

PHE-32P Stand Alone Control API

Description

Revision 1.0, May 2004

Rev. No.	Description	Date	Approved
1.0	Initial	May/31/04	Vincent

PHE-32P Stand Alone Control API Description

API Name: PHE32_INIT

int PHE32_INIT();

PHE-32P Initialize. Need to be called before all other API can be operated properly. The routine will scan all the available PHE-32P modules on PCI bus, and set the PHE-32P module in Software control mode.

Return Value:

0 if success

API Name: PHE32_SETADDR

int PHE32_SETADDR(int module_addr);

Select PHE-32P module. When more than one PHE-32P modules are used, we need to select one of the PHE-32P for power/driver control.

Argument:

int module_addr: The address setting on PHE-32P Module, Range: 0 ~ 3

Return Value:

0 if success

Example:

PHE32_SETADDR(0); select the PHE-32P with Module Address set at “0”

PHE32_POFF(); turn off the power of PHE-32P with Module Address set at “0”

PHE32_SETADDR(1); select the PHE-32P with Module Address set at “1”

PHE32_POFF(); turn off the power of PHE-32P with Module Address set at “1”

API Name: PHE32_PON

int PHE32_PON();

Turn Power On for the card riding on PHE-32C.

Return Value:

0 if success

API Name: PHE32_POFF

int PHE32_POFF();

Turn Power Off for the card riding on PHE-32C.

Return Value:

0 if success

API Name: PHE32_WINEN

int PHE32_WINEN(int delaytime);

Enable Windows Drivers for the card riding on PHE-32C.

Argument:

int delaytime: Delay time between every drivers being enabled

Return Value:

0 if success

API Name: PHE32_WINDIS

int PHE32_WINDIS(int delaytime);

Dsiable Windows Drivers for the card riding on PHE-32C.

Argument:

int delaytime: Delay time between every drivers being disabled

Return Value:

0 if success

API Name: PHE32_CHECKCFG

int PHE32_CHECKCFG(char* logfile);

Check PCI configuration for the device riding on PHE-32 Module. Users need to setup the checking file through GUI.

Note: GUI will be released before April 2004.

Argument:

char* logfile: log file of checking if specified

Return Value:

0 if success

Example:

PHE32_CHECKCFG("test.log");

Check the PCI configuration register on the PHE-32 Module, and write the result to "test.log".

PHE32_CHECKCFG(NULL);

Check the PCI configuration register on the PHE-32 Module.

API Name: PHE32_WAITCARDREMOVE

int PHE32_WAITCARDREMOVE();

Wait until the card is removed from the PHE-32 Module.

Return Value:

0 if success

API Name: PHE32_WAITCARDPLUG

int PHE32_WAITCARDPLUG();

Wait until the card is plugged on the PHE-32 Module.

Return Value:

0 if success

API Name: PHE32_EXIT

int PHE32_EXIT();

Reset the PHE-32P Module from Software Control Mode and Remove opened handle and free allocated memory.

Return Value:

0 if success

API Name: PHE32_LEDGO

int PHE32_LEDGO();

Turn on the “GO” LED on the PHE-32P Module.

Return Value:

0 if success

API Name: PHE32_LEDNG

int PHE32_LEDNG();

Turn on the “NG” LED on the PHE-32P Module.

Return Value:

0 if success

API Name: PHE32_LED OFF

int PHE32_LED OFF();

Turn off the “NG” and “GO” LED on the PHE-32P Module.

Return Value:

0 if success

API Name: PHE32_ BEEP

int PHE32_ BEEP(int freq, int time);

Turn on the “Beeper” on the PHE-32P Module.

Argument:

int freq: 0~4 (Lower-Frequency to Higher Frequency)

int time: time duration for beeping.

Return Value:

0 if success

API Name: PHE32_ BEEP2

int PHE32_ BEEP2(int freq, int time);

Turn on the “Beeper” twice on the PHE-32P Module.

Argument:

int freq: 0~4 (Lower-Frequency to Higher Frequency)

int time: time duration for beeping.

Return Value:

0 if success

VC++ Example Code 1: PHE32Test**Description:**

Call each API Individually. Users can run this code to understand the APIs.

```

//*****
//   File: PhepTest.cpp
//   Purpose: PHE-32P Stand Alone DLL Test Program
//
//   Author : Vincent Yang, Soliton Tech. Co. LTD.
//   Date : Mar/15/2004
//   Modification :
//   v1.00.00 : Initial
//
//*****
#include "stdafx.h"
#include "phe32ldll.h"
#include "conio.h"
int main(int argc, char* argv[])
{
    int status;

    PHE32_INIT();

    char ch;

    do
    {
        printf("Press Command\n");
        printf("1 : Power on\n");
        printf("2 : Power off\n");
        printf("3 : Window Enable\n");
        printf("4 : Window Disable\n");
        printf("5 : Wait Card Remove\n");
        printf("6 : Wait Card Plug\n");
        printf("7 : LED Go\n");
        printf("8 : LED NG\n");
        printf("9 : LED Off\n");
        printf("a : Beep 1\n");
        printf("b : Beep 2\n");
        printf("q : Quit\n");
        status = 0;
        ch = getch();
        switch(ch)
        {
            case '1':
                status = PHE32_PON();
                break;
            case '2':
                status = PHE32_POFF();
                break;
            case '3':
                status = PHE32_WINEN(500);
                break;
            case '4':
                status = PHE32_WINDIS(100);
                break;
            case '5':
                status = PHE32_WAITCARDREMOVE();
                break;
            case '6':
                status = PHE32_WAITCARDPLUG();
                break;
        }
    }
}

```

```
        case '7':
            status = PHE32_LEDGO();
            break;
        case '8':
            status = PHE32_LEDNG();
            break;
        case '9':
            status = PHE32_LEDOFF();
            break;
        case 'a':
            status = PHE32_BEEP(1, 150);
            break;
        case 'b':
            status = PHE32_BEEP2(1, 150);
            break;
        default:
            break;
    }
    printf("Status = %d\n", status);
} while(ch != 'q');
PHE32_EXIT();
return 0;
}
```

VC++ Example Code 2: PhepGUI

Using PHE-32P stand alone API in MFC. Please refer to source code in “PhepGUI” directory.