

# کار با پرت پارالل

با سلام  
برای کنترل ربات با کامپیوتر باید از پرت ها برای برقراری آنها با کامپیوتر استفاده کرد .  
چون پرت پارالل دارای سرعت خوب و کارایی زیاد است من در این مقاله می خواهم طرز  
نوشتن رو این پرت را به شما یاد دهم .

خوب حالا کد زیر در سی پلاس پلاس (تحت ویندوز) تایپ کرده و آن را DLL کنید .

```
// inpout32drv.cpp : Defines the entry point for the DLL application.
//

#include "stdafx.h"
#include "hwinterfacedrv.h"
#include "resource.h"
#include "conio.h"
#include "stdlib.h"

void _stdcall Out32(short PortAddress, short data);
short _stdcall Inp32(short PortAddress);
int inst();
int start();

char str[10];
int vv;

HANDLE hdriver;
char path[MAX_PATH];
HINSTANCE hmodule;
SECURITY_ATTRIBUTES sa;
int sysver;

int Opendriver(void);
void Closedriver(void);

BOOL APIENTRY DllMain( HINSTANCE hModule,
                      DWORD ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    hmodule = hModule;
    switch(ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        sysver = SystemVersion();
        if(sysver==2)
        {
            Opendriver();
        }
        break;
    case DLL_PROCESS_DETACH:
        if(sysver==2)
        {
            CloseHandle(hdriver);
        }
        break;
    }
    return TRUE;
}

/*****/

void Closedriver(void)
```

```

{
    CloseHandle(hdriver);
}

void _stdcall Out32(short PortAddress, short data)
{
    switch(sysver)
    {
    case 1:
        _outp( PortAddress,data);
        break;

    case 2:
        unsigned int error;
        DWORD BytesReturned;
        BYTE Buffer[3];
        unsigned short * pBuffer;
        pBuffer = (unsigned short *)&Buffer[0];
        *pBuffer = LOWORD(PortAddress);
        Buffer[2] = LOBYTE(data);

        error = DeviceIoControl(hdriver,
            IOCTL_WRITE_PORT_UCHAR,
            &Buffer,
            3,
            NULL,
            0,
            &BytesReturned,
            NULL);

        break;
    }
}

}

/*****

short _stdcall Inp32(short PortAddress)
{
    BYTE retval;
    switch(sysver)
    {

    case 1:
        retval = _inp(PortAddress);
        return retval;
        break;

    case 2:
        unsigned int error;
        DWORD BytesReturned;
        unsigned char Buffer[3];
        unsigned short * pBuffer;
        pBuffer = (unsigned short *)&Buffer;
        *pBuffer = LOWORD(PortAddress);
        Buffer[2] = 0;
        error = DeviceIoControl(hdriver,
            IOCTL_READ_PORT_UCHAR,
            &Buffer,
            2,
            &Buffer,
            1,
            &BytesReturned,
            NULL);

        return((int)Buffer[0]);

        break;
    }
    return 0;
}

/*****

int Opendriver(void)

```

```

{
    hdriver = CreateFile("\\\\.\\hwinterface",
        GENERIC_READ | GENERIC_WRITE,
        0,
        NULL,
        OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL,
        NULL);

    if(hdriver == INVALID_HANDLE_VALUE)
    {

        if(start())
        {
            inst();
            start();

            hdriver = CreateFile("\\\\.\\hwinterface",
                GENERIC_READ | GENERIC_WRITE,
                0,
                NULL,
                OPEN_EXISTING,
                FILE_ATTRIBUTE_NORMAL,
                NULL);

            }

        return 1;
    }
return 0;
}

/*****

int inst()
{

    SC_HANDLE Mgr;
    SC_HANDLE Ser;

    GetSystemDirectory(path , sizeof(path));
    HRSRC hResource = FindResource(hmodule, MAKEINTRESOURCE(IDR_BIN1), "bin");
    if(hResource)
    {
        HGLOBAL binGlob = LoadResource(hmodule, hResource);

        if(binGlob)
        {
            void *binData = LockResource(binGlob);

            if(binData)
            {
                HANDLE file;
                strcat(path, "\\Drivers\\hwinterface.sys");

                file = CreateFile(path,
                                GENERIC_WRITE,
                                0,
                                NULL,
                                CREATE_ALWAYS,
                                0,
                                NULL);

                if(file)
                {
                    DWORD size, written;

                    size = SizeofResource(hmodule, hResource);
                    WriteFile(file, binData, size, &written, NULL);
                    CloseHandle(file);
                }
            }
        }
    }
}

```

```

    }
}

Mgr = OpenSCManager (NULL, NULL,SC_MANAGER_ALL_ACCESS);
if (Mgr == NULL)
{
    //No permission to create
    service
        if (GetLastError() == ERROR_ACCESS_DENIED)
        {
            return 5; // error access denied
        }
        else
        {
            Ser = CreateService (Mgr,
                "hwinterface",
                "hwinterface",
                SERVICE_ALL_ACCESS,
                SERVICE_KERNEL_DRIVER,
                SERVICE_SYSTEM_START,
                SERVICE_ERROR_NORMAL,
                "System32\\Drivers\\hwinterface.sys",
                NULL,
                NULL,
                NULL,
                NULL,
                NULL
            );

        }

    CloseServiceHandle(Ser);
    CloseServiceHandle(Mgr);

    return 0;
}
/*****/
int start(void)
{
    SC_HANDLE Mgr;
    SC_HANDLE Ser;

    Mgr = OpenSCManager (NULL, NULL,SC_MANAGER_ALL_ACCESS);

    if (Mgr == NULL)
    {
        //No permission to create
        service
            if (GetLastError() == ERROR_ACCESS_DENIED)
            {
                Mgr = OpenSCManager (NULL, NULL,GENERIC_READ);
                Ser = OpenService(Mgr,"hwinterface",GENERIC_EXECUTE);
                if (Ser)
                { // we have permission to start the service
                    if(!StartService(Ser,0,NULL))
                    {
                        CloseServiceHandle (Ser);
                        return 4; // we could open the service but unable to start
                    }
                }
            }
        }
    }
    else
    { // Successfully opened Service Manager with full access
        Ser = OpenService(Mgr,"hwinterface",GENERIC_EXECUTE);
        if (Ser)
        {

```

```

        if(!StartService(Ser,0,NULL))
        {
            CloseServiceHandle (Ser);
            return 3; // opened the Service handle with full access
        }
        else
        {
            CloseServiceHandle (Ser);
            return 0;
        }
    }
}

return 1;
}

```

permission, but unable to start

## آزمایش عملی در C++Builder

برای نوشتن روی پرت پارالل در C++Builder یک پروژه جدید باز کنید .  
 آن را ذخیره کنید .  
 Dll ساخته شده را در محل پروژه بریزید .  
 حالا پروژه رو باز کرده و بنویسید :  
**Include** های زیر را به پروژه اضافه کنید :

```

#include <stdio.h>
#include <conio.h>
#include <windows.h>

```

حالا ما می خواهیم LED را روشن و خاموش کنیم .  
 پس پایه + LED را به پین ۱ پرت پارالل متصل کنید .  
 و پایه - را به بدنه کامپیوتر .  
 خوب حالا DLL را Export می کنیم :

```

typedef void _stdcall (*oupfuncPtr)(short portaddr, short datum);

```

حالا یک متغیر از نوع INT در پایین `TForm1 *Form1` ; به صورت زیر تعریف کنید .  
**int lamp;**

حالا یک تایمر بر روی فرم بگذارید و روی آن دابل کلیک کنید و بنویسید :

```

HINSTANCE hLib;
oupfuncPtr oup32;
hLib = LoadLibrary("inpout32.dll");
oup32 = (oupfuncPtr) GetProcAddress(hLib, "Out32");
if (lamp == false)
{
    (oup32)(888,1);
    lamp =! lamp;
}

```

```
else  
{  
  (oup32)(888,0);  
  lamp =! lamp;  
}
```

۸۸۸ شماره پرت پارالل است .  
۱ شماره پین یک است و با این کار روی پین یک ولتاژ وارد شده و چراغ روشن می شود .  
اگر شماره ۰ را وارد کنید تمامی پین ها ۰ می شود .

نویسنده : مهرداد فرخ منش