a constant to define the number 0 of digital input ports.

Related Library

- [sd_204.php](#)

DI_1

DI_1 is a constant to define the number 1 of digital input ports.

Related Library

- [sd_204.php](#)

DI_2

DI_2 is a constant to define the number 2 of digital input ports.

Related Library

- [sd_204.php](#)

DI_3

DI_3 is a constant to define the number 3 of digital input ports.

Related Library

- [sd_204.php](#)

RR_A

RR_A is a constant to show the host address among the DNS types.

Related Library

- [sn_dns.php](#)

RR_AAAA

RR_AAAA is a constant to show the IPv6 address among the DNS types.

Related Library

- [sn_dns.php](#)

RR_NS

RR_MX is a constant to show the mail exchange among the DNS types.

Related Library

- [sn_dns.php](#)

RR_MX

RR_NS is a constant to show the authoritative name server among the DNS types.

Related Library

- [sn_dns.php](#)