

PHPoC Support Package (PSP)

Library Reference

sd_101.php.....	3
sd_104.php.....	4
sd_204.php.....	5
sn_dns.php	6
sn_esmtp.php	7
sn_smtp.php	8
sn_tcp_ac.php	9
sn_tcp_ws.php	10
sn_mysql.php.....	11

sd_101.php	PBH-101 기본 라이브러리
정의 된 상수	LED 포트: LED_A/B/C/D/E/F/G/H LED 출력 유형: LOW/HIGH/TOGGLE
정의 된 함수	<div>void led_setup(int \$pin, string \$mode) : 지정 된 LED 포트의 유형을 설정</div> <div>void led_out(int \$pin, int \$type) : 지정 된 LED 포트에 지정 된 유형을 출력(LOW, HIGH 또는 TOGGLE)</div> <div>int led_in(int \$pin) : 지정 된 LED 포트의 상태를 반환(HIGH 또는 LOW)</div> <div>void uart_setup(int \$uart_id, int \$baud [, string \$ctrl = "N81N"]) : 지정 된 시리얼포트의 통신속도, 패리티비트, 데이터비트, 정지비트 및 흐름제어를 설정</div> <div>int uart_read(int \$uart_id, string &\$rbuf [, int \$rlen = MAX_STRING_LEN]) : 지정 된 시리얼포트로 수신 된 데이터를 읽음</div> <div>int uart_readn(int \$uart_id, string &\$rbuf, int \$rlen) : 지정 된 시리얼포트로 지정 된 크기 이상의 데이터가 수신 되면 읽음</div> <div>int uart_write(int \$uart_id, int/string \$wbuf[, int \$wlen = MAX_STRING_LEN]) : 지정 된 시리얼포트로 데이터를 전송</div> <div>int uart_txfree(int \$uart_id) : 지정 된 시리얼포트의 송신버퍼 여유공간을 반환</div> <div>void st_ioctl(int \$st_id, string \$cmd) : 지정 된 ST(타이머)의 설정 및 사용</div> <div>void st_free_setup(int \$st_id [, string \$div = "ms"]) : 지정 된 ST(타이머)를 프리모드로 설정</div> <div>int st_free_get_count(int \$st_id) : 지정 된 프리모드 ST(타이머) 값을 읽음</div>

sd_104.php	PBH-104 기본 라이브러리
정의 된 상수	LED 포트: LED_A/B/C/D/E/F/G/H
	LED 출력 유형: LOW/HIGH/TOGGLE
정의 된 함수	void led_setup(int \$pin, string \$mode) : 지정 된 LED 포트의 유형을 설정
	void led_out(int \$pin, int \$type) : 지정 된 LED 포트에 지정 된 유형을 출력(LOW, HIGH 또는 TOGGLE)
	int led_in(int \$pin) : 지정 된 LED 포트의 상태를 반환(HIGH 또는 LOW)
	void uart_setup(int \$uart_id, int \$baud [, string \$ctrl = "N81N"]) : 지정 된 시리얼포트의 통신속도, 패리티비트, 데이터비트, 정지비트 및 흐름제어를 설정
	int uart_read(int \$uart_id, string &\$rbuf [, int \$rlen = MAX_STRING_LEN]) : 지정 된 시리얼포트로 수신 된 데이터를 읽음
	int uart_readn(int \$uart_id, string &\$rbuf, int \$rlen) : 지정 된 시리얼포트로 지정 된 크기 이상의 데이터가 수신 되면 읽음
	int uart_write(int \$uart_id, int/string \$wbuf[, int \$wlen = MAX_STRING_LEN]) : 지정 된 시리얼포트로 데이터를 전송
	int uart_txfree(int \$uart_id) : 지정 된 시리얼포트의 송신버퍼 여유공간을 반환
	void st_ioctl(int \$st_id, string \$cmd) : 지정 된 ST(타이머)의 설정 및 사용
	void st_free_setup(int \$st_id [, string \$div = "ms"]) : 지정 된 ST(타이머)를 프리모드로 설정
	int st_free_get_count(int \$st_id) : 지정 된 프리모드 ST(타이머) 값을 읽음

sd_204.php	PBH-204 기본 라이브러리
정의 된 상수	LED 포트: LED_A/B/C/D/E/F/G/H
	IO 포트: DO_0/1/2/3, DI_0/1/2/3
	LED/IO 출력 유형: LOW/HIGH/TOGGLE
정의 된 함수	void led_setup(int \$pin, string \$mode) : 지정 된 LED 포트의 유형을 설정
	void led_out(int \$pin, int \$type) : 지정 된 LED 포트에 지정 된 유형을 출력(LOW, HIGH 또는 TOGGLE)
	int led_in(int \$pin) : 지정 된 LED 포트의 상태를 반환(HIGH 또는 LOW)
	void dio_out(int \$pin, int \$type) : 지정 된 DO 포트에 지정 된 유형을 출력(LOW, HIGH 또는 TOGGLE)
	int dio_in(int \$pin) : 지정 된 DI 포트의 상태를 반환(HIGH 또는 LOW)
	void uart_setup(int \$uart_id, int \$baud [, string \$ctrl = "N81N"]) : 지정 된 시리얼포트의 통신속도, 패리티비트, 데이터비트, 정지비트 및 흐름제어를 설정
	int uart_read(int \$uart_id, string &\$rbuf [, int \$rlen = MAX_STRING_LEN]) : 지정 된 시리얼포트로 수신 된 데이터를 읽음
	int uart_readn(int \$uart_id, string &\$rbuf, int \$rlen) : 지정 된 시리얼포트로 지정 된 크기 이상의 데이터가 수신 되면 읽음
	int uart_write(int \$uart_id, int/string \$wbuf[, int \$wlen = MAX_STRING_LEN]) : 지정 된 시리얼포트로 데이터를 전송
	int uart_txfree(int \$uart_id) : 지정 된 시리얼포트의 송신버퍼 여유공간을 반환
	void st_ioctl(int \$st_id, string \$cmd) : 지정 된 ST(타이머)의 설정 및 사용
	void st_free_setup(int \$st_id [, string \$div = "ms"]) : 지정 된 ST(타이머)를 프리모드로 설정
	int st_free_get_count(int \$st_id) : 지정 된 프리모드 ST(타이머) 값을 읽음

sn_dns.php	DNS(Domain Name System) 라이브러리
정의 된 함수	void dns_setup(int \$udp_id [, string \$server_addr = ""]) : DNS 수행을 위한 UDP 아이디 및 DNS 서버 주소 설정
	void dns_send_query(string \$name, int \$type) : 지정 된 호스트이름과 유형으로 DNS 쿼리를 보냄
	string dns_loop(void) : DNS쿼리에 대한 응답을 받아 결과를 반환
	string dns_lookup(string \$name, int \$type) : 지정 된 호스트이름과 유형으로 DNS 쿼리를 보내고 서버로부터 응답을 받아 결과를 반환

sn_esmtp.php	ESMTP(Extended Simple Mail Transfer Protocol) 라이브러리
정의 된 함수	void esmtp_setup(\$udp_id, \$tcp_id [, string \$dns_server = ""]) : Extended SMTP 수행을 위한 UDP/TCP 아이디 및 DNS 서버주소 설정
	void esmtp_hostname(string \$hostname) : 보내는 PHPoC 제품의 호스트이름을 설정
	void esmtp_account(string \$email, string \$name) : 보내는 메일주소와 이름을 설정
	void esmtp_auth(string \$auth_id, string \$auth_pwd) : 보내는 메일 서버의 계정정보를 설정
	void esmtp_msa(string \$msa_name, int \$msa_port) : 보내는 메일서버 이름과 포트를 설정
	false/string esmtp_loop(void) : 메일서버의 응답을 받아 메일전송과정을 진행
	void esmtp_start(string \$rcpt_email, string \$rcpt_name, string \$subject, string \$body) : 메일을 보내기 위해 필요한 항목들을 설정하고 준비
	string esmtp_send(string \$rcpt_email, string \$rcpt_name, string \$subject, string \$body) : 메일을 보내기 위한 설정 후 메일전송과정을 진행

sn_smtp.php	SMTP(Simple Mail Transfer Protocol) 라이브러리
정의 된 함수	void smtp_setup(\$udp_id, \$tcp_id [, string \$dns_server = ""]) : SMTP 수행을 위한 UDP/TCP 아이디 및 DNS 서버 주소 설정
	void smtp_hostname(string \$hostname) : 보내는 PHPoC제품의 호스트이름을 설정
	void smtp_account(string \$email, string \$name) : 보내는 메일주소와 이름을 설정
	false/string smtp_loop(void) : 메일서버의 응답을 받아 메일전송과정을 진행
	void smtp_start(string \$rcpt_email, string \$rcpt_name, string \$subject, string \$body) : 메일을 보내기 위해 필요한 항목들을 설정하고 준비
	string smtp_send(string \$rcpt_email, string \$rcpt_name, string \$subject, string \$body) : 메일을 보내기 위한 설정 후 메일전송과정을 진행

sn_tcp_ac.php	TCP 라이브러리
정의 된 함수	void tcp_client(int \$tcp_id, string \$addr, int \$port) : 지정 된 TCP를 TCP클라이언트로 접속 시도(IP주소, 포트번호 지정)
	void tcp_server(int \$tcp_id, int \$port) : 지정 된 TCP를 TCP서버로 접속 대기(포트번호 지정)
	int tcp_read(int \$tcp_id, string &\$rbuf [, int \$rlen = MAX_STRING_LEN]) : 선택 된 TCP소켓의 수신 데이터를 읽음
	int tcp_readn(int \$tcp_id, string &\$rbuf, int \$rlen) : 선택 된 TCP소켓의 수신 데이터가 특정 크기 이상이 되면 읽음
	int tcp_write(int \$tcp_id, int/string \$wbuf [, int \$wlen = MAX_STRING_LEN]) : 선택 된 TCP소켓으로 데이터 전송
	int tcp_txfree(int \$tcp_id) : 선택 된 TCP소켓의 송신버퍼 여유공간을 반환
	int tcp_state(int \$tcp_id) : 선택 된 TCP소켓의 상태를 반환

sn_tcp_ws.php	웹 소켓(Web Socket) 라이브러리
정의 된 함수	void ws_setup(int \$tcp_id, string \$path, string \$proto [, int \$port = 0]) : 지정 된 TCP를 웹 소켓 서버로 설정(경로, 프로토콜 및 포트 지정)
	int ws_read(int \$tcp_id, string &\$rbuf [, int \$rlen = MAX_STRING_LEN]) : 선택 된 웹 소켓의 수신 데이터를 읽음
	int ws_readn(int \$tcp_id, string &\$rbuf, int \$rlen) : 선택 된 웹 소켓의 수신 데이터가 특정 크기 이상이 되면 읽음
	int ws_read_line(int \$tcp_id, string &\$rbuf) : 선택 된 웹 소켓의 수신 데이터를 한 줄 읽음(CR+LF로 구분)
	int ws_write(int \$tcp_id, int/string \$wbuf [, int \$wlen = MAX_STRING_LEN]) : 선택 된 웹 소켓으로 데이터 전송
	int ws_txfree(int \$tcp_id) : 선택 된 웹 소켓의 송신버퍼 여유공간을 반환
	int ws_state(int \$tcp_id) : 선택 된 웹 소켓의 상태를 반환

sn_mysql.php	MYSQL 라이브러리
정의 된 함수	bool mysql_close(int \$tcp_id) : MYSQL접속을 종료
	false/int mysql_connect(string \$server, string \$user_name, string \$raw_pwd [, int \$tcp_id = 3]) : 지정 된 계정정보로 MYSQL DB서버로 접속 시도
	bool/string mysql_query(string \$query) : SQL 쿼리 전송
	bool/string mysql_select_db(string \$db_name) : SELECT DB 쿼리 전송
	string mysql_error(string \$result) : DB서버의 에러응답에 대한 메시지 반환
	int mysql_errno(string \$result) : DB서버의 에러응답에 대한 코드 반환
	int mysql_affected_rows(string \$result) : SQL쿼리에 의해 변경 된 레코드의 수 반환
	int mysql_num_rows(string \$result) : SQL쿼리의 결과 레코드 수 반환
	string mysql_result(string \$result, int \$row [, int \$col = 0]) : 특정 행, 열에 해당하는 결과 레코드를 반환
	array mysql_fetch_row(string \$result) : SQL쿼리의 결과 레코드를 배열형태로 반환