

ควบคุม ET-jrLPT ด้วย VC++

ศุภชัย บุศราทิจ
คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยราชภัฏเพชรบุรี

สำหรับการควบคุมอุปกรณ์ด้วยเครื่องคอมพิวเตอร์ส่วนบุคคล (PC/Personal Computer) นั้นสามารถทำได้หลายวิธี ตั้งแต่ใช้การ์ดเสียบบนบอร์ดทั้งที่เป็นแบบ ISA และ PCI, เชื่อมต่อผ่านพอร์ตอนุกรมแบบ RS232, เชื่อมต่อพอร์ตอนุกรมแบบ USB, เชื่อมต่อผ่านระบบเครือข่าย, เชื่อมต่อผ่าน IR และเชื่อมต่อผ่านพอร์ตขนาน เราจะพบว่ามีความยืดหยุ่นในเรื่องของภาษาเขียนโปรแกรมและเครื่องมือในการพัฒนามากกว่าการใช้ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ควบคุม(ไม่มองในแง่ของต้นทุน)

สำหรับบอร์ดควบคุมที่เป็นชนิดเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคลของบริษัทอีทีทีทีที่มีหลายชนิดทั้งที่เป็นพอร์ตอนุกรม, พอร์ตขนานและการ์ดเชื่อมต่อ ซึ่งในบทความครั้งนี้ที่ฉันได้เขียนถึงการเชื่อมต่อผ่านการ์ด ET-PCI8255 V3 ด้วย Visual C++ 6 ไปแล้ว แต่ว่าบทความก่อนหน้านี้ที่ฉันเขียนเกี่ยวกับพอร์ตขนานด้วย ET-jrLPT นั้นจะเป็น Delphi ซึ่งตัวอย่างจากทางอีทีทีทีก็เป็น Visual BASIC ทำให้นักพัฒนา Visual C++ ดาต้าๆ(อย่างผมเป็นต้น)รู้สึกว่ามันไม่เป็นธรรมด้วยเหตุนี้จึงเกิดบทความนี้ขึ้นมา นั่นคือ ผมจะสร้างตัวอย่างสำหรับการเขียนโปรแกรมเชื่อมต่อกับ ET-jrLPT ด้วย Visual C++ ในแบบง่ายๆ โดยแยกไลบรารีการเชื่อมต่อเป็นแฟ้มต่างหาก(ชื่อว่า ioport.h)เพื่อความสะดวกในการนำไปใช้

ตัวอย่างโปรแกรม

โปรแกรมตัวอย่างที่ฉันเขียนขึ้นมาเป็นการรวมการสั่งงานทั้งหมดที่สามารถทำได้กับบอร์ด ET-jrLPT โดยใช้ไดรเวอร์ UserPort.SYS สำหรับขับเคลื่อนการเชื่อมต่อบนระบบปฏิบัติการวินโดวส์ XP หรือตระกูล NT/2000 และจะข้ามการเรียกใช้โดยอัตโนมัติถ้าเป็นระบบปฏิบัติการในตระกูล 95/98/ME ซึ่งรายละเอียดเหล่านี้อ่านเพิ่มเติมได้จากบทความที่เป็น Delphi ครับ

หน้าต่างของโปรแกรมนั้นจะเป็นดังรูปที่ 1 นั่นคือ จะมีส่วนของการกำหนดว่าพอร์ตใดของ 8255 เป็นการทำงานแบบนำเข้าหรือส่งออก(input/output)เมื่อกำหนดเรียบร้อยแล้วให้คลิกที่ปุ่ม Connect เพื่อทำการเชื่อมต่อกับพอร์ตขนานและกำหนดค่าเริ่มต้นการทำงานของ 8255 ตามที่กำหนดเอาไว้ ถ้าพอร์ตใดกำหนดเอาไว้เป็นส่วนนำเข้า ผู้ใช้จะไม่สามารถทำการคลิกแก้ไข check box ในแต่ละบิตได้ โดยสถานะของแต่ละ check box นั้นจะเปลี่ยนแปลงเองโดยอัตโนมัติตามสถานะของพอร์ตที่อ่านเข้ามา และกรณีที่เป็นการส่งออกเราสามารถกำหนดสถานะของแต่ละบิตได้ด้วยการคลิกให้ check box ถูกเลือกและยกเลิกการเลือก แล้วโปรแกรมจะนำค่าสถานะของแต่ละบิตส่งไปที่พอร์ตนั้นๆเองโดยอัตโนมัติ

การทำงานของโปรแกรมในส่วนหลักจะอยู่ที่การทำงานของตัวตั้งเวลา(Timer)ที่จะทำงานเองตามกำหนดเวลาที่กำหนดเอาไว้ ทำให้เราไม่ต้องทำการวนรอบเพื่อสั่งให้อ่านหรือส่งข้อมูลจากพอร์ต แต่อย่างไรก็ดีถ้าจะให้โปรแกรมทำงานอย่างสมบูรณ์แบบนั้น ผู้เขียนโปรแกรมควรเลือกใช้เธรด(Thread)เป็นตัวอ่านและส่งข้อมูลแทนตัวตั้งเวลา เพราะจะทำงานได้เร็วกว่า และเช่นเดียวกันกับการเขียนบททดสอบถ้าเราใช้เธรด เราจะต้องประสบปัญหาเรื่องของความเร็วเครื่องเปลี่ยนแปลงการหวนเวลาก็เปลี่ยนแปลงไปด้วย

ดังนั้น การเขียนโปรแกรมเพื่อสั่งงานจากเครื่องคอมพิวเตอร์จึงเหมาะสมกับงานที่ใช้สำหรับการอ่านสถานะหรือสั่งการมากกว่าการควบคุมอุปกรณ์แบบทันเวลาจริง(Real Time) แต่ควรออกแบบโดยใช้ไมโครคอนโทรลเลอร์ช่วยในการควบคุมอุปกรณ์โดยเพิ่มในส่วนของการอ่านและสั่งงานผ่านการเชื่อมต่อกับคอมพิวเตอร์จะทำให้การทำงานสะดวกยิ่งขึ้น

โค้ดโปรแกรม

โปรแกรมส่วนแรกที่น่ามาแสดงคือ ioport.h ที่ทำการเชื่อมต่อกับ ET-jrLPT ซึ่งมีการออกแบบเหมือนกับยูนิคใน Delphi เพื่อให้เห็นว่าการเขียนโปรแกรมนั้นอัลกอริทึมเหมือนกันได้ แต่วิธีสั่งการของแต่ละภาษานั้นอาจจะแตกต่างกันไป

```
/******  
  Filename      : ioport.h  
  Author       : ศุภชัย บุศราทิจ (Supachai Busaratid)  
                Faculty of Information Technologies  
                Phetchaburi Rajabhat University  
                raek@etteam.com, bsupachai@gmail.com  
  
  Library      : UserPort.SYS  
  Operating System : Microsoft Windows xp Home  
  Compiler     : Microsoft Visual C++ 6 (Professional)  
  Hardware     : ET-jr LPT v1.0, ET-EXP4  
  PC Base     : IBM ThinkPad (P-M2.2GHz,256MB RAM)  
  Date        : March 14,2006  
  Thanks      : Kobkit Termpati  
                Busaratid Family  
                and ETT Team (www.etteam.com)  
*****/  
  
#include <string.h>  
#include <winsvc.h>  
#include <windows.h>  
  
#ifndef _IO_PORT_H  
#define _IO_PORT_H  
  
/******  
                I N T E R F A C E  
*****/  
BOOL StartDriver();  
BOOL StopDriver();  
BOOL StartUpIoPorts(int PortToAccess);  
  
void outportb(int portid, BYTE Value);  
BYTE inportb(int portid);  
  
void DelayPX();  
  
void WritePA(int PortB, BYTE Value);  
void WritePB(int PortB, BYTE Value);  
void WritePC(int PortB, BYTE Value);  
  
BYTE ReadPA(int PortB);  
BYTE ReadPB(int PortB);  
BYTE ReadPC(int PortB);  
  
void WritePCtrl(int PortB, BYTE Value);  
  
/******  
                Global variable and Source function  
*****/  
static BOOL bPrivException = false;  
static BYTE Result;
```

```

/*****
    o u t p o r t b
-----
เป็นคำสั่งส่งข้อมูลขนาด 8 บิต ไปที่พอร์ต
*****/
void outportb(int portid, BYTE Value)
{
    _asm {
        mov edx,portid
        mov al,Value
        out dx,al
    }
}

/*****
    i n p o r t b
-----
เป็นคำสั่งรับข้อมูลขนาด 8 บิต จากพอร์ต
*****/
BYTE inportb(int portid)
{
    BYTE Value;
    _asm {
        mov edx,portid
        in al,dx
        mov Value,al
    }
    return Value;
}

/*****
    S t a r t U p I o P o r t s
-----
เป็นคำสั่งเปิดการทำงานของพอร์ต
*****/
LONG WINAPI HandlerExceptionFilter ( EXCEPTION_POINTERS *pExPtrs )
{
    if (pExPtrs->ExceptionRecord->ExceptionCode==EXCEPTION_PRIV_INSTRUCTION) {
        pExPtrs->ContextRecord->Eip++;
        bPrivException = true;
        return EXCEPTION_CONTINUE_EXECUTION;
    }
    else
        return EXCEPTION_CONTINUE_SEARCH;
}

BOOL StartUpIoPorts(int PortToAccess)
{
    HANDLE hUserPort;
    hUserPort = CreateFile("\\\\.\\UserPort", GENERIC_READ, 0,
        NULL, OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL, NULL);
    CloseHandle(hUserPort); // Activate the driver
    Sleep(100); // We must make a process switch
}

```

```
SetUnhandledExceptionFilter(HandlerExceptionFilter);

bPrivException = false;
inportb(PortToAccess); // Try to access the given port address

if (bPrivException) {
    MessageBox(NULL,
        "Privileged instruction exception has occurred!\r\n\r\n"
        "To use this program under Windows NT | Windows 2000\r\n"
        "you need to install the driver "
        "\"UserPort.SYS\" and grant\r\n"
        "access to the ports used by this program.",
        NULL,MB_OK);
    return false;
}
return true;
}

/*****
    D e l a y P X
-----
หน่วงเวลาสำหรับพอร์ตของ 8255
*****/
void DelayPX()
{
    int i;
    for (i = 1; i < 50000; i++) ;
}

/*****
    W r i t e P A
-----
เป็นคำสั่งส่งข้อมูลไปที่พอร์ต A ของ 8255
*****/
void WritePA(int PortB, BYTE Value)
{
    outportb(PortB+2, 0xC ^ 0xB);
    outportb(PortB, Value);

    outportb(PortB+2, inportb(PortB+2) ^ 0x08);
    DelayPX();
    outportb(PortB+2, inportb(PortB+2) ^ 0x08);
}

/*****
    W r i t e P B
-----
เป็นคำสั่งส่งข้อมูลไปที่พอร์ต B ของ 8255
*****/
void WritePB(int PortB, BYTE Value)
{
    outportb(PortB+2, 0xD ^ 0xB);
    outportb(PortB, Value);

    outportb(PortB+2, inportb(PortB+2) ^ 0x08);
    DelayPX();
}
```

```
    outportb(PortB+2, inportb(PortB+2) ^ 0x08);
}

/*****
-----
      W r i t e P C
-----
เป็นคำสั่งส่งข้อมูลไปที่พอร์ต C ของ 8255
*****/
void WritePC(int PortB, BYTE Value)
{
    outportb(PortB+2, 0xE ^ 0xB);
    outportb(PortB, Value);

    outportb(PortB+2, inportb(PortB+2) ^ 0x08);
    DelayPX();
    outportb(PortB+2, inportb(PortB+2) ^ 0x08);
}

/*****
-----
      R e a d P A
-----
อ่านข้อมูลจากพอร์ต A ของ 8255
*****/
BYTE ReadPA(int PortB)
{
    /* กำหนด Address ให้อยู่ในตำแหน่ง PA */
    outportb(PortB+2, 0xC ^ 0xB);

    /* กำหนดให้ Data Register เป็นอินพุต " C5 = 1" */
    outportb(PortB+2, inportb(PortB+2) | 0x20);

    /* อ่านค่าจาก Data Register */
    outportb(PortB+2, inportb(PortB+2) ^ 0x04);
    Result = inportb(PortB);
    DelayPX();
    outportb(PortB+2, inportb(PortB+2) ^ 0x04);

    return Result;
}

/*****
-----
      R e a d P B
-----
อ่านข้อมูลจากพอร์ต B ของ 8255
*****/
BYTE ReadPB(int PortB)
{
    /* กำหนด Address ให้อยู่ในตำแหน่ง PB */
    outportb(PortB+2, 0xD ^ 0xB);

    /* กำหนดให้ Data Register เป็นอินพุต " C5 = 1" */
    outportb(PortB+2, inportb(PortB+2) | 0x20);
}
```

```
/* อ่านค่าจาก Data Register */
outportb(PortB+2, inportb(PortB+2) ^ 0x04);
Result = inportb(PortB);
DelayPX();
outportb(PortB+2, inportb(PortB+2) ^ 0x04);

return Result;
}

/*****
-----
ReadPC
-----
อ่านข้อมูลจากพอร์ต C ของ 8255
*****/

BYTE ReadPC(int PortB)
{
/* กำหนด Address ให้อยู่ในตำแหน่ง PC */
outportb(PortB+2, 0xE ^ 0xB);

/* กำหนดให้ Data Register เป็นอินพุต " C5 = 1" */
outportb(PortB+2, inportb(PortB+2) | 0x20);

/* อ่านค่าจาก Data Register */
outportb(PortB+2, inportb(PortB+2) ^ 0x04);
Result = inportb(PortB);
DelayPX();
outportb(PortB+2, inportb(PortB+2) ^ 0x04);

return Result;
}

/*****
-----
WritePCtrl
-----
เป็นคำสั่งส่งข้อมูลไปที่พอร์ตควบคุมของ 8255
*****/

void WritePCtrl(int PortB, BYTE Value)
{
outportb(PortB+2, 0xF ^ 0xB);
outportb(PortB, Value);

outportb(PortB+2, inportb(PortB+2) ^ 0x08);
DelayPX();
outportb(PortB+2, inportb(PortB+2) ^ 0x08);
}
```

```

/*****
-----
StopDriver
-----
เป็นคำสั่งหยุดการทำงานของไดรเวอร์
*****/

BOOL StopDriver()
{
    SC_HANDLE schService; /* Service Control Handle*/
    SC_HANDLE schSCManager;
    SERVICE_STATUS serviceStatus;

    /* 0. ถ้าไม่ใช่ Windows XP ไม่ต้องติดต่อกับ UserPort */
    if (LOBYTE(LOWORD(GetVersion)) < 5) {
        return false;
    } else {
        /* 1. เปิดตัวบริหารควบคุมบริการ (SCManager) */
        schSCManager = OpenSCManager(NULL, // machine (NULL == local)
                                     NULL, // database (NULL == default)
                                     SC_MANAGER_ALL_ACCESS); // access required

        /* 2. ถ้าเปิดใช้งานไม่สำเร็จ จะแจ้งเตือนข้อผิดพลาด */
        if (schSCManager == 0) {
            return false;
        } else
        /* 3. ถ้าเปิดใช้งานได้ */
        {
            /* 4. เปิดบริการ UserPort */
            schService = OpenService (schSCManager, "UserPort",
                                     SERVICE_ALL_ACCESS);

            /* 5. ถ้าบริการ UserPort เปิดขอใช้ไม่สำเร็จ จะทำการปิดการบริหารควบคุมบริการ */
            if (schService == 0) {
                CloseServiceHandle (schSCManager);
                return false;
            }
            /* 6. ถ้าเปิดบริการ UserPort สำเร็จ */
            else {
                /* 7. สั่งปิดบริการ (SERVICE_CONTROL_STOP) */
                ControlService (schService, SERVICE_CONTROL_STOP,
                               &serviceStatus);

                /* 8. ลบบริการ */
                DeleteService (schService);

                /* 9. ปิดการถือครอง */
                CloseServiceHandle (schService);
                CloseServiceHandle (schSCManager);
            } /* ? Open service */
        } /* ? Open SC manager */
    }
    return true;
}

```

```
/*-----  
S t a r t D r i v e r  
-----*/  
เป็นคำสั่งให้ดรเวอร์ UserPort.sys ทำงาน  
/*-----*/  
BOOL StartDriver()  
{  
    SC_HANDLE schService;  
    SC_HANDLE schSCManager;  
    HANDLE hDriver;  
    DWORD LastError;  
    char DriverName[256];  
  
    /* 0. ถ้าไม่ใช่ Windows XP ไม่ต้องติดต่อกับ UserPort */  
    if (LOBYTE(LOWORD(GetVersion))<5) {  
        return false;  
    } else {  
        /* 1. เปิดแฟ้มของไดรเวอร์ */  
        strcpy(DriverName, "C:\\Windows\\System32\\Drivers\\UserPort.sys");  
        hDriver = CreateFile( DriverName, GENERIC_READ, 0, NULL,  
                             OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0 );  
  
        /* 2. ถ้าเปิดไม่สำเร็จ แสดงว่ายังไม่ได้ติดตั้ง UserPort.sys เอาไว้ใน  
        C:\\Windows\\System32\\Drivers\\ */  
        if (hDriver == INVALID_HANDLE_VALUE) {  
            MessageBox(0, "ยังไม่ได้ติดตั้งแฟ้ม\n\r"  
                "\\Windows\\System32\\Drivers\\UserPort.sys",  
                "เกิดข้อผิดพลาดกับ UserPort", MB_OK);  
            return false;  
        } else  
        /* 3. ติดตั้งแฟ้มไดรเวอร์เรียบร้อยแล้ว */  
        {  
            /* 4. ปิดแฟ้ม */  
            CloseHandle(hDriver);  
  
            /* 5. เปิดตัวบริหารควบคุมบริการ */  
            schSCManager = OpenSCManager(NULL, // machine (NULL == local)  
                NULL, // database (NULL == default)  
                SC_MANAGER_ALL_ACCESS); // access required  
  
            /* 6. ถ้าเปิดไม่สำเร็จ จะแจ้งเตือนข้อผิดพลาด */  
            if (schSCManager == 0) {  
                /* 7. ถ้าความผิดพลาดเกี่ยวกับการสิทธิ์ของการเข้าถึง */  
                if (GetLastError() == ERROR_ACCESS_DENIED) {  
                    MessageBox(0, "คุณไม่มีสิทธิ์ในการติดตั้งไดรเวอร์\n\r"  
                        "กรุณาติดต่อกับผู้บริการระบบเครือข่าย",  
                        "เกิดข้อผิดพลาดกับ UserPort", MB_OK);  
                } else {  
                    /* 8. ถ้าเป็นกรณีอื่นๆ */  
                    MessageBox(0, "ไม่สามารถเริ่มต้นการทำงานของไดรเวอร์ได้สำเร็จ",  
                        "เกิดข้อผิดพลาดกับ UserPort", MB_OK);  
                    StopDriver();  
                }  
            }  
            return false;  
        } else
```



```
/* 9. ถ้าเปิดสำเร็จ */
{
    /* 10. สร้างตัวให้บริการชื่อว่า UserPort ในระบบ */
    schService = CreateService(
        schSCManager, // SCManager database
        "UserPort", // name of service
        "UserPort", // name to display
        SERVICE_START, // desired access
        SERVICE_KERNEL_DRIVER, // service type
        SERVICE_SYSTEM_START, // start type
        SERVICE_ERROR_NORMAL, // error
        // control type
        DriverName, // service's binary
        NULL, // no load ordering group
        NULL, // no tag identifier
        NULL, // no dependencies
        NULL, // LocalSystem account
        NULL); // no password

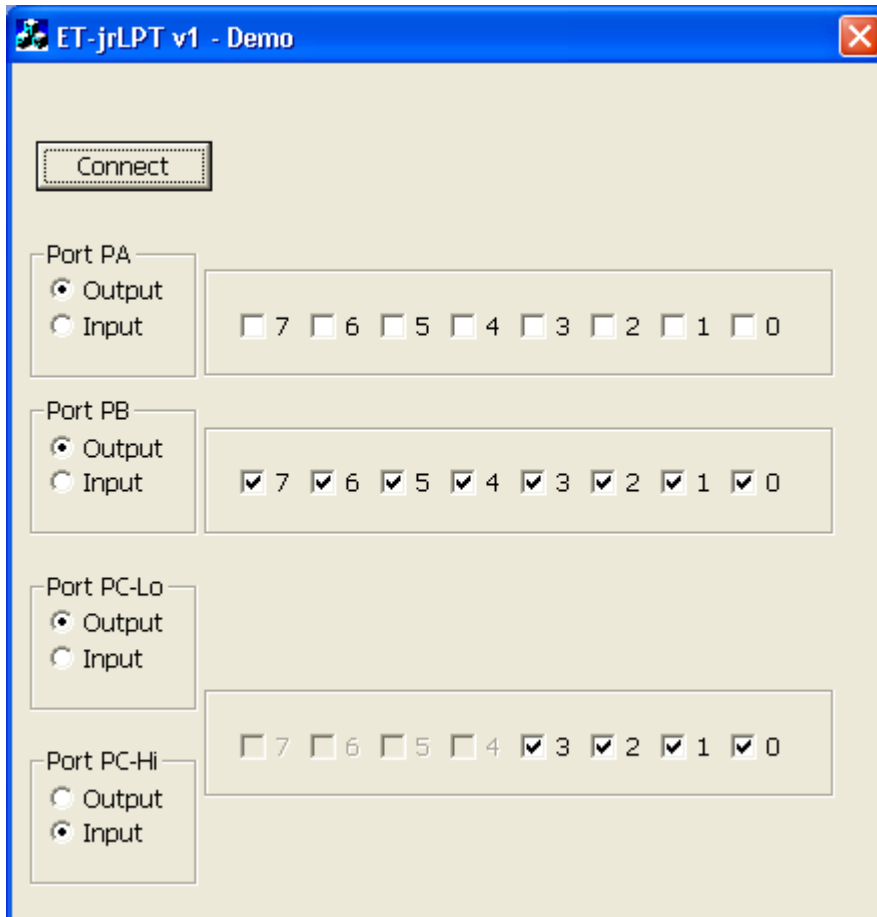
    /* 11. ถ้าสร้างไม่สำเร็จ จะแจ้งข้อผิดพลาดที่เกิดขึ้น */
    if (schService == 0) {
        /* 12. อ่านหมายเลขข้อผิดพลาด */
        GetLastError = GetLastError();

        /* 13. ถ้าเกิดข้อผิดพลาดมีบริการติดตั้งในระบบอยู่ก่อนแล้ว */
        if (LastError == ERROR_SERVICE_EXISTS) {
            MessageBox(0,
                "ไดรเวอร์ได้ถูกเรียกใช้งานไปก่อนหน้านี้แล้ว",
                "เกิดข้อผิดพลาดกับ UserPort", MB_OK);
            StopDriver();
        }
        else if (LastError == ERROR_ACCESS_DENIED)
            MessageBox(0, "คุณไม่มีสิทธิ์ในการติดตั้งไดรเวอร์\n\r"
                "กรุณาติดต่อกับผู้บริการระบบเครือข่าย",
                "เกิดข้อผิดพลาดกับ UserPort", MB_OK);
        else
            MessageBox(0,
                "ไม่สามารถเริ่มต้นการทำงานของไดรเวอร์ได้",
                "เกิดข้อผิดพลาดกับ UserPort", MB_OK);

        CloseServiceHandle (schSCManager);
        return false;
    } else
        /* 14. สร้างบริการสำเร็จ */
        {
            /* 15. สิ่งเริ่มต้นการทำงานของบริการ โดยไม่มีพารามิเตอร์ */
            StartService(schService, // service identifier
                0, // number of arguments
                NULL); // pointer to arguments

            /* 16. ปิดการถือครองบริการ */
            CloseServiceHandle (schService);
            CloseServiceHandle (schSCManager);
        } /* ? Create service. */
    } /* ? Open SC manager. */
} /* ? Driver does exist. */
```

```
}  
    return true;  
}  
  
#endif // _IO_PORT_H
```



รูปที่ 1 หน้าต่างโปรแกรมติดต่อกับ ET-jrLPT

โค้ดโปรแกรมส่วนที่สองนี้เป็นของการตอบสนองกับเหตุการณ์จากหน้าต่างโปรแกรมที่ออกแบบไว้ดังรูปที่ 1

```
// etLPTdemo01Dlg.cpp : implementation file  
//  
  
#include "stdafx.h"  
#include "etLPTdemo01.h"  
#include "etLPTdemo01Dlg.h"  
#include "IOPort.h"  
  
#ifdef _DEBUG  
#define new DEBUG_NEW  
#undef THIS_FILE
```

```
static char THIS_FILE[] = __FILE__;
#endif

bool bLPT_ok = false; // สถานะของ LPT
int PortAddr = 0x378; // ค่าตำแหน่งของ LPT
BYTE Control = 0x00; // บัฟเฟอร์ของ Control Port ของ 8255
BYTE PA_DAT = 0x00; // บัฟเฟอร์ของ PA
BYTE PB_DAT = 0x00; // บัฟเฟอร์ของ PB
BYTE PCH_DAT = 0x00; // บัฟเฟอร์ของ PC-Hi Bits
BYTE PCL_DAT = 0x00; // บัฟเฟอร์ของ PC-Low Bits
int m_nTimer = 0;

////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
   //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    }}AFX_DATA

// ClassWizard generated virtual function overrides
   //{{AFX_VIRTUAL(CAboutDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    }}AFX_VIRTUAL

// Implementation
protected:
   //{{AFX_MSG(CAboutDlg)
    }}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
   //{{AFX_DATA_INIT(CAboutDlg)
    }}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CAboutDlg)
    }}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
   //{{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    }}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CEtLPTdemo01Dlg dialog

CEtLPTdemo01Dlg::CEtLPTdemo01Dlg(CWnd* pParent /*=NULL*/)
    : CDialog(CEtLPTdemo01Dlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CEtLPTdemo01Dlg)
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CEtLPTdemo01Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CEtLPTdemo01Dlg)
    DDX_Control(pDX, IDC_RPC_OUT, m_pcl_o);
    DDX_Control(pDX, IDC_RPC_OUT, m_pch_o);
    DDX_Control(pDX, IDC_RPB_OUT, m_pb_o);
    DDX_Control(pDX, IDC_RPA_OUT, m_pa_o);
    DDX_Control(pDX, IDC_PC_B7, m_pc_b7);
    DDX_Control(pDX, IDC_PC_B6, m_pc_b6);
    DDX_Control(pDX, IDC_PC_B5, m_pc_b5);
    DDX_Control(pDX, IDC_PC_B4, m_pc_b4);
    DDX_Control(pDX, IDC_PC_B3, m_pc_b3);
    DDX_Control(pDX, IDC_PC_B2, m_pc_b2);
    DDX_Control(pDX, IDC_PB_B7, m_pb_b7);
    DDX_Control(pDX, IDC_PC_B1, m_pc_b1);
    DDX_Control(pDX, IDC_PC_B0, m_pc_b0);
    DDX_Control(pDX, IDC_PB_B6, m_pb_b6);
    DDX_Control(pDX, IDC_PB_B2, m_pb_b2);
    DDX_Control(pDX, IDC_PB_B5, m_pb_b5);
    DDX_Control(pDX, IDC_PB_B4, m_pb_b4);
    DDX_Control(pDX, IDC_PB_B3, m_pb_b3);
    DDX_Control(pDX, IDC_PB_B1, m_pb_b1);
    DDX_Control(pDX, IDC_PB_B0, m_pb_b0);
    DDX_Control(pDX, IDC_PA_B7, m_pa_b7);
    DDX_Control(pDX, IDC_PA_B6, m_pa_b6);
    DDX_Control(pDX, IDC_PA_B5, m_pa_b5);
    DDX_Control(pDX, IDC_PA_B4, m_pa_b4);
    DDX_Control(pDX, IDC_PA_B3, m_pa_b3);
    DDX_Control(pDX, IDC_PA_B2, m_pa_b2);
    DDX_Control(pDX, IDC_PA_B1, m_pa_b1);
    DDX_Control(pDX, IDC_PA_B0, m_pa_b0);
    DDX_Control(pDX, IDC_BTN_CONNECT, m_connect);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CEtLPTdemo01Dlg, CDialog)
   //{{AFX_MSG_MAP(CEtLPTdemo01Dlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BTN_CONNECT, OnBtnConnect)
    ON_WM_TIMER()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CEtLPTdemo01Dlg message handlers

BOOL CEtLPTdemo01Dlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here

    if (StartDriver()) {
        bLPT_ok = true;
        m_connect.EnableWindow(true);

        m_pa_o.EnableWindow(true); m_pa_o.SetCheck(1);
        m_pb_o.EnableWindow(true); m_pb_o.SetCheck(1);
        m_pch_o.EnableWindow(true); m_pch_o.SetCheck(1);
        m_pcl_o.EnableWindow(true); m_pcl_o.SetCheck(1);

        m_pa_b0.EnableWindow(false);
        m_pa_b1.EnableWindow(false);
        m_pa_b2.EnableWindow(false);
        m_pa_b3.EnableWindow(false);
        m_pa_b4.EnableWindow(false);
        m_pa_b5.EnableWindow(false);
        m_pa_b6.EnableWindow(false);
        m_pa_b7.EnableWindow(false);

        m_pb_b0.EnableWindow(false);
        m_pb_b1.EnableWindow(false);
        m_pb_b2.EnableWindow(false);
        m_pb_b3.EnableWindow(false);
        m_pb_b4.EnableWindow(false);
        m_pb_b5.EnableWindow(false);
        m_pb_b6.EnableWindow(false);
        m_pb_b7.EnableWindow(false);
    }
}
```

```
m_pc_b0.EnableWindow(false);
m_pc_b1.EnableWindow(false);
m_pc_b2.EnableWindow(false);
m_pc_b3.EnableWindow(false);
m_pc_b4.EnableWindow(false);
m_pc_b5.EnableWindow(false);
m_pc_b6.EnableWindow(false);
m_pc_b7.EnableWindow(false);
} else {
    m_connect.EnableWindow(false);

    m_pa_o.EnableWindow(false);
    m_pb_o.EnableWindow(false);
    m_pch_o.EnableWindow(false);
    m_pcl_o.EnableWindow(false);

    m_pa_b0.EnableWindow(false);
    m_pa_b1.EnableWindow(false);
    m_pa_b2.EnableWindow(false);
    m_pa_b3.EnableWindow(false);
    m_pa_b4.EnableWindow(false);
    m_pa_b5.EnableWindow(false);
    m_pa_b6.EnableWindow(false);
    m_pa_b7.EnableWindow(false);

    m_pb_b0.EnableWindow(false);
    m_pb_b1.EnableWindow(false);
    m_pb_b2.EnableWindow(false);
    m_pb_b3.EnableWindow(false);
    m_pb_b4.EnableWindow(false);
    m_pb_b5.EnableWindow(false);
    m_pb_b6.EnableWindow(false);
    m_pb_b7.EnableWindow(false);

    m_pc_b0.EnableWindow(false);
    m_pc_b1.EnableWindow(false);
    m_pc_b2.EnableWindow(false);
    m_pc_b3.EnableWindow(false);
    m_pc_b4.EnableWindow(false);
    m_pc_b5.EnableWindow(false);
    m_pc_b6.EnableWindow(false);
    m_pc_b7.EnableWindow(false);
}

return TRUE; // return TRUE unless you set the focus to a control
}

void CEtLPTdemo01Dlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFFF) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
```

```
{
    CDialog::OnSysCommand(nID, lParam);
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CEtLPTdemo01Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CEtLPTdemo01Dlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

BOOL CEtLPTdemo01Dlg::DestroyWindow()
{
    // TODO: Add your specialized code here and/or call the base class
    if (bLPT_ok) {
        if (m_nTimer) {
            KillTimer(m_nTimer);
            m_nTimer = 0;
        }
        StopDriver();
    }

    return CDialog::DestroyWindow();
}
```

```
void CETLPTdemo01Dlg::OnBtnConnect()
{
    // TODO: Add your control notification handler code here
    if (bLPT_ok) {
        if (m_nTimer) {
            KillTimer(m_nTimer);
            m_nTimer = 0;
        }
        if (StartUpIoPorts(PortAddr)) {
            Control = 0x80; /* 1000-0000 */
            if (m_pa_o.GetCheck()) {
                m_pa_b0.EnableWindow(true);
                m_pa_b1.EnableWindow(true);
                m_pa_b2.EnableWindow(true);
                m_pa_b3.EnableWindow(true);
                m_pa_b4.EnableWindow(true);
                m_pa_b5.EnableWindow(true);
                m_pa_b6.EnableWindow(true);
                m_pa_b7.EnableWindow(true);
            } else {
                m_pa_b0.EnableWindow(false);
                m_pa_b1.EnableWindow(false);
                m_pa_b2.EnableWindow(false);
                m_pa_b3.EnableWindow(false);
                m_pa_b4.EnableWindow(false);
                m_pa_b5.EnableWindow(false);
                m_pa_b6.EnableWindow(false);
                m_pa_b7.EnableWindow(false);
                Control |= 0x10; /* XXX1-XXXX */
            }
        }

        if (m_pb_o.GetCheck()) {
            m_pb_b0.EnableWindow(true);
            m_pb_b1.EnableWindow(true);
            m_pb_b2.EnableWindow(true);
            m_pb_b3.EnableWindow(true);
            m_pb_b4.EnableWindow(true);
            m_pb_b5.EnableWindow(true);
            m_pb_b6.EnableWindow(true);
            m_pb_b7.EnableWindow(true);
        } else {
            m_pb_b0.EnableWindow(false);
            m_pb_b1.EnableWindow(false);
            m_pb_b2.EnableWindow(false);
            m_pb_b3.EnableWindow(false);
            m_pb_b4.EnableWindow(false);
            m_pb_b5.EnableWindow(false);
            m_pb_b6.EnableWindow(false);
            m_pb_b7.EnableWindow(false);
            Control |= 0x02; /* XXXX-XX1X */
        }

        if (m_pch_o.GetCheck()) {
            m_pc_b4.EnableWindow(true);
            m_pc_b5.EnableWindow(true);
            m_pc_b6.EnableWindow(true);
            m_pc_b7.EnableWindow(true);
        } else {

```



```
        m_pc_b4.EnableWindow(false);
        m_pc_b5.EnableWindow(false);
        m_pc_b6.EnableWindow(false);
        m_pc_b7.EnableWindow(false);
        Control |= 0x08; /* XXXX-1XXX */
    }

    if (m_pcl_o.GetCheck()) {
        m_pc_b0.EnableWindow(true);
        m_pc_b1.EnableWindow(true);
        m_pc_b2.EnableWindow(true);
        m_pc_b3.EnableWindow(true);
    } else {
        m_pc_b0.EnableWindow(false);
        m_pc_b1.EnableWindow(false);
        m_pc_b2.EnableWindow(false);
        m_pc_b3.EnableWindow(false);
        Control |= 0x01; /* XXXX-XXX1 */
    }

    WritePCtrl(PortAddr, Control);
    m_nTimer = SetTimer(1, 100, 0);
    AfxMessageBox("เปิดพอร์ต 0x378 สำเร็จแล้ว");
//
    } else {
        AfxMessageBox("ไม่สามารถเปิดใช้พอร์ตหมายเลข 0x378");
    }
} else {
    AfxMessageBox("ไม่สามารถเชื่อมโยงได้เนื่องจากไดรเวอร์ไม่พร้อม");
}
}

void CEtLPTdemo01Dlg::OnTimer(UINT nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    if (bLPT_ok) {
        if (m_pa_o.GetCheck()) {

            PA_DAT = 0x00;

            if (m_pa_b0.GetCheck()) PA_DAT |= 0x01;
            if (m_pa_b1.GetCheck()) PA_DAT |= 0x02;
            if (m_pa_b2.GetCheck()) PA_DAT |= 0x04;
            if (m_pa_b3.GetCheck()) PA_DAT |= 0x08;
            if (m_pa_b4.GetCheck()) PA_DAT |= 0x10;
            if (m_pa_b5.GetCheck()) PA_DAT |= 0x20;
            if (m_pa_b6.GetCheck()) PA_DAT |= 0x40;
            if (m_pa_b7.GetCheck()) PA_DAT |= 0x80;

            WritePA(PortAddr, PA_DAT);

        } else {
            PA_DAT = ReadPA(PortAddr);

            if (PA_DAT & 0x01) m_pa_b0.SetCheck(1);
            else m_pa_b0.SetCheck(0);

            if (PA_DAT & 0x02) m_pa_b1.SetCheck(1);
            else m_pa_b1.SetCheck(0);
        }
    }
}
```

```
if (PA_DAT & 0x04) m_pa_b2.SetCheck(1);
else m_pa_b2.SetCheck(0);

if (PA_DAT & 0x08) m_pa_b3.SetCheck(1);
else m_pa_b3.SetCheck(0);

if (PA_DAT & 0x10) m_pa_b4.SetCheck(1);
else m_pa_b4.SetCheck(0);

if (PA_DAT & 0x20) m_pa_b5.SetCheck(1);
else m_pa_b5.SetCheck(0);

if (PA_DAT & 0x40) m_pa_b6.SetCheck(1);
else m_pa_b6.SetCheck(0);

if (PA_DAT & 0x80) m_pa_b7.SetCheck(1);
else m_pa_b7.SetCheck(0);
}

if (m_pb_o.GetCheck()) {

    PB_DAT = 0x00;

    if (m_pb_b0.GetCheck()) PB_DAT |= 0x01;
    if (m_pb_b1.GetCheck()) PB_DAT |= 0x02;
    if (m_pb_b2.GetCheck()) PB_DAT |= 0x04;
    if (m_pb_b3.GetCheck()) PB_DAT |= 0x08;
    if (m_pb_b4.GetCheck()) PB_DAT |= 0x10;
    if (m_pb_b5.GetCheck()) PB_DAT |= 0x20;
    if (m_pb_b6.GetCheck()) PB_DAT |= 0x40;
    if (m_pb_b7.GetCheck()) PB_DAT |= 0x80;

    WritePB(PortAddr, PB_DAT);

} else {
    PB_DAT = ReadPB(PortAddr);

    if (PB_DAT & 0x01) m_pb_b0.SetCheck(1);
    else m_pb_b0.SetCheck(0);

    if (PB_DAT & 0x02) m_pb_b1.SetCheck(1);
    else m_pb_b1.SetCheck(0);

    if (PB_DAT & 0x04) m_pb_b2.SetCheck(1);
    else m_pb_b2.SetCheck(0);

    if (PB_DAT & 0x08) m_pb_b3.SetCheck(1);
    else m_pb_b3.SetCheck(0);

    if (PB_DAT & 0x10) m_pb_b4.SetCheck(1);
    else m_pb_b4.SetCheck(0);

    if (PB_DAT & 0x20) m_pb_b5.SetCheck(1);
    else m_pb_b5.SetCheck(0);

    if (PB_DAT & 0x40) m_pb_b6.SetCheck(1);
    else m_pb_b6.SetCheck(0);
```

```
        if (PB_DAT & 0x80) m_pb_b7.SetCheck(1);
        else m_pb_b7.SetCheck(0);
    }

    if (m_pch_o.GetCheck()) {

        PCH_DAT = 0x00;

        if (m_pc_b4.GetCheck()) PCH_DAT |= 0x10;
        if (m_pc_b5.GetCheck()) PCH_DAT |= 0x20;
        if (m_pc_b6.GetCheck()) PCH_DAT |= 0x40;
        if (m_pc_b7.GetCheck()) PCH_DAT |= 0x80;

        PCH_DAT = PCH_DAT | (PCL_DAT & 0x0F);

        WritePC (PortAddr, PCH_DAT);

    } else {
        PCH_DAT = ReadPC (PortAddr) & 0xFF00;

        if (PCH_DAT & 0x10) m_pc_b4.SetCheck(1);
        else m_pc_b4.SetCheck(0);

        if (PCH_DAT & 0x20) m_pc_b5.SetCheck(1);
        else m_pc_b5.SetCheck(0);

        if (PCH_DAT & 0x40) m_pc_b6.SetCheck(1);
        else m_pc_b6.SetCheck(0);

        if (PCH_DAT & 0x80) m_pc_b7.SetCheck(1);
        else m_pc_b7.SetCheck(0);
    }

    if (m_pcl_o.GetCheck()) {

        PCL_DAT = 0x00;

        if (m_pc_b0.GetCheck()) PCL_DAT |= 0x01;
        if (m_pc_b1.GetCheck()) PCL_DAT |= 0x02;
        if (m_pc_b2.GetCheck()) PCL_DAT |= 0x04;
        if (m_pc_b3.GetCheck()) PCL_DAT |= 0x08;

        PCL_DAT = PCH_DAT | (PCH_DAT & 0xF0);

        WritePC (PortAddr, PCL_DAT);

    } else {
        PCL_DAT = ReadPC (PortAddr) & 0x00FF;

        if (PCL_DAT & 0x01) m_pc_b0.SetCheck(1);
        else m_pc_b0.SetCheck(0);

        if (PCL_DAT & 0x02) m_pc_b1.SetCheck(1);
        else m_pc_b1.SetCheck(0);

        if (PCL_DAT & 0x04) m_pc_b2.SetCheck(1);
        else m_pc_b2.SetCheck(0);
    }
}
```

```
        if (PCL_DAT & 0x08) m_pc_b3.SetCheck(1);  
        else m_pc_b3.SetCheck(0);  
    }  
}  
CDialog::OnTimer(nIDEvent);  
}
```

สรุป

จากบทความนี้ผู้เขียนโปรแกรมด้วย Visual C++(สำหรับ Visual C++ .NET ก็สามารถใช้งานได้เช่นเดียวกัน) คงจะได้ประโยชน์กันไม่มากนักน้อยนะครับ และหวังว่าคงพอเป็นแนวทางสำหรับการนำไปพัฒนาให้หลากหลายมากขึ้น ถ้าผู้อ่านต้องการนำโค้ดที่นำไปพัฒนาต่อมาเผยแพร่ก็ส่งมาให้ผมได้ นะครับ แล้วผมจะแปลงเป็นแฟ้มสำหรับเป็นบทความขึ้นบนเว็บอีทีทีให้ครับผม

สุดท้าย ต้องขอขอบคุณ คุณกอบกิจ เดิมผาติ และทีมงานอีทีที อีกเช่นเคยครับ เพราะถ้าไม่มีบุคคลเหล่านี้ก็คงไม่มีบทความที่ผมเขียนถูกบ้างผิดบ้างให้ได้อ่านกันหรอกครับ ขอให้สนุกกับการเขียนโปรแกรมครับ