พอร์ตขนานและการเชื่อมต่อ ด้วย Borland Delphi ตอนที่ 3

ศุภชัย บุศราทิจ (raek@etteam.com)
คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยราชภัฏเพชรบุรี

1. บทนำ

ก่อนอื่นต้องขออภัยที่บทความตอนนี้ออกมาค่อนข้างช้า สาเหตุเนื่องมาจากการเป็นพ่อคนตั้งแต่ปลายเดือนตุลาคมเป็นต้นมาทำให้ครอบครัวของผม ซึ่งประกอบด้วย ภรรยา และผม (กับพุดเดิ้ลและชิสุ) มีภาระมากขึ้น (แต่สนุกและมีความสุขนะครับ) การแบ่งเวลาเลยไม่ลงตัวนัก ผมเองก็พยายามเขียนบทความนี้มาหลายต่อหลายครั้งแต่ก็เริ่มได้นิดหน่อยก็ต้องวางมือไปทำเรื่องเฉพาะหน้าก่อน (ลูกร้องน่ะครับ) อย่างไรคงไม่ว่ากันนะครับ

จากบทความ 2 ตอนก่อนหน้านี้เราสามารถเขียนโปรแกรมเพื่อเป็น Output และสามารถเขียนโปรแกรมแยกแยะระบบปฏิบัติการพร้อมทั้งเลือกการทำงานได้อย่างถูกต้องคือ ต้องโหลด UserPort.SYS ถ้าเป็นวินโดวส์ XP และไม่ต้องโหลด UserPort.SYS ถ้าเป็นวินโดวส์ 98 กันไปแล้ว บทความในตอนนี้จะเป็นตอนส่งท้ายของเรื่องพอร์ตขนานซึ่งเป็นตัวอย่างเกี่ยวกับการติดต่อกับอุปกรณ์ 3 ชนิดคือ DIP Switch, Scan Key และ Stepping motor

นอกจาก 3 อุปกรณ์นี้แล้วใจจริงผู้เขียนต้องการสร้างตัวอย่างเกี่ยวกับการเชื่อมต่อ Scan Key, 7 Segment และระบบฐานข้อมูลเพื่อเป็นตัวอย่างให้เห็นแนวทางสำหรับการนำไปประยุกต์ใช้งานจริง แต่ไม่มั่นใจว่าเนื้อหาจะเขียนได้ครอบคลุมหรือไม่ เพราะลำพังเพียงเรื่องระบบฐานข้อมูลถ้าผู้อ่านไม่ได้ศึกษามาก่อนจะทำความเข้าใจได้ยากมากเรื่องหนึ่งเลยทีเดียว ส่วนในการทำงานของภาคฮาร์ดแวร์นั้นจะง่ายกว่าเนื่องจากสามารถนำโปรแกรมตัวอย่างมาปรับปรุงการทำงานได้ แต่อย่างไรก็ดีผมขอเวลาทดสอบสักนิดหนึ่งล่ะกันครับ คือ ถ้าฐานข้อมูลดูยุ่งยากเกินไป ผมอาจจะเปลี่ยนมาเป็นสร้างโปรแกรมจัดการกับแฟ้มขึ้นมาเอง ซึ่งจะง่ายกว่าการใช้ฐานข้อมูลค่อนข้างมาก (ในแง่การติดตั้งและใช้งานระบบฐานข้อมูล) ครั้งหน้าเมื่อได้อ่านบทความคงได้ทราบกันล่ะครับว่าผมได้ตัดสินใจอย่างไร

2. ฮาร์ดแวร์และซอฟต์แวร์ที่เกี่ยวข้อง

มีหลายคนถามผมว่าในการพัฒนานั้นเลือกใช้ระบบของฮาร์ดแวร์เป็นอะไรและใช้ซอฟต์แวร์อะไรบ้าง ผมก็ขอกล่าวถึงในหัวข้อนี้ไปเลยก็แล้วกันนะครับ

2.1 ฮาร์ดแวร์

1) Note Book ที่ใช้ทดสอบโปรแกรมและประกอบการสอนเป็น IBM ThinkPAD หน่วยความจำ 128MB กับฮาร์ดดิสก์ความจุ 15GB

2) เครื่องที่บ้าน เป็น Celeron 2.6GHz (ตัวอื่นๆคืนหน่วยงานไปแล้วครับ) หน่วยความจำ 512MB การ์ดแสดงผลเป็น nVidia GeForce 6600GT ฮาร์ดดิสก์ความจุ 40GB ตัวเครื่องทั้งชุด (ยกเว้น RAM ที่เพิ่มกับ VGA) เป็นของ Liberta ราคาที่ซื้อมาตอนนั้น (2 ปีก่อน) รวม VAT แล้วก็ 26000 บาท (ไม่รวมการ์ดแสดงผลและแรมที่เพิ่มนะครับ ... อันนี้ก็ใช้เงินสะสมที่ได้จากการสอนพิเศษบวกกับจากอีทีที จึงได้เครื่องนี้มาครอบครอง) นอกจากนี้ผมยังมีอีกเครื่องหนึ่งครับ เป็น AMD64 อันนี้เอาไว้สำหรับทดสอบงานหนักๆ โดยเฉพาะเกี่ยวกับ Linux + OpenGL ซึ่งเมื่อก่อนผมจะมีเว็บด้านนี้อยู่ แต่ปัจจุบันหายไปหมดแล้วเพราะไม่มี Server จะใช้งาน (เฮ้อ... ทำอะไรก็ร่วงหมดเลยเรา)

3) บอร์ดติดต่อ LPT เป็นของ ETT ครับ

4) บอร์ด I/O ผมใช้ ET-EXP4 ที่ทาง ETT มอบมาให้ทำการทดลอง

5) กล้องถ่ายภาพ เป็นของ Olympus รุ่น u300 ความละเอียดอยู่ที่ 3.2 ล้านจุด ซึ่งซื้อมาหลายปีแล้วครับ (ตอนนั้นราคา 17000 บาท ใช้เงินที่สะสมจากค่าเขียนบทความและทำเว็บของอีทีทีกับการสอนพิเศษอีกเช่นเคยครับ) เป็นกล้องที่ผมถือว่าใช้งานคุ้มมากๆ

6) นอกจากกล้องถ่ายรูปแล้วผมก็มี Web Camera ของ Intel (ซื้อมาตัวละ 4000 บาท) ซึ่งก็ไม่ค่อยได้ใช้แล้ว (เสียดายเงินเหมือนกัน) เพราะคุณภาพของภาพนั้นไม่ค่อยดีนัก (ซื้อมาตั้ง 5 ปีแล้วนี่นา) เนื่องจากซอฟต์แวร์ของกล้องนั้นออกแบบมาเพื่อทำงานได้ดีบน Windows 98 พอเปลี่ยนมาใช้กับ XP ซอฟต์แวร์มันทำงานไม่ได้ เลยได้ภาพไม่สวยนัก ... เกือบลืมไปครับ ผมมีกล้องอีก 2 ตัว เป็นแบบ Wireless ซึ่งได้มาจาก ETT มันก็ทำงนได้ดีครับ แต่ผมต้องใช้กับ Frame Grabber (เป็นของ Eurasys ราคา 23000 บาท) ซึ่งตอนนี้ก็ไม่ค่อยได้ใช้เช่นเดียวกัน เพราะเครื่องคอมพิวเตอร์ที่ผมใช้นั้นสล็อต PCI มันไม่พอใช้งาน (มี 3 สล็อต ซึ่งผมเสียบการ์ดเครือข่าย, โมเด็ม และการ์ดเสียงมันก็หมดพื้นที่แล้ว)

จากสเปคเครื่องและอุปกรณ์ควบคุมจะพบว่าไม่จำเป็นต้องมี PC ความเร็วสูงหรือราคาแพงๆ เราก็สามารถเขียนโปรแกรมได้เหมือนกัน สาเหตุที่ผมเพิ่มแรมกับการ์ดแสดงผลก็เพราะเอาไว้เขียนโปรแกรมด้าน

คอมพิวเตอร์กราฟิก กับเล่นเกม (เล่นเฉพาะยามเครียดหาทางออกไม่ค่อยได้ มักผ่อนคลายด้วยเกม แต่ก็เล่นได้ไม่เกิน 3 ชม. ก็เบื่อครับ ฮ่ะๆๆ) เท่านั้นครับ และจากประสบการณ์พบว่าอย่ารีบซื้อฮาร์ดแวร์เกินไปครับ ควรสอบถามหรือดูสถานการ์ให้ดีก่อนแล้วค่อยตัดสินใจ เพราะผมเองนี่เสียเงินไปเยอะ แล้วค่อยมาเสียดายว่าไม่คุ้มๆ อะไรปานนั้นเลยครับ แต่ยังโชคดีที่อุปกรณ์ทดลองนั้น ETT ส่งมาให้เลยได้ใช้รุ่นที่มีคุณสมบัติลงตัว ไม่ต้องซื้อกวาดทุกรุ่นเพื่อดูว่าอะไรดีหรือไม่ดี สรุปได้ว่าการทดลองนี่เปลืองเงินไม่ใช่เล่นถ้าไม่มีผู้สนับสนุนครับ

2.2 ซอฟต์แวร์

1) ตกแต่งภาพ ผมใช้ TheGIMP ครับ เมื่อก่อนใช้ PhotoShop 5 เป็นส่วนใหมญ่

2) พิมพ์เอกสารก็ใช้ NotePad ก่อน แล้วก็มาจัดใน OpenOffice หรือ MS-Word ครับ

3) ตัวแปลภษาผมใช้ Delphi 7 Professional ที่ทางคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเพชรบุรี ซื้อมาให้คณาจารย์ศึกษาค้นคว้าครับ ส่วนเครื่องที่บ้านผมนั้นจะเป็น Delphi 6 Personal ที่โหลดมาฟรีจาก Borland

4) ระบบปฏิบัติการ ผมใช้ Microsoft Windows XP Home Edition ที่มากับ NoteBook ครับ ส่วนเครื่องตั้งโต๊ะผมเองเป็น XP รุ่น Home เช่นเดียวกัน แต่ตอนนี้มัน Activate ไม่ได้แล้วเนื่องจากผม Format เครื่องบ่อย มันเลยบอกว่ารหัสของผมนั้นมีการ Activate บ่อยเกินไป (เลยต้องอาศัยพลังด้านมืดเพื่อให้มันทำงานได้ต่อไป ฮ่ะๆ) กำลังคิดว่าจะเปลี่ยนไปใช้ Linux ดีไหม ผมก็ติดตรงที่ว่าการเขียนโปรแกรมทดสอบกับฮาร์ดแวร์ของผมนั้นต้องใช้วินโดวส์ หรือไม่อย่างนั้นผมก็เลิกเขียนบทความด้านนี้ซะ (ฮ่ะๆๆ ซึ่งก็ไม่แน่เหมือนกันนะครับ เนื่องจากไฟมอดลงไปเยอะเลย ประกอบกับไปไหนไม่ได้ เลยไม่มีความรู้ใหม่มาเล่าสู่กันฟัง.... เงินนี่เป็นอุปสรรคสำคัญจริงๆ เฮ้อ) เพราะ NoteBook ที่ผมใช้นั้นไม่มีพอร์ต RS232 (ไม่ได้ซื้อด้วยแหละครับ เพราะช่วงที่ผ่านมาการเงินติดขัดนิดหน่อย การใช้จ่ายจึงต้องขออนุมัติเป็นงวดๆไปครับ ...)

5) โปรแกรมแปลง PDF นั้นผมใช้ PDF Creator ซึ่งเป็นของฟรีครับผม

6) โปรแกรมสร้างเว็บ ตอนแรกผมใช้ FrontPage สำหรับสร้างโครงเอกสารครับ แต่ทุกวันนี้ผมใช้ NotePad แล้วเขียน Tag ภาษา HTML เอง ตอนนี้ก็กำลังดูเครื่องมือตัวใหม่ที่ไม่เสียเงินทองอยู่ครับ

7) Web Server อันนี้สำคัญมากครับเพราะใช้สำหรับทดสอบเว็บว่าทำงานถูกต้องหรือไม่ ตัวที่ผมใช้ทดสอบเป็น Apache 2.0.5 ที่ติดตั้ง MySQL กับ PHP5.0.5 เอาไว้ด้วย แต่ผมใช้เฉพาะคุณสมบัติของ Apache เท่านั้น เนื่องจากบนเว็บ ETT ผมเคยลองใช้ PHP แล้วมันหน่วงๆ เลยหลบๆมันไปซะ ส่วน MySQL นั่นผมเข้าไปใช้ไม่ได้ (Telnet ไม่สำเร็จ) ลูกเล่นเกี่ยวกับ Database และ PHP เลยไม่มีในเว็บ ETT แต่มันก็เร็วขึ้น แต่ก็ต้องแลกกับการอัพข้อมูลเป็นแฟ้มปริมาณมาก เวลาเสียหรือผิดพลาดต้องตรวจสอบกันเหนื่อย (^_^)

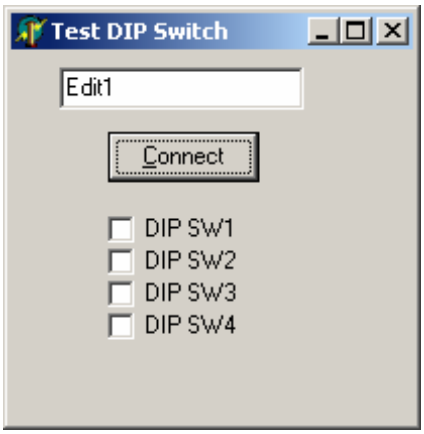
8) AntiVirus อันนี้สำคญมากครับ เพราะเวลาเข้าไปอ่าน e-mail หรือดาวน์โหลดโปรแกรมมักจะเสี่ยงต่อการติดไวรัส ตัวป้องกันและฆ่าไวรัสที่ผมเลือกใช้เป็นของฟรีเช่นเดียวกัน แต่คุณสมบัติก็ทำงานได้ดีทีเดียว ตัวซอฟต์แวร์ที่ผมใช้ก็คือ AVG AntiVirus รุ่น Free Edition

เรื่องของซอฟต์แวร์นี่จะหนักหนากว่าฮาร์ดแวร์เพราะถ้าเราเลือกใช้ซอฟต์แวร์ที่ละเมิดลิขสิทธิ์กันมากๆ ก็เป็นการทำร้ายบริษัทที่สร้างมันขึ้นมา แต่ถ้าซื้อถูกต้องทั้งหมดมันก็หมดตัว (แถมเป็นหนี้อีก) อันนี้ต้องชั่งใจดีๆ ครับ ถ้าใช้ของไม่ถูกต้องแต่เป็นไปตามกฎหมายคือ ทำเพื่อศึกษาค้นคว้า (ไม่เกี่ยวกับหาเงิน ... อย่างของผมนี่

ศึกษาค้นคว้านะครับ ผมไม่มี Job เงินๆทองๆเกี่ยวกับการเขียนโปรแกรมขายกับเขาหรอก) แต่ถ้าทำเงินได้จากซอฟต์แวร์ก็แนะนำว่าน่าจะสะสมเงินซื้อเอาไว้ใช้งานก็ดีครับ เพราะผมยังใช้ Visual C++ Professional ที่ซื้อมาเมื่อหลายปีอยู่เลยครับ ไม่จำเป็นต้องเปลี่ยนไปใช้ของใหม่ก็ได้ ถ้าที่มีอยู่มันดีอยู่แล้ว

เอาล่ะนอกเรื่องไปหลายหน้ากระดาษแล้ว แต่คงพอจะเป็นประโยชน์กันบ้างล่ะ มาเข้าเรื่องกันดีกว่าครับ

3. ตัวอย่างการติดต่อกับ DIP Switch



ตัวอย่างนี้เป็นการอ่านข้อมูลจาก DIP Switch ของบอร์ด ET-EXP4 โดยการอ่านนั้นจะอ่านผ่านทางพอร์ต C ของ 8255 และในการอ่านนั้นจะทำการ AND บิตกับข้อมูลด้วย $F0 ซึ่งหมายความว่าเราไม่สนใจข้อมูลบิต 0 ถึง 3 เพราะในวงจรนั้นบิตเหล่านี้ไม่ได้เชื่อมต่อเข้ากับ DIP Switch โค้ดสำหรับการอ่านและ and บิต เขียนได้ดังนี้

```
a := ReadPC($378) and $F0;
```

เมื่อได้ข้อมูลเก็บในตัวแปร A ขั้นตอนต่อไปของเราคือแปลงข้อมูลเป็นข้อความเพื่อส่งไปแสดงผลที่ Text Box หลังจากนั้นจึงทำการตรวจสอบว่าสวิทช์ตัวใดบ้างที่มีสถานะเป็น on แล้วสั่งให้ Checked Box ของสวิทช์ตัวนั้นถูกกากบาท โค้ดขอส่วนนี้เขียนได้ดังต่อไปนี้

```
edit1.Text := IntToStr(a);
dip1.Checked := false;
dip2.Checked := false;
dip3.Checked := false;
dip4.Checked := false;
```

มีต่อ>>

```
if (a and $10)=$10 then dip1.Checked := true;
if (a and $20)=$20 then dip2.Checked := true;
if (a and $40)=$40 then dip3.Checked := true;
if (a and $80)=$80 then dip4.Checked := true;
```

คำสั่งที่ใช้ในการแปลงตัวเลขเป็นข้อความมีรูปแบบดังนี้

ตัวแปรสตริง := IntToStr(ตัวเลขหรือตัวแปรที่เก็บค่าตัวเลข);

โค้ดการทำงานของโปรแกรมตัวอย่างทั้งหมดเป็นดังนี้

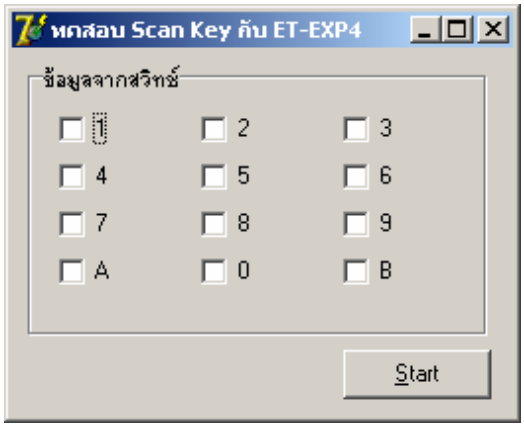
```
unit dipSWMain;

interface

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

type
  TForm1 = class(TForm)
    Button1: TButton;
    Timer1: TTimer;
    dip1: TCheckBox;
    dip2: TCheckBox;
    dip3: TCheckBox;
    dip4: TCheckBox;
    Edit1: TEdit;
    procedure Button1Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
```

มีต่อ>>

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    if StartUpIOPorts($378) = true then
    begin
        WritePCtrl($378, $89);
        Timer1.Enabled := true;
    end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var
   a : byte;
begin
   a := ReadPC($378) and $F0;
   edit1.Text := IntToStr(a);
   dip1.Checked := false;
   dip2.Checked := false;
   dip3.Checked := false;
   dip4.Checked := false;
   if (a and $10)=$10 then dip1.Checked := true;
   if (a and $20)=$20 then dip2.Checked := true;
   if (a and $40)=$40 then dip3.Checked := true;
   if (a and $80)=$80 then dip4.Checked := true;
end;

end.
```

เมื่อโปรแกรมทำงานบนจอแสดงผลจะแสดงให้ผู้ใช้ทราบว่าสวิทช์ตัวใดมีสถานะเป็น On หรือ Off โดยดูจาก Checked Box คือ ถ้า Checked Box อันใดมีเครื่องหมายกากบาทจะหมายความว่าสถานะเป็น On แต่ถ้าไม่มีแสดงว่าสถานะเป็น Off

4. ตัวอย่างการติดต่อกับ Scan Key



ตัวอย่างนี้จะยากกว่าการอ่านข้อมูลจาก DIP Switch เพราะวงจรมีการต่อสวิทช์ที่ซับซ้อน ในการทำงานของโปรแกรมตัวอย่างนี้จะทำการตรวจสอบการกดสวิทช์โดยใช้อัลกอริทึ่มหรือลำดับการทำงานดังนี้

1. ทดสอบคอลัมน์ที่ 1
2. อ่านข้อมูลจากพอร์ต C
3. เลื่อนบิตไปทางขวา 4 บิต
4. ถ้าข้อมูลจากข้อ 3 เป็น 1 แสดงว่ากดปุ่ม 1
5. ถ้าข้อมูลจากข้อ 3 เป็น 2 แสดงว่ากดปุ่ม 4
6. ถ้าข้อมูลจากข้อ 3 เป็น 4 แสดงว่ากดปุ่ม 7
7. ถ้าข้อมูลจากข้อ 3 เป็น 8 แสดงว่ากดปุ่ม A
8. ทดสอบคอลัมน์ที่ 2
9. อ่านข้อมูลจากพอร์ต C
10. เลื่อนบิตไปทางขวา 4 บิต
11. ถ้าข้อมูลจากข้อ 9 เป็น 1 แสดงว่ากดปุ่ม 2
12. ถ้าข้อมูลจากข้อ 9 เป็น 2 แสดงว่ากดปุ่ม 5
13. ถ้าข้อมูลจากข้อ 9 เป็น 4 แสดงว่ากดปุ่ม 8
14. ถ้าข้อมูลจากข้อ 9 เป็น 8 แสดงว่ากดปุ่ม 0
15. ทดสอบคอลัมน์ที่ 3
16. อ่านข้อมูลจากพอร์ต C
17. เลื่อนบิตไปทางขวา 4 บิต
18. ถ้าข้อมูลจากข้อ 17 เป็น 1 แสดงว่ากดปุ่ม 3
19. ถ้าข้อมูลจากข้อ 17 เป็น 2 แสดงว่ากดปุ่ม 6
20. ถ้าข้อมูลจากข้อ 17 เป็น 4 แสดงว่ากดปุ่ม 9
21. ถ้าข้อมูลจากข้อ 17 เป็น 8 แสดงว่ากดปุ่ม B

โค้ดโปรแกรมของตัวอย่างนี้คือ

```
unit UnitScanKey;

interface

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

type
  TFormManScanKey = class(TForm)
    GroupBox1: TGroupBox;
```

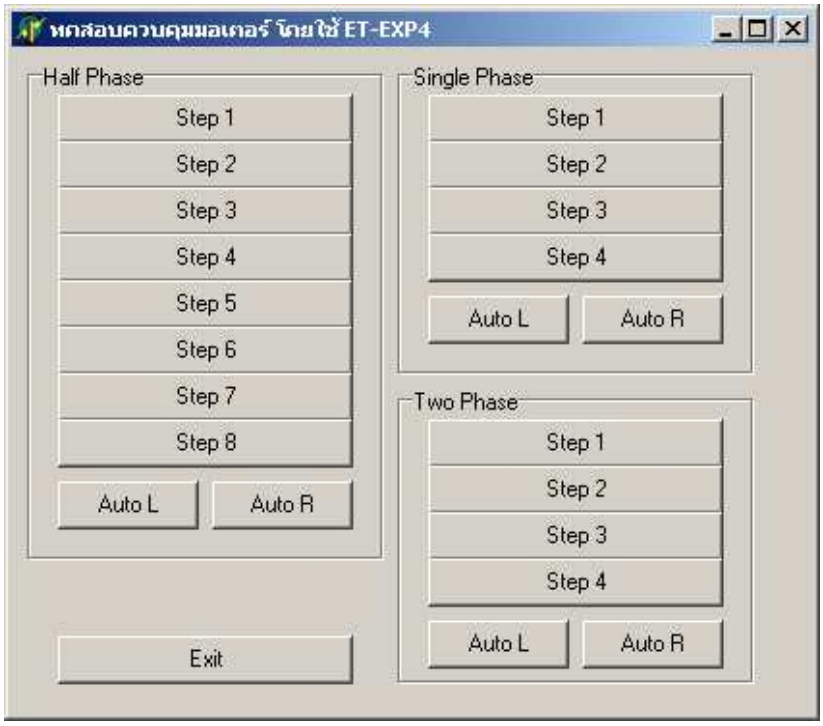
มีต่อ>>

```
    btnStart: TButton;
    cb1: TCheckBox;
    cb2: TCheckBox;
    cb3: TCheckBox;
    cb4: TCheckBox;
    cb5: TCheckBox;
    cb6: TCheckBox;
    cb7: TCheckBox;
    cb8: TCheckBox;
    cb9: TCheckBox;
    cba: TCheckBox;
    cb0: TCheckBox;
    cbb: TCheckBox;
    Timer1: TTimer;
    procedure btnStartClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormManScanKey: TFormManScanKey;

implementation

{$R *.dfm}

procedure TFormManScanKey.btnStartClick(Sender: TObject);
begin
  if StartUpIoPorts($378) = true then
  begin
    WritePCtrl($378, $88);
    Timer1.Enabled := true;
  end;
end;

procedure TFormManScanKey.Timer1Timer(Sender: TObject);
var

  key : byte;
begin
{ ทดสอบกับคอลัมน์ที่ 1 (PC0) หรือ 0000-0110 }
  WritePC($378, $06);
  key := ReadPC($378);
  Key := $0F - (Key shr 4); // เลื่อนบิตไปทางขวา 4 บิต

  { ทดสอบกับปุ่ม 1 }
  if Key = $1 then cb1.Checked := true
  else cb1.Checked := false;
```

มีต่อ>>

```
  { ทดสอบกับปุ่ม 4 }
  if Key = $02 then cb4.Checked := true
  else cb4.Checked := false;

  { ทดสอบกับปุ่ม 7 }
  if Key = $04 then cb7.Checked := true
  else cb7.Checked := false;

  { ทดสอบกับปุ่ม A }
  if Key = $08 then cba.Checked := true
  else cba.Checked := false;

{ ทดสอบกับคอลัมน์ที่ 2 (PC1) หรือ 0000-0101 }
  WritePC($378, $05);
  key := ReadPC($378);
  Key := $0F - (Key shr 4); // เลื่อนบิตไปทางขวา 4 บิต

  { ทดสอบกับปุ่ม 2 }
  if Key = $1 then cb2.Checked := true
  else cb2.Checked := false;

  { ทดสอบกับปุ่ม 5 }
  if Key = $02 then cb5.Checked := true
  else cb5.Checked := false;

  { ทดสอบกับปุ่ม 8 }
  if Key = $04 then cb8.Checked := true
  else cb8.Checked := false;

  { ทดสอบกับปุ่ม 0 }
  if Key = $08 then cb0.Checked := true
  else cb0.Checked := false;

{ ทดสอบกับคอลัมน์ที่ 3 (PC2) หรือ 0000-0011 }
  WritePC($378, $03);
  key := ReadPC($378);
  Key := $0F - (Key shr 4); // เลื่อนบิตไปทางขวา 4 บิต

  { ทดสอบกับปุ่ม 3 }
  if Key = $1 then cb3.Checked := true
  else cb3.Checked := false;

  { ทดสอบกับปุ่ม 6 }
  if Key = $02 then cb6.Checked := true
  else cb6.Checked := false;

  { ทดสอบกับปุ่ม 9 }
  if Key = $04 then cb9.Checked := true
  else cb9.Checked := false;

  { ทดสอบกับปุ่ม B }
  if Key = $08 then cbb.Checked := true
  else cbb.Checked := false;
end;
end.
```

5. ตัวอย่างการติดต่อกับ Stepping Motor



การที่จะทำให้สเต็ปเปอร์มอเตอร์ทำงานได้ หรือหมุนไปในทิศทางเดียวกันนั้น จะต้องอาศัยการส่งสัญญาณเป็นเฟสแก่ขดลวด วิธีการกระตุ้นเฟส เพื่อขับให้สเต็ปเปอร์มอเตอร์ขับเคลื่อนไปได้นั้นมีด้วยกัน 3 วิธี คือ

- กระตุ้นเฟสเดี่ยว (Single phase)
- กระตุ้นสองเฟส (Two phase)
- กระตุ้นแบบครึ่งสเต็ป (Half step)

ดังตารางต่อไปนี้

สเต็ป	เฟส-1	เฟส-2	เฟส-3	เฟส-4
1	1	-	-	-
2	-	1	-	-
3	-	-	1	-
4	-	-	-	1

ตารางที่ 3-1 แสดงลำดับการกระตุ้นเฟสแบบเฟสเดี่ยว

สเต็ป	เฟส-1	เฟส-2	เฟส-3	เฟส-4
1	1	1	-	-
2	-	1	1	-
3	-	-	1	1
4	1	-	-	1

ตารางที่ 3-2 แสดงลำดับการกระตุ้นเฟสแบบสองเฟส

สเต็ป	เฟส-1	เฟส-2	เฟส-3	เฟส-4
1	1	-	-	-
2	1	1	-	-
3	-	1	-	-
4	-	1	1	-
5	-	-	1	-
6	-	-	1	1
7	-	-	-	1
8	1	-	-	1

ตารางที่ 3-3 แสดงลำดับการกระตุ้นเฟสแบบครึ่งเฟส

สำหรับการหมุนมอเตอร์ในทิศที่ตรงกันข้ามนั้นสามารถทำได้โดยการส่งสเต็ปในด้านที่กลับกัน ตัวอย่างการขับมอเตอร์แบบเฟสเดี่ยวเขียนได้ดังนี้

```
procedure TForm1.btnSPRClick(Sender: TObject); { หมุนไปทางขวา }
begin
  btnSP1Click(Sender); Sleep(100);
  btnSP2Click(Sender); Sleep(100);
  btnSP3Click(Sender); Sleep(100);
  btnSP4Click(Sender); Sleep(100);
end;

procedure TForm1.btnSPLClick(Sender: TObject); { หมุนไปทางซ้าย }
begin
  btnSP4Click(Sender); Sleep(100);
  btnSP3Click(Sender); Sleep(100);
  btnSP2Click(Sender); Sleep(100);
  btnSP1Click(Sender); Sleep(100);
end;
```

ในการขับมอเตอร์แบบสองเฟสเขียนได้ดังนี้

```
procedure TForm1.btnTPRClick(Sender: TObject); { หมุนไปทางขวา }
begin
  btnTP1Click(Sender); Sleep(100);
  btnTP2Click(Sender); Sleep(100);
  btnTP3Click(Sender); Sleep(100);
  btnTP4Click(Sender); Sleep(100);
end;

procedure TForm1.btnTPLClick(Sender: TObject); { หมุนไปทางซ้าย }
begin
  btnTP4Click(Sender); Sleep(100);
  btnTP3Click(Sender); Sleep(100);
```

มีต่อ>>

```
  btnTP2Click(Sender); Sleep(100);
  btnTP1Click(Sender); Sleep(100);
end;
```

การขับมอเตอร์แบบครึ่งเฟสเขียนได้ดังนี้

```
procedure TForm1.btnHPRClick(Sender: TObject); { หมุนไปทางขวา }
begin
  btnHP1Click(Sender); Sleep(100);
  btnHP2Click(Sender); Sleep(100);
  btnHP3Click(Sender); Sleep(100);
  btnHP4Click(Sender); Sleep(100);
  btnHP5Click(Sender); Sleep(100);
  btnHP6Click(Sender); Sleep(100);
  btnHP7Click(Sender); Sleep(100);
  btnHP8Click(Sender); Sleep(100);
end;

procedure TForm1.btnHPLClick(Sender: TObject); { หมุนไปทางซ้าย }
begin
  btnHP8Click(Sender); Sleep(100);
  btnHP7Click(Sender); Sleep(100);
  btnHP6Click(Sender); Sleep(100);
  btnHP5Click(Sender); Sleep(100);
  btnHP4Click(Sender); Sleep(100);
  btnHP3Click(Sender); Sleep(100);
  btnHP2Click(Sender); Sleep(100);
  btnHP1Click(Sender); Sleep(100);
end;
```

โค้ดโปรแกรมทั้งหมดเป็นดังต่อไปนี้

```
unit StepMain;

interface

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

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
```

มีต่อ>>

```
    btnHP1: TButton;
    btnHP2: TButton;
    btnHP3: TButton;
    btnHP4: TButton;
    btnHP5: TButton;
    btnHP6: TButton;
    btnHP7: TButton;
    btnHP8: TButton;
    btnHPL: TButton;
    btnHPR: TButton;
    btnSP1: TButton;
    btnSP2: TButton;
    btnSP3: TButton;
    btnSP4: TButton;
    btnSPL: TButton;
    btnSPR: TButton;
    btnTP1: TButton;
    btnTP2: TButton;
    btnTP3: TButton;
    btnTP4: TButton;
    btnTPL: TButton;
    btnTPR: TButton;
    btnExit: TButton;
    procedure FormActivate(Sender: TObject);
    procedure btnStopClick(Sender: TObject);
    procedure btnSP1Click(Sender: TObject);
    procedure btnSP2Click(Sender: TObject);
    procedure btnSP3Click(Sender: TObject);
    procedure btnSP4Click(Sender: TObject);
    procedure btnSPRClick(Sender: TObject);
    procedure btnSPLClick(Sender: TObject);
    procedure btnTP1Click(Sender: TObject);
    procedure btnTP2Click(Sender: TObject);
    procedure btnTP3Click(Sender: TObject);
    procedure btnTP4Click(Sender: TObject);
    procedure btnTPRClick(Sender: TObject);
    procedure btnTPLClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure btnHP1Click(Sender: TObject);
    procedure btnHP2Click(Sender: TObject);
    procedure btnHP3Click(Sender: TObject);
    procedure btnHP4Click(Sender: TObject);
    procedure btnHP5Click(Sender: TObject);
    procedure btnHP6Click(Sender: TObject);
    procedure btnHP7Click(Sender: TObject);
    procedure btnHP8Click(Sender: TObject);
    procedure btnHPRClick(Sender: TObject);
    procedure btnHPLClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

มีต่อ>>

```
var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.FormActivate(Sender: TObject);
begin
  if (StartUpIOPorts($378) = false) then
  begin
    ShowMessage('ไม่สามารถเปิดใช้งานพอร์ต LPT ได้');
    Close;
  end;
  WritePCtrl($378, $80);
end;

procedure TForm1.btnStopClick(Sender: TObject);
begin
  WritePA($378, $00);
end;

procedure TForm1.btnSP1Click(Sender: TObject);
begin
  WritePA($378, $01);
end;

procedure TForm1.btnSP2Click(Sender: TObject);
begin
  WritePA($378, $02);
end;

procedure TForm1.btnSP3Click(Sender: TObject);
begin
  WritePA($378, $04);
end;

procedure TForm1.btnSP4Click(Sender: TObject);
begin
  WritePA($378, $08);
end;

procedure TForm1.btnSPRClick(Sender: TObject);
begin
  btnSP1Click(Sender); Sleep(100);
  btnSP2Click(Sender); Sleep(100);
  btnSP3Click(Sender); Sleep(100);
  btnSP4Click(Sender); Sleep(100);
end;

procedure TForm1.btnSPLClick(Sender: TObject);
begin
  btnSP4Click(Sender); Sleep(100);
```

มีต่อ>>

```
  btnSP3Click(Sender); Sleep(100);
  btnSP2Click(Sender); Sleep(100);
  btnSP1Click(Sender); Sleep(100);
end;

procedure TForm1.btnTP1Click(Sender: TObject);
begin
  WritePA($378, $03);
end;

procedure TForm1.btnTP2Click(Sender: TObject);
begin
  WritePA($378, $06);
end;

procedure TForm1.btnTP3Click(Sender: TObject);
begin
  WritePA($378, $0C);
end;

procedure TForm1.btnTP4Click(Sender: TObject);
begin
  WritePA($378, $09);
end;

procedure TForm1.btnTPRClick(Sender: TObject);
begin
  btnTP1Click(Sender); Sleep(100);
  btnTP2Click(Sender); Sleep(100);
  btnTP3Click(Sender); Sleep(100);
  btnTP4Click(Sender); Sleep(100);
end;

procedure TForm1.btnTPLClick(Sender: TObject);
begin
  btnTP4Click(Sender); Sleep(100);
  btnTP3Click(Sender); Sleep(100);
  btnTP2Click(Sender); Sleep(100);
  btnTP1Click(Sender); Sleep(100);
end;

procedure TForm1.btnExitClick(Sender: TObject);
begin
  WritePA($378, $00);
  Application.Terminate;
end;

procedure TForm1.btnHP1Click(Sender: TObject);
begin
  WritePA($378, $01);
end;
```

มีต่อ>>

```
procedure TForm1.btnHP2Click(Sender: TObject);
begin
  WritePA($378, $03);
end;

procedure TForm1.btnHP3Click(Sender: TObject);
begin
  WritePA($378, $02);
end;

procedure TForm1.btnHP4Click(Sender: TObject);
begin
  WritePA($378, $06);
end;

procedure TForm1.btnHP5Click(Sender: TObject);
begin
  WritePA($378, $04);
end;

procedure TForm1.btnHP6Click(Sender: TObject);
begin
  WritePA($378, $0C);
end;

procedure TForm1.btnHP7Click(Sender: TObject);
begin
  WritePA($378, $08);
end;

procedure TForm1.btnHP8Click(Sender: TObject);
begin
  WritePA($378, $09);
end;

procedure TForm1.btnHPRClick(Sender: TObject);
begin
  btnHP1Click(Sender); Sleep(100);
  btnHP2Click(Sender); Sleep(100);
  btnHP3Click(Sender); Sleep(100);
  btnHP4Click(Sender); Sleep(100);
  btnHP5Click(Sender); Sleep(100);
  btnHP6Click(Sender); Sleep(100);
  btnHP7Click(Sender); Sleep(100);
  btnHP8Click(Sender); Sleep(100);
end;

procedure TForm1.btnHPLClick(Sender: TObject);
begin
  btnHP8Click(Sender); Sleep(100);
  btnHP7Click(Sender); Sleep(100);
  btnHP6Click(Sender); Sleep(100);
```

มีต่อ>>

```
  btnHP5Click(Sender); Sleep(100);
  btnHP4Click(Sender); Sleep(100);
  btnHP3Click(Sender); Sleep(100);
  btnHP2Click(Sender); Sleep(100);
  btnHP1Click(Sender); Sleep(100);
end;

end.
```

6. สรุป

และแล้วตัวอย่างของบทความเรื่องพอร์ตขนานและการเชื่อมต่อก็จบลงจนได้ครับ ในครั้งหน้าถ้ามีโอกาสผมจะลองเขียนตามที่ได้เกริ่นเอาไว้ให้ผู้อ่านได้อ่านกัน (คงจะเป็นเดือนหน้านะครับ) ส่วนบทความในครั้งนี้หวังว่าคงเป็นตัวอย่างที่เพียงพอสำหรับคนที่สนใจเขียนโปรแกรมเพื่อเชื่อมต่อพอร์ตขนานด้วยโปรแกรม Delphi แต่ถ้าสนใจ Visual BASIC ก็สามารถหาอ่านได้มากแหล่งต่างๆกันอยู่แล้ว

สุดท้ายต้องขอขอบคุณครอบครัวและทางทีมงานของอีทีทีอีกเช่นเคยครับ เพราะถ้าไม่ผู้สนับสนุนเหล่านี้ผมคงหมดกำลังใจเขียนบทความไปแล้ว โดยเฉพาะคุณกอบกิจ เติมผาติ ที่คอยมอบโอกาสให้ผมเสมอมา ผมขอขอบคุณเอาไว้ ณ ที่นี่ครับ บทความเรื่องต่อไปที่ผมอยากทำที่สุดคงเป็นการสานต่อชุดคำสั่ง 68HC11 ให้จบครับ เนื่องจากทิ้งไปหลายปีแล้ว (ที่กังวลคือเวลาผมแตะ 68HC11 ทีไรมีอันเป็นไปทุกที) ส่วน AVR นั้นบอร์ดที่ผมซื้อมาเมื่อ 5 ปีก่อนมันหมดอายุไปแล้ว (เวลาผ่านเร็วจริงๆ) เนื่องจากห้องทำงานของผม (ที่บ้าน) อยู่ในบริเวณที่ร้อนที่สุดของบ้าน ประกอบกับเก็บเอาไว้นาน ไอซีมันเลยป่วนไปแล้ว เอาเป็นว่าถ้าสบโอกาสเรื่องอะไรผมจะเขียนเรื่องนั้นล่ะกันครับ บทความตอนนี้คงจบลงแค่นี้ครับ