

พอร์ตขนานและการเชื่อมต่อ ด้วย Borland Delphi ตอนที่ 2

ศุภชัย บุศราทิจ (raek@etteam.com)

คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยราชภัฏเพชรบุรี

1. บทนำ

ในบทความตอนแรกเราได้เรียนรู้เกี่ยวกับพอร์ตขนานกัน ไปคร่าวๆ พร้อมกับวงจร LPT Port ที่เชื่อมเข้ากับ 8255 และได้เขียน โปรแกรมติดต่อสื่อสารกับพอร์ตเป็นที่เรียบร้อยแล้ว สำหรับผู้อ่านที่ไม่ต้องการทำบอร์ดเองก็สามารถสั่งซื้อได้โดยตรงกับบริษัททีที (ขอโฆษณาหน่อยนะครับ) ซึ่งวงจรของบอร์ดทดลองนั้นก็จะมีเหมือนกับวงจรมัดผมได้แสดงไปแล้วเมื่อครั้งก่อน

สำหรับครั้งนี้จุดประสงค์ของบทความของเราก็คือ เขียนโปรแกรมควบคุมไอซี 8255 ให้ทำงานตามโหมดที่เราต้องการ (รายละเอียดของ 8255 ได้กล่าวไปแล้วในบทความตอนแรก) นั่นคือ เราจะส่งงานพอร์ต A, B, C และพอร์ตควบคุมได้อย่างไร เมื่อส่งควบคุมพอร์ตได้แล้ว ขั้นตอนต่อไปจะเป็นเรื่องของการเชื่อมต่อกับอุปกรณ์ภายนอก ซึ่งจะเกี่ยวข้องกับวงจรขับหลอด LED และควบคุม 7-Segment ทั้งแบบ 1 หลักและหลายหลัก ซึ่งเป็นพื้นฐานสำหรับบทความตอนที่ 2 ที่เกี่ยวข้องกับการขับวงจรมอเตอร์อีกด้วย

สุดท้ายสิ่งที่เราจะเรียนรู้เพิ่มขึ้นในบทความนี้ก็คือ ทำอย่างไรเราจะตรวจสอบรุ่นของวินโดวส์ได้ว่าเป็น Windows 98/ME/NT/2000/XP/2003 และทำอย่างไรเราจะสามารถเขียนโปรแกรมตัวเดียวแต่ทำงานได้ทั้งบนวินโดวส์หลายรุ่น ซึ่งผู้เขียนหวังว่าบทความในตอนที่ 2 นี้จะมีประโยชน์กับผู้อ่านบทความบ้างไม่มากก็น้อยครับ

2. การควบคุม 8255

จากหัวข้อ 4.2 ในบทความตอนที่ 1 นั้น ทำให้ทราบว่าไอซี 8255 มีพอร์ตให้ใช้งานด้วยกัน 3 พอร์ต คือ PA หรือพอร์ต A, PB หรือพอร์ต B และ PC หรือพอร์ต C ซึ่งแต่ละพอร์ตนั้นจะเป็นพอร์ตขนาด 8 บิต และจะถูกควบคุมการทำงานด้วยการกำหนดค่าให้กับ PCTRL หรือพอร์ตควบคุม (Control Port) ดังนั้น สิ่งแรกที่เราจะต้องทราบก่อนที่จะทำการเขียนโปรแกรมเพื่อรับข้อมูลหรือส่งข้อมูลนั้นควรศึกษาเรื่องการควบคุมการทำงานกันก่อน

หมายเหตุ ในการเขียนโปรแกรมด้วย LPT นั้นมีปัญหาใหญ่อยู่ 1 เรื่องนั่นคือ ค่าการหน่วงเวลา ทั้งนี้ก็เนื่องจากคอมพิวเตอร์นั้นมีความเร็วสูงมาก และหน่วยประมวลผลแต่ละรุ่นก็มีความเร็วไม่เท่ากัน โปรแกรมจึงทำงานเร็ว/ช้าไม่เท่ากัน ดังนั้น ผู้เขียนโปรแกรมจึงมีหน้าที่ปรับค่านับเวลาที่เหมาะสมกับเครื่องที่ใช้งานด้วยครับ

2.1 การสั่งพอร์ตควบคุม

เราสามารถสั่งพอร์ตควบคุมตามโปรแกรมดังต่อไปนี้

```
procedure WritePCtrl(PortB : Integer; value : byte);
begin
  { เลือกพอร์ตเป็นพอร์ตควบคุม }
  outportb(PortB+2, $F xor $B);

  { ส่งข้อมูลไปที่พอร์ต }
  outportb(PortB, value);

  { ส่งสัญญาณให้ 8255 ทำงาน }
  outportb(PortB+2, inportb(PortB+2) xor $08);
  DelayPX;
  outportb(PortB+2, inportb(PortB+2) xor $08);
End;
```

ในการใช้งานนั้น PortB คือหมายเลขพอร์ตของ LPT Port และ value คือค่าที่ใช้สำหรับควบคุมการทำงานของพอร์ต 8255 (ตาราง 8,9 และ 10 ของบทความตอนที่ 1) ส่วนฟังก์ชัน DelayPX ผู้เขียนได้ปรับการหน่วงเวลาเอาไว้ดังนี้

```
procedure DelayPX;
var
  i : integer;
begin
  for i := 1 to 50000 do
  begin
    { หน่วงเวลาฟรีๆจ๊ะ }
  end;
end;
```

2.2 การรับ/ส่งข้อมูลพอร์ต A/B/C

เมื่อทำการสั่งการทำงานของพอร์ตผ่านทางพอร์ตควบคุมเป็นที่เรียบร้อยแล้ว เราสามารถเขียนโปรแกรมเพื่อส่ง/รับข้อมูลจากพอร์ต A, B และ C ด้วยฟังก์ชัน ต่อไปนี้ครับ

```
procedure WritePA(PortB : Integer; Value : byte);
begin
  { เลือกพอร์ตเป็นพอร์ต A }
  outportb(PortB+2, $C xor $B);

  { ส่งข้อมูลออกพอร์ต A }
  outportb(PortB, value);
```

มีต่อ:→

```

{ สัญญาณให้ 8255 ทำงาน }
    outportb(PortB+2, inportb(PortB+2) xor $08);
    DelayPX;
    outportb(PortB+2, inportb(PortB+2) xor $08);
end;

procedure WritePB(PortB : Integer; Value : byte);
begin
{ เลือกพอร์ตเป็นพอร์ต B }
    outportb(PortB+2, $D xor $B);

{ ส่งข้อมูลออกพอร์ต B }
    outportb(PortB, value);

{ สัญญาณให้ 8255 ทำงาน }
    outportb(PortB+2, inportb(PortB+2) xor $08);
    DelayPX;
    outportb(PortB+2, inportb(PortB+2) xor $08);
end;

procedure WritePC(PortB : Integer; Value : byte);
begin
{ เลือกพอร์ตเป็นพอร์ต C }
    outportb(PortB+2, $E xor $B);

{ ส่งข้อมูลออกพอร์ต C }
    outportb(PortB, value);

{ สัญญาณให้ 8255 ทำงาน }
    outportb(PortB+2, inportb(PortB+2) xor $08);
    DelayPX;
    outportb(PortB+2, inportb(PortB+2) xor $08);
end;

```

ฟังก์ชัน WritePA, WritePB และ WritePC นั้นต้องการพารามิเตอร์ 2 ตัวด้วยกันคือ PortB สำหรับกำหนดค่าของพอร์ต LPT ส่วน Value คือ ค่าตัวเลขขนาด 8 บิต (แบบไม่สนใจเครื่องหมายลบ)

```

function ReadPA(PortB : integer):byte;
begin
{ กำหนด Address ให้อยู่ในตำแหน่ง PA }
    outportb(PortB+2, $C xor $B);

{ กำหนดให้ Data Register เป็นอินพุต " C5 = 1" }
    outportb(PortB+2, inportb(PortB+2) or $20);

{ อ่านค่าจาก Data Register }
    outportb(PortB+2, inportb(PortB+2) xor $04);
    Result := inportb(PortB);
    DelayPX;
    outportb(PortB+2, inportb(PortB+2) xor $04);
end;

```

```
function ReadPB (PortB : integer) :byte;
begin
  { กำหนด Address ให้อยู่ในตำแหน่ง PB }
  outportb (PortB+2, $D xor $B);

  { กำหนดให้ Data Register เป็นอินพุต " C5 = 1" }
  outportb (PortB+2, inportb (PortB+2) or $20);

  { อ่านค่าจาก Data Register }
  outportb (PortB+2, inportb (PortB+2) xor $04);
  Result := inportb (PortB);
  DelayPX;
  outportb (PortB+2, inportb (PortB+2) xor $04);
end;

function ReadPC (PortB : integer) :byte;
begin
  { กำหนด Address ให้อยู่ในตำแหน่ง PC }
  outportb (PortB+2, $E xor $B);

  { กำหนดให้ Data Register เป็นอินพุต " C5 = 1" }
  outportb (PortB+2, inportb (PortB+2) or $20);

  { อ่านค่าจาก Data Register }
  outportb (PortB+2, inportb (PortB+2) xor $04);
  Result := inportb (PortB);
  DelayPX;
  outportb (PortB+2, inportb (PortB+2) xor $04);
end;
```

ฟังก์ชัน ReadPA, ReadPB และ ReadPC ต้องการพารามิเตอร์ (PortB) เป็นค่าตำแหน่งของพอร์ต LPT หลังจากนั้นจะดึงค่ากลับมาจากพอร์ตที่อ่านกลับมาเป็นข้อมูลขนาด 8 บิต (ไม่สนใจเครื่องหมายลบ)

หมายเหตุ ต้องเรียกฟังก์ชัน WritePCtrl โดยกำหนด value ให้ถูกต้องด้วย เช่น ถ้ากำหนดให้ PA เป็นพอร์ตส่งข้อมูลออก แต่ในโปรแกรมเขียนโปรแกรมเพื่ออ่านข้อมูล การทำงานของโปรแกรมจะผิดพลาดได้

3. การเรียกใช้งาน UserPort.SYS

จากตัวอย่างของบทความตอนที่ 1 นั้นเราได้เขียนโปรแกรมติดต่อกับ UserPort.SYS โดยก่อนที่จะใช้งานโปรแกรมได้นั้นจำเป็นต้องเรียกโปรแกรม UserPort.EXE เพื่อทำการเปิดบริการ (Service) แล้วโปรแกรมของเราจึงสามารถทำงานผ่าน UserPort.SYS ได้ ปัญหาของการทำอย่างนี้มีด้วยกัน 2 ข้อคือ

- ถ้าระบบปฏิบัติการเป็น Windows 98/ME ซึ่งไม่จำเป็นต้องใช้ UserPort.SYS ในการติดต่อกับฮาร์ดแวร์ โปรแกรมที่เขียนขึ้นมานั้นก็จะทำงานผิดพลาด เพราะเราเขียนโปรแกรมโดยเรียก UserPort.SYS ให้ทำงานเสมอ

- ถ้าต้องการเขียนโปรแกรมให้ผู้อื่นใช้งาน การที่จะต้องมาเรียกโปรแกรม UserPort.EXE เพื่อเปิดบริการนั้นจะเป็นการเพิ่มขึ้นขั้นตอนการทำงาน และไม่สะดวกสำหรับผู้ใช้ อาจจะทำให้เกิดปัญหาถึงขั้นไม่อยากใช้โปรแกรมที่เราเขียน ซึ่งถือได้ว่าเป็นการสร้างภาพพจน์ที่ไม่ดีกับเราเป็นอย่างมาก

ด้วยปัญหาทั้งสองข้อนั้นทำให้เราต้องเขียนโปรแกรมเพิ่มขึ้นมาเพื่อตรวจสอบรุ่นของวินโดวส์ และเปิดบริการของ UserPort.SYS โดยอัตโนมัติ ซึ่งมีข้อแม้ว่าการเปิดบริการของ UserPort.SYS นั้นจะต้องกระทำเฉพาะระบบปฏิบัติการวินโดวส์ 2000/XP/2003 เท่านั้น ส่วน 98/ME นั้นไม่ต้องเปิดบริการของ UserPort.SYS แต่อย่างไร ดังนั้น สามารถเขียนขั้นตอนการทำงานโปรแกรมได้ 2 ฟังก์ชันคือ StopDriver และ StartDriver ดังนี้

```
function StartDriver:boolean;
var
  schService      : SC_HANDLE;
  schSCManager    : SC_HANDLE;
  hDriver         : THandle;
  LastError       : DWORD;
  param           : PChar;
  DriverName      : PChar;
begin
{ 0. ถ้าไม่ใช่ Windows XP ไม่ต้องติดต่อกับ UserPort }
  if (LOBYTE(LOWORD(GetVersion)) < 5) then
  begin
    StartDriver := false;
  end
  else
  begin
{ 1. เปิดแฟ้มของไดรเวอร์ }
    DriverName := 'C:\Windows\System32\Drivers\UserPort.sys';
    hDriver := CreateFile( DriverName, GENERIC_READ, 0, nil,
      OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);

{ 2. ถ้าเปิดไม่สำเร็จ แสดงว่ายังไม่ได้ติดตั้ง UserPort.sys เอาไว้ใน  
C:\Windows\System32\Drivers\ }
    if (hDriver = INVALID_HANDLE_VALUE) then
    begin
      MessageBox(0, 'ยังไม่ได้ติดตั้งแฟ้ม'+#10+#13+  
'\Windows\System32\Drivers\UserPort.sys',  
'เกิดข้อผิดพลาดกับ UserPort', MB_OK);
      StartDriver := false;
    end
    else
{ 3. ติดตั้งแฟ้มไดรเวอร์เรียบร้อยแล้ว }
    begin
{ 4. ปิดแฟ้ม }
      CloseHandle(hDriver);

{ 5. เปิดตัวบริหารควบคุมบริการ }
      schSCManager := OpenSCManager(nil, // machine (NULL == local)
        nil, // database (NULL == default)
        SC_MANAGER_ALL_ACCESS); // access required
```

มีต่อ: →

```

{ 6. ถ้าเปิดไม่สำเร็จ จะแจ้งเตือนข้อผิดพลาด }
if (schSCManager = 0) then
begin
{ 7. ถ้าความผิดพลาดเกี่ยวกับการสิทธิ์ของการเข้าถึง }
if (GetLastError()= ERROR_ACCESS_DENIED) then
MessageBox(0, 'คุณไม่มีสิทธิ์ในการติดตั้งไดรเวอร์'+#10+#13+
'กรุณาติดต่อกับผู้บริการระบบเครือข่าย',
'เกิดข้อผิดพลาดกับ UserPort',MB_OK)
else
{ 8. ถ้าเป็นกรณีอื่นๆ }
MessageBox(0,'ไม่สามารถเริ่มต้นการทำงานของไดรเวอร์ได้สำเร็จ',
'เกิดข้อผิดพลาดกับ UserPort',MB_OK);
StartDriver := false;
end
else
{ 9. ถ้าเปิดสำเร็จ }
begin
{ 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
nil, // no load ordering group
nil, // no tag identifier
nil, // no dependencies
nil, // LocalSystem account
nil); // no password
{ 11. ถ้าสร้างไม่สำเร็จ จะแจ้งข้อผิดพลาดที่เกิดขึ้น }
if (schService = 0) then
begin
{ 12. อ่านหมายเลขข้อผิดพลาด }
LastError := GetLastError();
{ 13. ถ้าเกิดข้อผิดพลาดมีบริการติดตั้งในระบบอยู่ก่อนแล้ว }
if (LastError = ERROR_SERVICE_EXISTS) then
MessageBox(0,'ไดรเวอร์ได้ถูกเรียกใช้งานไปก่อนหน้านี้แล้ว',
'เกิดข้อผิดพลาดกับ UserPort',MB_OK)
else if (LastError = ERROR_ACCESS_DENIED) then
MessageBox(0,'คุณไม่มีสิทธิ์ในการติดตั้งไดรเวอร์'+#10+#13+
'กรุณาติดต่อกับผู้บริการระบบเครือข่าย',
'เกิดข้อผิดพลาดกับ UserPort',MB_OK)
else
MessageBox(0,'ไม่สามารถเริ่มต้นการทำงานของไดรเวอร์ได้',
'เกิดข้อผิดพลาดกับ UserPort',MB_OK);
CloseServiceHandle (schSCManager);
StartDriver := false;
end
else
{ 14. สร้างบริการสำเร็จ }
begin
{ 15. สั่งเริ่มต้นการทำงานของบริการ โดยไม่มีพารามิเตอร์ }
Param := '';
StartService (schService, // service identifier
0, // number of arguments
Param); // pointer to arguments

```

มีต่อ: →

```

{ 16. ปิดการถือครองบริการ }
    CloseServiceHandle (schService);
    CloseServiceHandle (schSCManager);
    StartDriver := true;
end; { ? Create service. }
end; { ? Open SC manager. }
end; { ? Driver does exist. }
end;
end;

```

```

function StopDriver : boolean;
var
    schService      : SC_HANDLE;    { Service Control Handle}
    schSCManager    : SC_HANDLE;
    serviceStatus   : SERVICE_STATUS;
begin
{ 0. ถ้าไม่ใช่ Windows XP ไม่ต้องติดต่อกับ UserPort }
    if (LOBYTE (LOWORD (GetVersion)) < 5) then
    begin
        StopDriver := false;
    end
    else
    begin
{ 1. เปิดตัวบริหารควบคุมบริการ (SCManager) }
        schSCManager := OpenSCManager (nil, // machine (NULL == local)
                                        nil, // database (NULL == default)
                                        SC_MANAGER_ALL_ACCESS); // access required

{ 2. ถ้าเปิดใช้งานไม่สำเร็จ จะแจ้งเตือนข้อผิดพลาด }
        if (schSCManager = 0) then
        begin
            Result := false;
        end
        else
{ 3. ถ้าเปิดใช้งานได้ }
            begin
{ 4. เปิดบริการ UserPort }
                schService := OpenService (schSCManager, 'UserPort',
                                            SERVICE_ALL_ACCESS);

{ 5. ถ้าบริการ UserPort เปิดขอใช้ไม่สำเร็จ จะทำการปิดการบริหารควบคุมบริการ }
                if (schService = 0) then
                begin
                    CloseServiceHandle (schSCManager);
                    StopDriver := false;
                end
                else
{ 6. ถ้าเปิดบริการ UserPort สำเร็จ }
                    begin
{ 7. สั่งปิดบริการ (SERVICE_CONTROL_STOP) }
                        ControlService (schService, SERVICE_CONTROL_STOP,
                                        serviceStatus);

{ 8. ลบบริการ }
                            DeleteService (schService);
                    end
                end
            end
        end
    end
end;

```

มีต่อ: →

{ 9. ปิดการถือครอง }

```

CloseServiceHandle (schService);
CloseServiceHandle (schSCManager);
StopDriver := true;
end; { ? Open service }
end; { ? Open SC manager }
end;
end;

```

การตรวจสอบรุ่นของวินโดวส์นั้นเราใช้ฟังก์ชัน `GetVersion` ซึ่งมีรูปแบบดังนี้

```
Function GetVersion:DWORD;
```

ค่าที่คืนมานั้นดูจากข้อมูลขนาด 8 บิต โดยมีข้อกำหนดคือ

Platform	High-order bit	Next 7 bits	Low-order byte
Windows NT 3.51	0	Build number	3
Windows NT 4.0	0	Build number	4
Windows 2000 or Windows XP	0	Build number	5
Windows 95, Windows 98, or Windows Me	1	Reserved	4
Win32s with Windows 3.1	1	Build number	3

เนื่องจากค่าที่คืนกลับมานั้นเป็น `DWORD` ที่มีขนาด 4 ไบต์ แต่การตรวจสอบรุ่นนั้นเราจะใช้ค่าจากไบต์แรกเท่านั้น และต้องการตรวจสอบเฉพาะตระกูล 2000/XP (รวมทั้ง 2003) ด้วย ดังนั้นเพื่อความสะดวกเราก็แค่ดูว่าค่า 1 ไบต์นั้น ถ้าน้อยกว่า 5 แสดงว่าไม่ใช่ 2000/XP/2003 แต่ถ้าเป็น 5 ก็แสดงว่าเป็น 2000/XP/2003 โค้ดโปรแกรมที่ได้คือ

```
(LOBYTE (LOWORD (GetVersion)) < 5
```

เพียงเท่านี้เราก็สามารถแยกแยะรุ่นของวินโดวส์ได้สำเร็จแล้วครับ ส่วนรายละเอียดเรื่องของการบริการ หรือ Service ของวินโดวส์นั้น ผู้เขียนขอไม่กล่าวถึงนะครับ เพราะมันไปเกี่ยวข้องกับ DDK (Device Developer Kit) ซึ่งผู้เขียนเองก็ไม่สามารถหาความรู้มาใช้งานได้ เนื่องจากปัจจุบัน ชุดพัฒนานี้ทางบริษัทไมโครซอฟต์ เปลี่ยนจากแจกจ่ายฟรี แบบเดียวกับ Platform SDK มาเป็นคิดเงินเป็นที่เรียบร้อยแล้ว (กลัวคิดเงิน เราก็ไม่ใช้ก็ได้เนอะ ... ฮะๆ) ประกอบกับชื่อมันก็ไม่ได้ใช้ สุดท้ายจึงไม่มีข้อมูล DDK อยู่ในมือเลยครับ (ร้ายยาวแต่จริงๆแล้ว ก็

เพราะไม่มีความรู้ด้าน DDK เลยครับ) แต่ก็มี Service อีกประเภทหนึ่งที่เราสามารถหาละเอียดมาอ่านได้ (งงใหม่ ว่าบางอย่างหาอ่านได้ แต่บางอย่างหาอ่านไม่ได้ เพราะขาย) นั่นคือ Service ที่เป็นของ Platform SDK ซึ่งระบบนี้จะทำงานได้กับ Windows NT/2000/XP/2003 เท่านั้น ซึ่งเป็นแหล่งความรู้เดียวที่ผมสืบค้นได้ โดยสรุปแล้วระบบ Service นั้นเป็นซอฟต์แวร์ชนิดหนึ่งที่รันเป็น background ซึ่งผู้เขียนโปรแกรมจะต้องขอเปิด service แล้วจึงจะเรียก service นั้นทำงานได้ แล้ว service ก็จะทำงานเป็น background ของระบบ เมื่อใช้งานเสร็จเราก็ต้องสั่งปิด service เพื่อลบโปรแกรมตัวนี้ออกจากหน่วยความจำ

โค้ดโปรแกรม IOPort.PAS ที่ปรับปรุงแก้ไขแล้วจึงเป็นดังนี้

```
{*****}
Filename      : ioport.pas
Author       : Thomas Franzon
Modified    : 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
Compiler     : Borland Delphi 6 personal
              or later
Hardware     : ET-jr LPT v1.0
              ET-EXP4
PC Base      : IBM ThinkPad (P-M2.2GHz,256MB RAM)
Date        : Aug 4, 2005
              Aug 15,2005 (StartDriver/StopDriver)
              Aus 23,2005 (Checked the WinVer)
              (Initialization)
              (Finalization)
              (DelayPX)
              (SendPA/PB/PC)
              (ReadPA/PB/PC)
Thanks      : Kobkit Termpati
              Busaratid Family
              and ETT Team
*****}

unit IOPort;

{*****}
      I N T E R F A C E
*****}
interface
uses windows, Winsvc;

function StartDriver : boolean;
function StopDriver : boolean;
function StartUpIoPorts(PortToAccess : integer) : boolean;

procedure outportb(portid : integer; value : BYTE);
function inportb(portid : integer) : byte;

procedure DelayPX;
```

```

procedure WritePA(PortB : Integer; Value : byte);
procedure WritePB(PortB : Integer; Value : byte);
procedure WritePC(PortB : Integer; Value : byte);
function  ReadPA(PortB : integer):byte;
function  ReadPB(PortB : integer):byte;
function  ReadPC(PortB : integer):byte;
procedure WritePCtrl(PortB : Integer; value : byte);

{*****
  IMPLEMENTATION
*****}
implementation

var
  bPrivException : boolean;

{*****
  outportb
-----
  เป็นคำสั่งส่งข้อมูลขนาด 8 บิต ไปที่พอร์ต
*****}
procedure outportb(portid : integer; value : BYTE);
Begin
  asm
    mov edx,portid
    mov al,value
    out dx,al
  end;
end;

{*****
  inportb
-----
  เป็นคำสั่งรับข้อมูลขนาด 8 บิต จากพอร์ต
*****}
function inportb(portid : integer) : byte;
Var value : byte;
Begin
  asm
    mov edx,portid
    in al,dx
    mov value,al
  end;
  inportb := value;
end;

```

```

{*****
      StartUpIoPorts
-----
      เป็นคำสั่งเปิดการทำงานของพอร์ต
      *****}
function StartUpIoPorts(PortToAccess : integer) : boolean;
Var hUserPort : THandle;
Begin
  { 0. ถ้าไม่ใช่ Windows XP ไม่ต้องติดต่อกับ UserPort }
  if (LOBYTE(LOWORD(GetVersion))<5) then
  begin
    StartUpIoPorts := false;
  end
  else
  begin
    { 1. ทำการเรียกใช้งานไดรเวอร์ }
    hUserPort := CreateFile("\\.\UserPort", GENERIC_READ, 0, nil, OPEN_EXISTING,
      FILE_ATTRIBUTE_NORMAL, 0);
    CloseHandle(hUserPort); // Activate the driver
    Sleep(100); // We must make a process switch

    try
      { 2. ทดสอบรับข้อมูลจากพอร์ต }
      inportb(PortToAccess); // Try to access the given port address
      bPrivException := false;
    except
      { 3. ถ้าเกิดความผิดพลาด ให้แจ้งเตือน }
      MessageBox(0, "ไม่สามารถเปิดพอร์ตใช้งานได้', 'เกิดข้อผิดพลาด', MB_OK);
      bPrivException := true;
    end;
    StartUpIoPorts := not bPrivException;
  end;
end;

{*****
      DelayPX
-----
      หน่วงเวลาสำหรับพอร์ตของ 8255
      *****}
procedure DelayPX;
var
  i : integer;
begin
  for i := 1 to 50000 do
  begin
    end;
  end;
end;

```

```

{*****
  WritePA
-----
  เป็นคำสั่งส่งข้อมูลไปที่พอร์ต A ของ 8255
  *****}
procedure WritePA(PortB : Integer; Value : byte);
begin
  outputb(PortB+2, $C xor $B);
  outputb(PortB, value);

  outputb(PortB+2, inportb(PortB+2) xor $08);
  DelayPX;
  outputb(PortB+2, inportb(PortB+2) xor $08);
end;

{*****
  WritePB
-----
  เป็นคำสั่งส่งข้อมูลไปที่พอร์ต B ของ 8255
  *****}
procedure WritePB(PortB : Integer; Value : byte);
begin
  outputb(PortB+2, $D xor $B);
  outputb(PortB, value);

  outputb(PortB+2, inportb(PortB+2) xor $08);
  DelayPX;
  outputb(PortB+2, inportb(PortB+2) xor $08);
end;

{*****
  WritePC
-----
  เป็นคำสั่งส่งข้อมูลไปที่พอร์ต C ของ 8255
  *****}
procedure WritePC(PortB : Integer; Value : byte);
begin
  outputb(PortB+2, $E xor $B);
  outputb(PortB, value);

  outputb(PortB+2, inportb(PortB+2) xor $08);
  DelayPX;
  outputb(PortB+2, inportb(PortB+2) xor $08);
end;

```

```
{*****
ReadPA
```

อ่านข้อมูลจากพอร์ต A ของ 8255

```
*****}
```

```
function ReadPA(PortB : integer):byte;
begin
  { กำหนด Address ให้อยู่ในตำแหน่ง PA }
  outportb(PortB+2, $C xor $B);

  { กำหนดให้ Data Register เป็นอินพุต " C5 = 1" }
  outportb(PortB+2, inportb(PortB+2) or $20);

  { อ่านค่าจาก Data Register }
  outportb(PortB+2, inportb(PortB+2) xor $04);
  Result := inportb(PortB);
  DelayPX;
  outportb(PortB+2, inportb(PortB+2) xor $04);
end;
```

```
{*****
ReadPB
```

อ่านข้อมูลจากพอร์ต B ของ 8255

```
*****}
```

```
function ReadPB(PortB : integer):byte;
begin
  { กำหนด Address ให้อยู่ในตำแหน่ง PB }
  outportb(PortB+2, $D xor $B);

  { กำหนดให้ Data Register เป็นอินพุต " C5 = 1" }
  outportb(PortB+2, inportb(PortB+2) or $20);

  { อ่านค่าจาก Data Register }
  outportb(PortB+2, inportb(PortB+2) xor $04);
  Result := inportb(PortB);
  DelayPX;
  outportb(PortB+2, inportb(PortB+2) xor $04);
end;
```

```
{*****
  ReadPC
-----
```

```
อ่านข้อมูลจากพอร์ต C ของ 8255
*****}
```

```
function ReadPC(PortB : integer):byte;
begin

  { กำหนด Address ให้อยู่ในตำแหน่ง PC }

  outportb(PortB+2, $E xor $B);

  { กำหนดให้ Data Register เป็นอินพุต " C5 = 1" }

  outportb(PortB+2, inportb(PortB+2) or $20);

  { อ่านค่าจาก Data Register }

  outportb(PortB+2, inportb(PortB+2) xor $04);
  Result := inportb(PortB);
  DelayPX;
  outportb(PortB+2, inportb(PortB+2) xor $04);

end;
```

```
{*****
  WritePCtrl
-----
```

```
เป็นคำสั่งส่งข้อมูลไปที่พอร์ตควบคุมของ 8255
*****}
```

```
procedure WritePCtrl(PortB : Integer; value : byte);
begin

  outportb(PortB+2, $F xor $B);
  outportb(PortB, value);

  outportb(PortB+2, inportb(PortB+2) xor $08);
  DelayPX;
  outportb(PortB+2, inportb(PortB+2) xor $08);

End;
```

```

{*****
  StopDriver
-----
  เป็นคำสั่งหยุดการทำงานของไดรเวอร์
  *****}
function StopDriver : boolean;
var
  schService  : SC_HANDLE; { Service Control Handle}
  schSCManager : SC_HANDLE;
  serviceStatus : SERVICE_STATUS;
begin
  { 0. ถ้าไม่ใช่ Windows XP ไม่ต้องติดต่อกับ UserPort }
  if (LOBYTE(LOWORD(GetVersion))<5) then
  begin
    StopDriver := false;
  end
  else
  begin
    { 1. เปิดตัวบริหารควบคุมบริการ (SCManager) }
    schSCManager := OpenSCManager(nil, // machine (NULL == local)
                                   nil, // database (NULL == default)
                                   SC_MANAGER_ALL_ACCESS); // access required

    { 2. ถ้าเปิดใช้งานไม่สำเร็จ จะแจ้งเตือนข้อผิดพลาด }
    if (schSCManager = 0) then
    begin
      Result := false;
    end
    else
    { 3. ถ้าเปิดใช้งานได้ }
    begin
      { 4. เปิดบริการ UserPort }
      schService := OpenService (schSCManager, 'UserPort', SERVICE_ALL_ACCESS);

      { 5. ถ้าบริการ UserPort เปิดขอใช้ไม่สำเร็จ จะทำการปิดการบริหารควบคุมบริการ }
      if (schService = 0) then
      begin
        CloseServiceHandle(schSCManager);
        StopDriver := false;
      end
      { 6. ถ้าเปิดบริการ UserPort สำเร็จ }
      else
      begin
        { 7. สั่งปิดบริการ (SERVICE_CONTROL_STOP) }
        ControlService(schService, SERVICE_CONTROL_STOP, serviceStatus);

        { 8. ลบบริการ }
        DeleteService(schService);

        { 9. ปิดการถือครอง }
        CloseServiceHandle(schService);
        CloseServiceHandle(schSCManager);
        StopDriver := true;
      end; { ? Open service }
    end; { ? Open SC manager }
  end;
end;

```

```

end;

{*****}
      StartDriver
-----
      เป็นคำสั่งให้ดรเวอร์ UserPort.sys ทำงาน
      *****}
function StartDriver:boolean;
var
  schService   : SC_HANDLE;
  schSCManager : SC_HANDLE;
  hDriver      : THandle;
  LastError    : DWORD;
  param        : PChar;
  DriverName   : PChar;
begin
  { 0. ถ้าไม่ใช่ Windows XP ไม่ต้องติดต่อกับ UserPort }
  if (LOBYTE(LOWORD(GetVersion))<5) then
  begin
    StartDriver := false;
  end
  else
  begin
    { 1. เปิดแฟ้มของไดรเวอร์ }
    DriverName := 'C:\Windows\System32\Drivers\UserPort.sys';
    hDriver := CreateFile( DriverName, GENERIC_READ, 0, nil, OPEN_EXISTING,
                          FILE_ATTRIBUTE_NORMAL, 0 );

    { 2. ถ้าเปิดไม่สำเร็จ แสดงว่ายังไม่ได้ติดตั้ง UserPort.sys เอาไว้ใน
      C:\Windows\System32\Drivers\ }
    if (hDriver = INVALID_HANDLE_VALUE) then
    begin
      MessageBox(0,'ยังไม่ได้ติดตั้งแฟ้ม'+#10+#13+
                  '\Windows\System32\Drivers\UserPort.sys',
                  'เกิดข้อผิดพลาดกับ UserPort',MB_OK);
      StartDriver := false;
    end
    else
    { 3. ติดตั้งแฟ้มไดรเวอร์เรียบร้อยแล้ว }
    begin
      { 4. ปิดแฟ้ม }
      CloseHandle(hDriver);

      { 5. เปิดตัวบริหารควบคุมบริการ }
      schSCManager := OpenSCManager(nil, // machine (NULL == local)
                                    nil, // database (NULL == default)
                                    SC_MANAGER_ALL_ACCESS); // access required

      { 6. ถ้าเปิดไม่สำเร็จ จะแจ้งเตือนข้อผิดพลาด }
      if (schSCManager = 0) then
      begin
        { 7. ถ้าความผิดพลาดเกี่ยวกับการสิทธิ์ของการเข้าถึง }
        if (GetLastError()= ERROR_ACCESS_DENIED) then
          MessageBox(0, 'คุณไม่มีสิทธิ์ในการติดตั้งไดรเวอร์'+#10+#13+
                      'กรุณาติดต่อกับผู้บริการระบบเครือข่าย',
                      'เกิดข้อผิดพลาดกับ UserPort',MB_OK)
        else
          StartDriver := true;
        end
      end
    end
  end
end;

```



```

        else
    { 8. ถ้าเป็นกรณีอื่นๆ }
    MessageBox(0,"ไม่สามารถเริ่มต้นการทำงานของไดรเวอร์ได้สำเร็จ',
        'เกิดข้อผิดพลาดกับ UserPort',MB_OK);
    StartDriver := false;

end
else
{ 9. ถ้าเปิดสำเร็จ }
begin
    { 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
        nil, // no load ordering group
        nil, // no tag identifier
        nil, // no dependencies
        nil, // LocalSystem account
        nil); // no password

    { 11. ถ้าสร้างไม่สำเร็จ จะแจ้งข้อผิดพลาดที่เกิดขึ้น }
    if (schService = 0) then
    begin
        { 12. อ่านหมายเลขข้อผิดพลาด }
        LastError := GetLastError();

        { 13. ถ้าเกิดข้อผิดพลาดมีบริการติดตั้งในระบบอยู่ก่อนแล้ว }
        if (LastError = ERROR_SERVICE_EXISTS) then
            MessageBox(0,"ไดรเวอร์ได้ถูกเรียกใช้งานไปก่อนหน้านี้แล้ว',
                'เกิดข้อผิดพลาดกับ UserPort',MB_OK)
        else if (LastError = ERROR_ACCESS_DENIED) then
            MessageBox(0,'คุณไม่มีสิทธิ์ในการติดตั้งไดรเวอร์'+#10+#13+
                'กรุณาติดต่อกับผู้บริการระบบเครือข่าย',
                'เกิดข้อผิดพลาดกับ UserPort',MB_OK)
        else
            MessageBox(0,"ไม่สามารถเริ่มต้นการทำงานของไดรเวอร์ได้',
                'เกิดข้อผิดพลาดกับ UserPort',MB_OK);

            CloseServiceHandle (schSCManager);
            StartDriver := false;

    end
    else
    { 14. สร้างบริการสำเร็จ }
    begin
        { 15. สั่งเริ่มต้นการทำงานของบริการ โดยไม่มีพารามิเตอร์ }
        Param := "";
        StartService (schService, // service identifier
            0, // number of arguments
            Param); // pointer to arguments
    end
end

```

```

{ 16. ปิดการถือครองบริการ }
CloseServiceHandle (schService);
CloseServiceHandle (schSCManager);
    StartDriver := true;
end; { ? Create service. }
end; { ? Open SC manager. }
end; { ? Driver does exist. }
end;
end;

{*****}
    INITIALIZATION
    ทำงานเมื่อเรียกใช้ Unit
    *****}
initialization

    StartDriver;

{*****}
    FINALIZATION
    ทำงานเมื่อ Unit ถูกยกเลิก
    *****}
finalization

    StopDriver;

end.

```

4. ควบคุมหลอด LED

4.1 ตัวอย่างที่ 1 (lab02-led1.rar)

ตัวอย่างนี้เป็นการส่งข้อมูล \$AA ไปที่พอร์ต PA นั้นหมายความว่าเมื่อต่อวงจรเข้ากับบอร์ด ET-EXP4 จะทำให้หลอด LED นั้นติดและดับสลับกัน หน้าตาของตัวโปรแกรมเป็นดังนี้



โค้ดหลักของโปรแกรมเขียนเอาไว้ดังนี้

```

unit UnitMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ioport;

type
  TFormMain = class(TForm)
    Label1: TLabel;
    cbLPT: TComboBox;
    btnSendLED: TButton;
    btnExit: TButton;
    procedure FormCreate(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure btnSendLEDClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  formMain: TFormMain;

implementation

{$R *.dfm}

procedure TFormMain.FormCreate(Sender: TObject);
begin
  cbLPT.Items.Add('LPT1');
  cbLPT.Items.Add('LPT2');
  cbLPT.ItemIndex := 0;
end;

procedure TFormMain.btnExitClick(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TFormMain.btnSendLEDClick(Sender: TObject);
var
  PortB : integer;
begin
  if (cbLPT.ItemIndex = 0) then PortB := $378
  else PortB := $278;

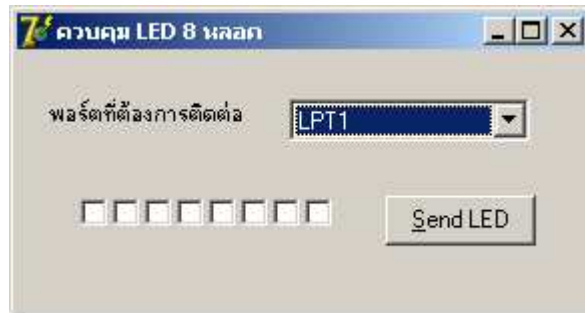
  if (StartUpIOPorts(PortB)) then
  begin
    WritePCtrl(PortB, $80);
    WritePA(PortB, $AA);
  end;
end;

end.

```

4.2 ตัวอย่างที่ 2 (lab02-led2.rar)

ตัวอย่างนี้จะแตกต่างจากโปรแกรมแรกตรงที่จะทำการสร้าง checked box จำนวน 8 อันเพื่อใช้แทนสถานะของ LED แต่ละหลอด ถ้าต้องการให้หลอดใดติดเราก็คลิกที่ช่องนั้น เมื่อเลือกเสร็จแล้วให้คลิกที่ปุ่ม Send LED เพื่อทำการส่งข้อมูลไปที่ PA



โค้ดหลักของโปรแกรมเขียนเอาไว้ดังนี้

```
unit UnitMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ioport;

type
  TFormMain = class(TForm)
    Label1: TLabel;
    cbLPT: TComboBox;
    btnSendLED: TButton;
    cb1: TCheckBox;
    cb2: TCheckBox;
    cb3: TCheckBox;
    cb4: TCheckBox;
    cb5: TCheckBox;
    cb6: TCheckBox;
    cb7: TCheckBox;
    cb8: TCheckBox;
    procedure FormCreate(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure btnSendLEDClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  formMain: TFormMain;
```

```

implementation

{$R *.dfm}

procedure TFormMain.FormCreate(Sender: TObject);
begin
  cbLPT.Items.Add('LPT1');
  cbLPT.Items.Add('LPT2');
  cbLPT.ItemIndex := 0;
end;

procedure TFormMain.btnExitClick(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TFormMain.btnSendLEDClick(Sender: TObject);
var
  PortLPT : integer;
  PortData : Byte;
begin
  if (cbLPT.ItemIndex = 0) then PortLPT := $378
  else PortLPT := $278;

  if (StartUpIOPorts(PortLPT)) then
  begin
    WritePCtrl(PortLPT, $80);
    PortData := $00;
    if cb1.Checked = true then PortData := PortData or $01;
    if cb2.Checked = true then PortData := PortData or $02;
    if cb3.Checked = true then PortData := PortData or $04;
    if cb4.Checked = true then PortData := PortData or $08;
    if cb5.Checked = true then PortData := PortData or $10;
    if cb6.Checked = true then PortData := PortData or $20;
    if cb7.Checked = true then PortData := PortData or $40;
    if cb8.Checked = true then PortData := PortData or $80;
    WritePA(PortLPT, PortData);
  end;
end;
end.

```

4.3 ตัวอย่างที่ 3 (lab02-led3.rar)

ตัวอย่างนี้คล้ายกับตัวอย่างที่ 2 แตกต่างตรงที่ checked box ถูกเปลี่ยนหน้าที่เป็นตัวแสดงสถานะว่าไปหลอดใดติด เพราะการขับเคลื่อน LED นั้นจะเป็นหน้าที่ของ Timer ซึ่งในการตอบสนองของ Timer นั้นจะพบว่า ทุกครั้งที่ Timer ทำงานจะทำการเลื่อนการแสดงผล LED เป็นหลอดถัดไป และทำอย่างนี้ต่อไปเรื่อยๆ ผลที่เราเห็นจึงเหมือนกับว่าเป็นไฟวิ่งนั่นเอง



โค้ดหลักของโปรแกรมเขียนเอาไว้ดังนี้

```
unit UnitMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ioport, ExtCtrls;

type
  TFormMain = class(TForm)
    Label1: TLabel;
    cbLPT: TComboBox;
    btnStart: TButton;
    GroupBox1: TGroupBox;
    cb1: TCheckBox;
    cb2: TCheckBox;
    cb3: TCheckBox;
    cb4: TCheckBox;
    cb5: TCheckBox;
    cb6: TCheckBox;
    cb7: TCheckBox;
    cb8: TCheckBox;
    Timer1: TTimer;
    procedure FormCreate(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure btnStartClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
    PortLPT : integer;
    PortData : Byte;
  public
    { Public declarations }
  end;

var
  formMain: TFormMain;

implementation

{$R *.dfm}
```

```

procedure TFormMain.FormCreate(Sender: TObject);
begin
  cbLPT.Items.Add('LPT1');
  cbLPT.Items.Add('LPT2');
  cbLPT.ItemIndex := 0;
end;

procedure TFormMain.btnExitClick(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TFormMain.btnStartClick(Sender: TObject);
begin
  if Timer1.Enabled = false then
  begin
    if (cbLPT.ItemIndex = 0) then PortLPT := $378
    else PortLPT := $278;
    if (StartUpIOPorts(PortLPT)) then
    begin
      WritePCtrl(PortLPT, $80);
    end;
    Timer1.Enabled := true;
    btnStart.Caption := '&Stop';
    PortData := $01;
  end
  else
  begin
    Timer1.Enabled := false;
    btnStart.Caption := '&Start';
  end;
end;

procedure TFormMain.Timer1Timer(Sender: TObject);
begin
{ Update display }
  if PortData = $01 then cb1.Checked := true
  else cb1.Checked := false;
  if PortData = $02 then cb2.Checked := true
  else cb2.Checked := false;
  if PortData = $04 then cb3.Checked := true
  else cb3.Checked := false;
  if PortData = $08 then cb4.Checked := true
  else cb4.Checked := false;
  if PortData = $10 then cb5.Checked := true
  else cb5.Checked := false;
  if PortData = $20 then cb6.Checked := true
  else cb6.Checked := false;
  if PortData = $40 then cb7.Checked := true
  else cb7.Checked := false;
  if PortData = $80 then cb8.Checked := true
  else cb8.Checked := false;

{ Send data }
  WritePA(PortLPT, PortData);

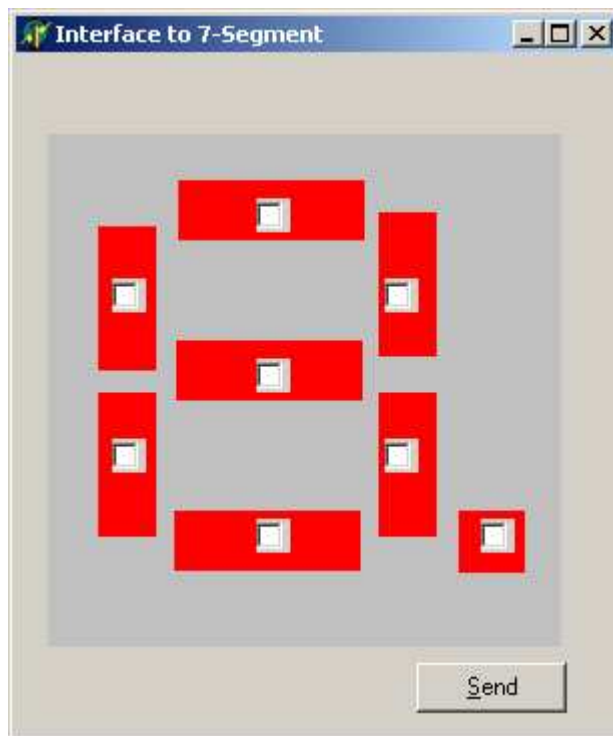
```

```
{ Shifting }  
  PortData := PortData shl 1;  
  if PortData = $0 then PortData := $01;  
end;  
end.
```

5. ควบคุม 7-Segment

5.1 ตัวอย่างที่ 1 (lab02-7seg1.rar)

หลังจากที่ทำการเขียนโปรแกรมขับหลอด LED สำเร็จแล้ว ขั้นตอนต่อไปของการเขียนโปรแกรมคือการควบคุม 7-Segment ซึ่งตัวอย่างนี้เป็นการขับ 7-Segment 1 ตัว โดยอาศัยพอร์ตของ PA และวงจรถอง ET-EXP4 (ถ้าต้องการดูรูปวงจรสามารถหาได้จากบทความของ MCS-51 ครับ) หน้าตาของโปรแกรมแรกเป็นดังรูปด้านล่าง หลักการทำงานของโปรแกรมคือ ผู้ใช้ต้องคลิกที่ checked box ที่จัดวางเอาไว้ตามลักษณะของ 7-Segment แล้วคลิกที่ปุ่ม Send เพื่อควบคุมให้หลอดของ 7-Segment ติดตามลักษณะที่กำหนด



โค้ดหลักของโปรแกรมเขียนเอาไว้ดังนี้


```

unit main7Seg;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IOPort;

type
  TForm1 = class(TForm)
    Image1: TImage;
    I0: TCheckBox;
    I5: TCheckBox;
    I1: TCheckBox;
    I6: TCheckBox;
    I4: TCheckBox;
    I2: TCheckBox;
    I3: TCheckBox;
    I7: TCheckBox;
    btnSend: TButton;
    procedure btnSendClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.btnSendClick(Sender: TObject);
var a:byte;
begin
  if StartupIOPorts($378) = true then
  begin
    WritePCtrl($378, $80);
    a:=0;
    if (I0.Checked) then a:=$01;
    if (I1.Checked) then a:=a or $02;
    if (I2.Checked) then a:=a or $04;
    if (I3.Checked) then a:=a or $08;
    if (I4.Checked) then a:=a or $10;
    if (I5.Checked) then a:=a or $20;
    if (I6.Checked) then a:=a or $40;
    if (I7.Checked) then a:=a or $80;
    WritePA($378, a);
  end;
end;

end.

```

5.2 ตัวอย่างที่ 2 (lab02-7seg2.rar)

ตัวอย่างนี้เป็นการทำตัวเลขวิ่ง คือ เริ่มจาก 0 แล้วเปลี่ยนเป็น 1 แล้วเปลี่ยนเป็น 2 แล้วเปลี่ยนเรื่อยๆ จนถึงเลข 9 โดยการทำงานจะเป็นอัตโนมัติเพราะใช้ Timer เป็นตัวควบคุมการทำงาน หน้าตาของโปรแกรมเป็นดังนี้



โค้ดหลักของโปรแกรมเขียนเอาไว้ดังนี้

```
unit main7Seg;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IOPort;

type
  TForm1 = class(TForm)
    btnConnect: TButton;
    Timer1: TTimer;
    Timer2: TTimer;
    procedure btnConnectClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  a : byte;
  segment : array[0..9] of byte;
  idx : integer;

implementation

{$R *.dfm}

procedure TForm1.btnConnectClick(Sender: TObject);
begin
  if StartUpIOPorts($378) = true then
  begin
    WritePCtrl($378, $80);
```

```

    Timer2.Enabled := true;
    Timer1.Enabled := true;
end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    a:=Segment[idx];
    WritePA($378, a);
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
    idx := idx+1;
    if idx = 10 then idx:=0;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    Segment[0] := $3F;
    Segment[1] := $06;
    Segment[2] := $5B;
    Segment[3] := $4F;
    Segment[4] := $66;
    Segment[5] := $6D;
    Segment[6] := $7D;
    Segment[7] := $07;
    Segment[8] := $7F;
    Segment[9] := $6F;
    idx := 0;
end;
end.

```

5.3 ตัวอย่างที่ 3 (lab02-7seg3.rar)

ตัวอย่างนี้เป็นตัวอย่างสุดท้ายของบทความตอนที่ 2 นี้ครับ เป็นการควบคุม 7-Segment จำนวน 2 หลัก หน้าตาของโปรแกรมเป็นดังรูปด้านล่าง ส่วนหลักของการทำงานนั้นก็คือ

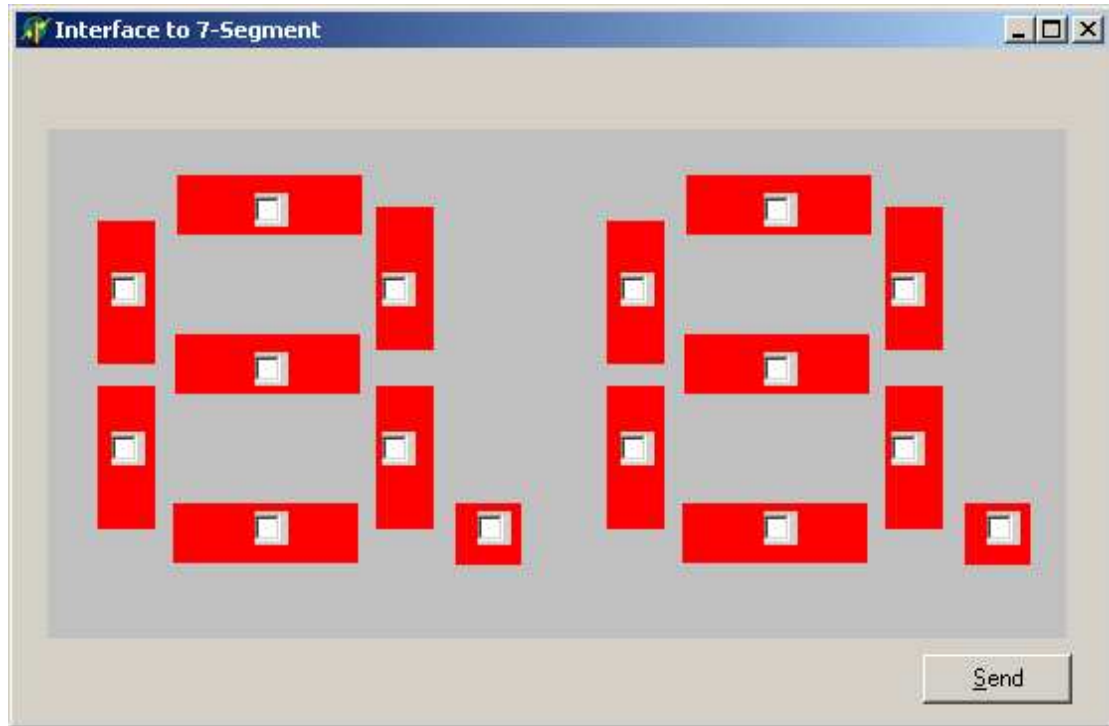
```

WritePA($378, 0);
WritePB($378, 0); { เลือกพอร์ต PB0 }
WritePA($378, a[0]);
Sleep(1);

WritePA($378, 0);
WritePB($378, 1); { เลือกพอร์ต PB1 }
WritePA($378, a[1]);
Sleep(1);

```

จากโค้ดจะเห็นว่าเราต้องเลือก PB0 แล้วส่งข้อมูลสำหรับ 7-Segment ตัวแรกไป PA ก่อนหลังจากนั้น หุขุรอสักนิดหนึ่งแล้วเลือก PB1 แล้วส่งข้อมูลชุดที่ 2 ไปยัง PA แล้วรอสักครู่ ทำสลับกันแบบนี้ตลอดเวลา จะทำให้สิ่งที่เห็นนั้นเหมือนกับ 7-Segment ดิจิตทั้ง 2 หลักในเวลาเดียวกัน



โค้ดหลักของโปรแกรมเขียนเอาไว้ดังนี้

```
unit main7Seg;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IOPort;

type
  TForm1 = class(TForm)
    Image1: TImage;
    I0: TCheckBox;
    I5: TCheckBox;
    I1: TCheckBox;
    I6: TCheckBox;
    I4: TCheckBox;
    I2: TCheckBox;
    I3: TCheckBox;
    I7: TCheckBox;
    Image2: TImage;
    r0: TCheckBox;
    r1: TCheckBox;
    r2: TCheckBox;
    r3: TCheckBox;
    r4: TCheckBox;
    r6: TCheckBox;
    r7: TCheckBox;
    r5: TCheckBox;
    Timer1: TTimer;
    btnSend: TButton;
  end;

end.
```

```

procedure btnSendClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  a :array[0..1] of byte;
  step : integer;

implementation

{$R *.dfm}

procedure TForm1.btnSendClick(Sender: TObject);
begin
  a[0]:=0;
  if (I0.Checked) then a[0]:=$01;
  if (I1.Checked) then a[0]:=a[0] or $02;
  if (I2.Checked) then a[0]:=a[0] or $04;
  if (I3.Checked) then a[0]:=a[0] or $08;
  if (I4.Checked) then a[0]:=a[0] or $10;
  if (I5.Checked) then a[0]:=a[0] or $20;
  if (I6.Checked) then a[0]:=a[0] or $40;
  if (I7.Checked) then a[0]:=a[0] or $80;

  a[1]:=0;
  if (r0.Checked) then a[1]:=$01;
  if (r1.Checked) then a[1]:=a[1] or $02;
  if (r2.Checked) then a[1]:=a[1] or $04;
  if (r3.Checked) then a[1]:=a[1] or $08;
  if (r4.Checked) then a[1]:=a[1] or $10;
  if (r5.Checked) then a[1]:=a[1] or $20;
  if (r6.Checked) then a[1]:=a[1] or $40;
  if (r7.Checked) then a[1]:=a[1] or $80;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  if StartUpIOPorts($378) = true then
  begin
    WritePCtrl($378, $80);
    step := 0;
    btnSendClick(Sender);
    Timer1.Enabled := true;
  end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  WritePA($378, 0);
  WritePB($378, 0); { เลือกพอร์ต PBO }
  WritePA($378, a[0]);

```

```
Sleep(1);  
  
WritePA($378, 0);  
WritePB($378, 1); { เลือกพอร์ต PB1 }  
WritePA($378, a[1]);  
Sleep(1);  
end;  
  
end.
```

6. สรุป

จากบทความตอนนี้จะพบว่าเราสามารถสั่งงานอุปกรณ์พื้นฐานกันได้แล้ว ซึ่งการขับหลอด LED นี้ยังถือได้ว่าเป็นพื้นฐานของการขับมอเตอร์แบบ Stepping เลยทีเดียวครับ (ลองอ่านบทความของ MCS-51 ประกอบนะครับ) ส่วนคราวหน้าจะเป็นบทความสุดท้ายของบทความชุดนี้นะครับ คือ จะเป็นเรื่องของ DIP Switch, Scan Key และ Stepping Motor โดยวงจรมันผมใช้กับบอร์ด ET-EXP4 เช่นเดิมครับ (วงจรมหาได้จากหน้า product ของ ET-EXP4 หรือหาจากบทความ MCS-51 ก็ได้ครับ)

สุดท้ายนี้ต้องขอขอบคุณทาง ETT โดยเฉพาะคุณกอบกิจ เป็นอย่างสูงครับที่ทำให้บทความนี้ยังได้อ่านกันต่อไป ส่วนครั้งหน้าถ้าไม่ผิดพลาดอะไรผมจะอัพบทความให้ตอนกลางเดือนนะครับ เพราะวันที่ 29-30 ตุลาคม 2548 ผมมีสอบปลายภาค และช่วงสัปดาห์ที่ 1-2 เป็นวันเวลาที่ผมหมอนัดภรรยาผมให้ไปคลอดครับ (ใช้ได้เป็นพ่อคนแล้ว) เลยไม่แน่ใจว่าจะปั่นบทความเสร็จทันหรือไม่ ผิดพลาดประการใดต้องขออภัยเอาไว้ล่วงหน้าด้วยนะครับ