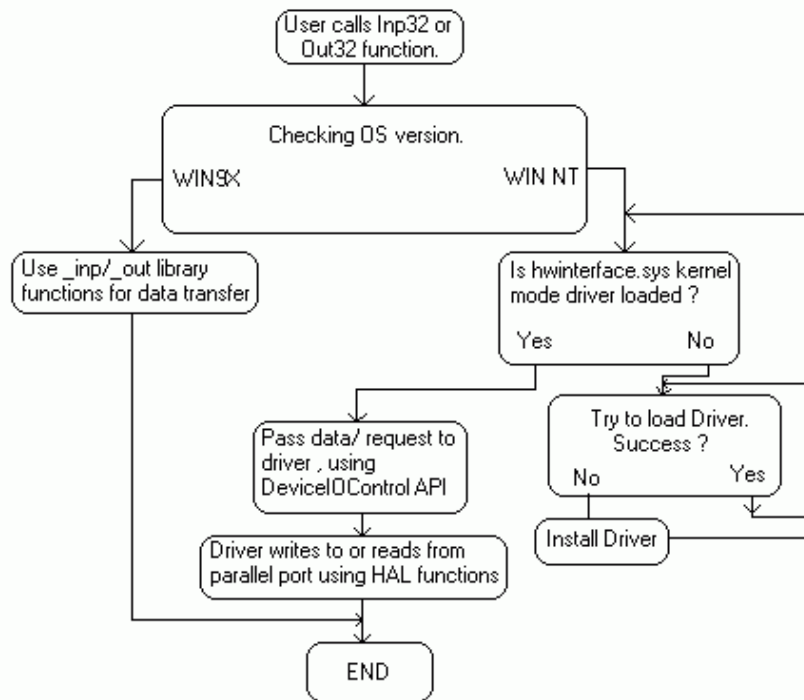


เนื่องจากการเขียนโปรแกรมผ่าน Library Driver ของ PCI Card นั้นถึงแม้ว่าจะมีข้อดี คือ โปรแกรมที่สร้างขึ้นจะมีความอ่อนตัวในการใช้งานสามารถนำไปใช้งานกับเครื่องคอมพิวเตอร์ใดๆก็ได้โดยไม่มีข้อจำกัด แต่ก็มีความยุ่งยากซับซ้อนอยู่พอสมควร แต่อย่างไรก็ตามเราสามารถลดความยุ่งยากของการสร้างโปรแกรม Application เพื่อใช้งานร่วมกับการ์ด ET-PCI8255 V3 ได้ โดยการจัดหา Driver สำเร็จรูปที่สร้างไว้ในรูปแบบของ Library แบบ DLL (Dynamic Link Library) ซึ่งทำหน้าที่เกี่ยวกับการติดต่อกับ I/O Port มาใช้งานแทน ซึ่งวิธีการนี้จะมีข้อดี คือ ง่ายและรวดเร็วสามารถเขียนโค้ดโปรแกรมเพียงสั้นๆก็สามารถติดต่อกับ I/O Port ของการ์ดได้แล้ว

แต่วิธีการนี้จะมีข้อจำกัดอย่างหนึ่ง คือ จะต้องทำการติดตั้งการ์ด ET-PCI8255 V3 ให้เรียบร้อยเสียก่อน จากนั้นก็เข้าไปทำการตรวจสอบตำแหน่งแอดเดรสการทำงานของการ์ดที่ได้รับการจัดสรรจาก BIOS ว่ามีค่าเป็นเท่าใด เพื่อจะใช้นำค่าตำแหน่งแอดเดรสเริ่มต้นของ "Input / Output Range" ที่ตรวจสอบได้มาใช้ในการกำหนดเป็นค่า Base Address สำหรับใช้เป็นจุดอ้างอิงตำแหน่งแอดเดรสให้กับ "I/O Function Register" และ "Port Address" ของ 8255 อีกครั้งหนึ่ง แต่โปรแกรม Application ที่เขียนขึ้นด้วยวิธีการนี้อาจไม่สามารถเคลื่อนย้ายไปทำงานยังคอมพิวเตอร์เครื่องอื่นๆได้ เนื่องจากตำแหน่งแอดเดรสของการ์ดในเครื่องคอมพิวเตอร์แต่ละเครื่องนั้นอาจได้รับการจัดสรรตำแหน่งแอดเดรสไม่ตรงกัน แต่ก็อาจแก้ไขได้โดยอาจออกแบบโปรแกรมให้สามารถกำหนดค่า Base Address จากผู้ใช้แล้วบันทึกเป็น INI File ไว้ โดยทุกๆครั้งที่สั่ง Run โปรแกรมก็สั่งให้โปรแกรมอ่านค่าตำแหน่ง Base Address จาก INI File ขึ้นมาเพื่อใช้เป็นจุดอ้างอิงตำแหน่งการทำงานของการ์ดแทนดังนี้

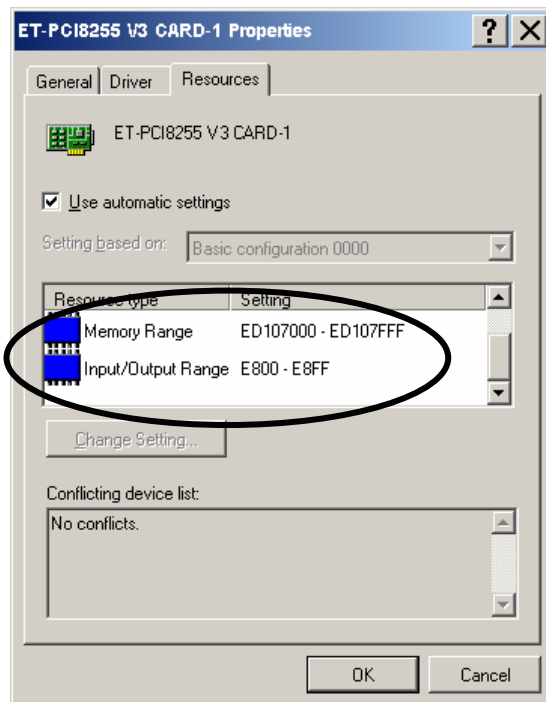
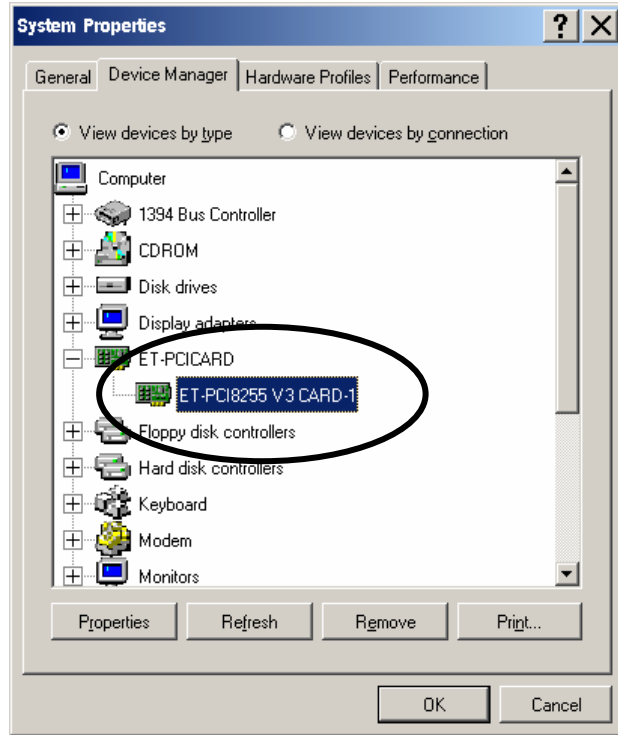


รูปแสดง หลักการทำงานของ Driver สำหรับติดต่อกับ I/O Port (InpOut32.DLL)

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

การตรวจสอบตำแหน่งการทำงานของการ์ด

โดยการ์ด ET-PCI8255 V3 นั้นหลังจากทำการติดตั้งการ์ดเข้ากับ PCI Slot เป็นที่เรียบร้อยแล้ว ถ้าไม่เกิดข้อผิดพลาดใดๆขึ้น ตัวการ์ดก็จะได้รับการจัดสรรตำแหน่ง Address ต่างๆ โดย BIOS ซึ่งเราสามารถเข้าไปตรวจสอบตำแหน่งการทำงานของการ์ดได้จาก Control Panel → System → Device Manager → ET-PCICARD → ET-PCI8255 V3 CARD-n (n=1..4) ดังรูป



ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

การใช้งาน InpOut32.DLL

InpOut32.DLL เป็น Library สำหรับใช้ติดต่อกับ I/O Port ภายใต้ Driver Mode สามารถใช้งานได้กับระบบปฏิบัติการ Windows ได้ทุกรุ่น โดย InpOut32.DLL เป็น Library แบบ DLL ซึ่งได้รับการพัฒนาขึ้นเพื่อใช้ติดต่อกับ I/O Port แบบ 32 Bit ได้ทั้งหมด และที่สำคัญคือ เป็น Library ซึ่งแจกฟรี โดยปัจจุบัน (มกราคม 2548) ผู้สนใจสามารถ Download มาใช้ได้ฟรีจาก <http://www.logix4u.net> โดยในการติดตั้งใช้งานนั้นให้ทำการ Copy ไฟล์ InpOut32.DLL ไปไว้ยัง Directory ของ Windows ดังนี้

- ถ้าเป็น Windows 95/98 ให้ Copy ไปไว้ที่ <Windows>\System
- ถ้าเป็น Windows Me/NT/2000/XP ให้ Copy ไปไว้ที่ <Windows>\System32

หมายเหตุ <Windows> หมายถึง Folder หรือ Directory ของโปรแกรม Windows เช่นถ้าติดตั้งไว้ที่ฮาร์ดดิสก์ Drive C โดยอยู่ใน Sub Directory ชื่อ Windows ก็จะหมายถึง C:\Windows แต่ถ้าติดตั้ง Windows ไว้ภายใต้ชื่อ Directory อื่นๆ เช่น WINXP ก็ต้องเปลี่ยนตามความเป็นจริงให้ถูกต้องด้วย

โดยเราจะใช้ฟังก์ชันที่บรรจุไว้ใน InpOut32.DLL สำหรับติดต่อกับ I/O Port ของการ์ด ET-PCI8255 V3 จำนวน 2 ฟังก์ชัน เท่านั้น คือ Inp32 สำหรับการอ่านค่า Input จากการ์ด และ Out32 สำหรับเขียนค่าไปยังการ์ด โดยรูปแบบการใช้งานฟังก์ชันทั้ง 2 เป็นดังนี้

ตัวแปรขนาด 8 BIT = INP32(ตำแหน่งพอร์ตแอดเดรส) ‘การอ่านค่า Input จากพอร์ต

OUT32(ตำแหน่งพอร์ตแอดเดรส ,ข้อมูลขนาด 8 BIT) ‘การเขียนค่าออกไปยังพอร์ต

โดยการอ่านค่าข้อมูลการพอร์ตจะสร้างเป็นฟังก์ชัน ซึ่งจะต้องสร้างตัวแปรขนาด 8 บิตขึ้นมาเพื่อใช้รับค่าข้อมูลที่อ่านได้จากพอร์ต โดยตำแหน่งของพอร์ตแอดเดรสจะมีขนาดเป็น 16 บิต ส่วนในการเขียนค่าข้อมูลออกไปยังพอร์ตนั้นนิยมสร้างเป็นโปรแกรมย่อยเนื่องจากไม่ต้องการการส่งค่ากลับมายังโปรแกรมหลักอีก ซึ่งจะต้องทำการกำหนดค่าของตำแหน่งพอร์ตแอดเดรสขนาด 16 บิต พร้อมกับข้อมูลขนาด 8 บิต ให้กับคำสั่งด้วย

ซึ่งในการเขียนโปรแกรมเพื่อติดต่อกับฟังก์ชันใน InpOut32.DLL นั้น ในครั้งแรกจะต้องมีการประกาศเรียกใช้ไฟล์ “InpOut32.DLL” ให้กับโปรแกรมได้รับทราบเสียก่อน จากนั้นจึงจะสร้างฟังก์ชัน และ โปรแกรมย่อยมารองรับการใช้งาน DLL อีกครั้งหนึ่ง โดยรูปแบบคำสั่งในการเรียกใช้ไฟล์ DLL และการสร้างโปรแกรมย่อยหรือฟังก์ชันนั้น จะต้องอ้างอิงกับข้อกำหนดของตัวภาษาที่จะใช้ในการสร้างโปรแกรม Application ด้วย

การเรียกใช้งาน InpOut32.DLL ด้วยโปรแกรม Delphi

ในการเรียกใช้งาน InpOut32.DLL ด้วยโปรแกรม Delphi นั้น สามารถทำได้หลายรูปแบบ ซึ่งในที่นี้จะขอแสดงตัวอย่างของการประกาศการเรียกใช้ไฟล์ DLL ด้วยวิธีการแบบ Implicit Loading ไว้ในส่วนเริ่มต้นของโปรแกรมเสียก่อน จากนั้นจึงจะสามารถเรียกใช้ฟังก์ชันดังกล่าวได้จากส่วนต่างๆของโปรแกรม โดยสามารถแสดงตัวอย่างการประกาศและการเรียกใช้งาน DLL ได้ดังต่อไปนี้

```
Unit Unit1;

Interface

Uses Windows,  ;

Type TForm1 = class(TForm)
    Procedure SETUP_8255Click(Sender: TObject);
    Procedure PA01Click(Sender: TObject);

Private
    { Private declarations }

Public
    { Public declarations }

End;

Var Form1 : TForm1;

    Procedure Out32(Port:Word;Data:Byte);Stdcall;External'InpOut32.DLL';
    Function Inp32(Port:Word):Byte; Stdcall; External 'InpOut32.DLL';

Implementation
.
Procedure TForm1.SETUP_8255Click(Sender: TObject);
Begin
    Out32($E8CC,$80);           // Write Control Port 8255#1 = Output
End;

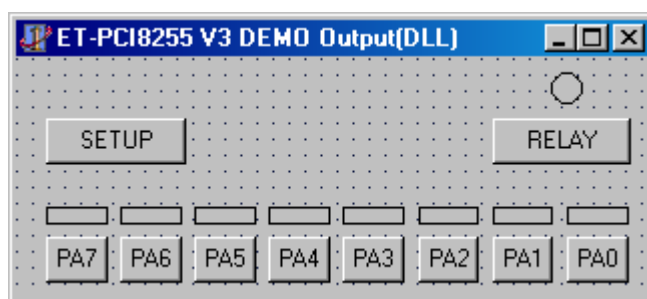
Procedure TForm1.PA01Click(Sender: TObject);
Var InOutData : Byte;
Begin
    InOutData := Inp32($E8C0);   // Read Input Port-A
End;
```

ตัวอย่างการเรียกใช้งาน InpOut32.DLL ด้วยโปรแกรม Delphi 5

ตัวอย่างที่ 1.1 DEMO Output (เรียกใช้ Library ไฟล์จาก InpOut32.DLL)

สำหรับโปรแกรมตัวอย่างนี้ จะแสดงให้เห็นการประยุกต์ใช้งาน 8255 ให้ทำหน้าที่เป็น Output สำหรับใช้ควบคุมการทำงานของอุปกรณ์ไฟฟ้าต่างๆในลักษณะของการ ON/OFF โดยในตัวอย่างนี้จะแสดงให้เห็นเฉพาะการเขียนโปรแกรมควบคุม 8255 ตัวที่ 1 โดยแสดงการควบคุม Output ของพอร์ต PA เพียงพอร์ตเดียวเท่านั้น เพื่อให้ Code ของโปรแกรมมีขนาดเล็ก และง่ายต่อการทำความเข้าใจ โดยเมื่อผู้อ่านสามารถทำความเข้าใจกับการทำงานของโปรแกรมในส่วนนี้ได้แล้ว ก็จะสามารถนำตัวอย่างไปดัดแปลงแก้ไขเพิ่มเติม เพื่อใช้งานกับพอร์ตอื่นๆที่เหลืออยู่ได้โดยไม่ยาก เนื่องจากหลักการทำงานของโปรแกรมจะใช้วิธีการที่เหมือนกันทั้งหมด แตกต่างกันเฉพาะในส่วนของตำแหน่งแอดเดรสที่อ้างถึงพอร์ตของ 8255 แต่ละพอร์ต เท่านั้นเอง สำหรับขั้นตอนการเขียนโปรแกรมเพื่อส่งข้อมูลออกพอร์ต 8255 ของการ์ด ET-PCI8255 V3 มีรายละเอียดดังนี้

1. เปิด Project ใหม่โดยไปที่ File > New Application จะได้ฟอร์มเปล่าขึ้นมา 1 ฟอร์ม
2. ให้นำคอนโทรลชนิดต่างๆ มาจัดวางใน Form ดังรูป



รูปแสดง การจัดวาง Component บน Form ของโปรแกรม ตัวอย่างที่ 1.1

โดยในตัวอย่างนี้จะใช้ Component ทั้งหมด 2 ชนิดด้วยกัน คือ Command Button และ Shape ซึ่ง Command Button จะมีทั้งหมด 10 ชุด ใช้สำหรับ Setup การทำงานของการ์ด ET-PCI8255 V3 จำนวน 1 ชุด โดยเมื่อกดปุ่ม SETUP จะเป็นการสั่ง Initial การทำงานของการ์ด ET-PCI8255 V3 โดยจะเริ่มต้นด้วยการเปิดการทำงานของสัญญาณ Chip Select และปิดการทำงานของ Relay ไว้ พร้อมกับ Initial การทำงานของ 8255 ตัวที่ 1 ให้ทำหน้าที่เป็น Output Port ทั้ง 3 พอร์ต (PA,PB และ PC ทำหน้าที่เป็น Output) ส่วน Command Button ที่เหลือจะใช้สำหรับสั่ง ON และ OFF การทำงานของ Relay และ Output ของ PA0..PA7 ตามลำดับ สำหรับ Shape นั้นจะใช้สำหรับทำหน้าที่เป็นเสมือน หลอด LED หรือไฟแสดงสถานะการทำงานของ Relay และ Output ของ PA0..PA7 โดยถ้า Output มีค่าเป็น Logic "0" จะให้สถานะเป็นสีแดง แต่ถ้า Output มีค่าเป็น Logic "1" จะให้สถานะเป็นสีฟ้าของ Form แทน

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

3. ทำการกำหนดค่า Properties ให้กับคอนโทรลต่างๆ ดังนี้

ชื่อ	คอนโทรล	พรีอเพอร์ตี้	ค่าที่กำหนด
Form1	Form	Name	Form1
		Border Style	bsSizeable
		Caption	ET-PCI8255 V3 DEMO Output(DLL)
SETUP	Command Button	Name	SETUP
		Enable	True
		Caption	SETUP
PA0 . . PA7	Command Button	Name	PA01,PA11,PA21,...PA71
		Enable	False
		Caption	PA0,PA1,PA2,...PA7
	Shape	Name	LEDPA01,LEDPA11,LEDPA21,...LEDPA71
		Fill Style	Solid
		Shape	Rectangle
RELAY	Command Button	Name	RELAY
		Enable	False
		Caption	RELAY
	Shape	Name	LEDRelay
		Fill Style	Solid
		Shape	Circle

สำหรับการกำหนดค่า พรีอเพอร์ตี้ ของคอนโทรลต่างๆนั้น จะแสดงเฉพาะค่าที่เปลี่ยนแปลงไปจากค่า Default เดิมเท่านั้น โดยในส่วนของ Command Button นั้นจะเปิดการทำงาน (Enabled) ไว้เฉพาะที่เป็นของ Setup เพียงตัวเดียวเท่านั้น ส่วน Command Button ตัวอื่นๆจะต้องปิด(Disable) ไว้ก่อน เพื่อบังคับให้ผู้ใช้จำเป็นต้องทำการ Setup การทำงานให้กับการ์ดเสียก่อน จากนั้นจึงจะสามารถสั่งงาน Output ต่างๆได้ โดยการทำงานของ Command Button ของ Relay และ PA0..PA7 นั้นจะถูกเปิดการทำงานเมื่อทำการสั่ง Setup แล้ว

โดย Command Button ของพอร์ต PA นั้นจะมีทั้งหมด 8 ชุด คือ PA0 ถึง PA7 โดยกำหนดให้ตำแหน่งของ Command Button สำหรับควบคุมการทำงานของ Output ทั้ง 8 มีชื่อเป็น PA01 ถึง PA71 ตามลำดับ ซึ่ง PA01

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

หมายถึงพอร์ต PA0 ของ 8255 ตัวที่1 ส่วน PA11 ก็จะหมายถึงพอร์ต PA1 ของ 8255 ตัวที่1 และ PA71 ก็จะหมายถึงพอร์ต PA7 ของ 8255 ตัวที่1 เป็นต้น

4. เขียนโค้ดคำสั่งเพื่อควบคุมการทำงานของโปรแกรมดังต่อไปนี้

```
{
Demo1 Output Demo Access Input / Output Library "INPOUT32.DLL"
}

Unit Unit1;
Interface
Uses   Windows,Messages,SysUtils,Classes,Graphics,Controls,Forms,Dialogs,StdCtrls,ExtCtrls;

Type   TForm1 = class(TForm)
        SETUP_8255: TButton;
        RELAY: TButton;
        PA01: TButton;
        PA11: TButton;
        PA21: TButton;
        PA31: TButton;
        PA41: TButton;
        PA51: TButton;
        PA61: TButton;
        PA71: TButton;
        LEDRELAY: TShape;
        LEDPA01: TShape;
        LEDPA11: TShape;
        LEDPA21: TShape;
        LEDPA31: TShape;
        LEDPA41: TShape;
        LEDPA51: TShape;
        LEDPA61: TShape;
        LEDPA71: TShape;
        Procedure SETUP_8255Click(Sender: TObject);
        Procedure RELAYClick(Sender: TObject);
end;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure PA01Click(Sender: TObject);
Procedure PA11Click(Sender: TObject);
Procedure PA21Click(Sender: TObject);
Procedure PA31Click(Sender: TObject);
Procedure PA41Click(Sender: TObject);
Procedure PA51Click(Sender: TObject);
Procedure PA61Click(Sender: TObject);
Procedure PA71Click(Sender: TObject);

Private
  { Private declarations }
Public
  { Public declarations }
End;

Var
  Form1 : TForm1;
  BaseAddress : Word;
  Procedure Out32(Port:Word; Data:Byte); Stdcall;External'InpOut32.DLL';
  Function Inp32(Port:Word):Byte; Stdcall; External 'InpOut32.DLL';

Implementation

/** I/O Address of ET-PCI8255 V3 **/
Const IO_BaseAddress = $E800; // I/O Base Address

// Tiger-320 Register Offset
Const PIB = $00; // Reset & PIB Cycle
Const AUXC = $02; // Aux Direction Port
Const AUXD = $03; // Aux Data Port

Const PA1 = $C0; // Port-A 8255#1
Const PB1 = $C4; // Port-B 8255#1
Const PC1 = $C8; // Port-C 8255#1
Const PCC1 = $CC; // Port Control 8255#1
Const PA2 = $D0; // Port-A 8255#2
Const PB2 = $D4; // Port-B 8255#2
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Const  PC2 = $D8;           // Port-C 8255#2
Const  PCC2 = $DC;         // Port Control 8255#2
Const  PA3 = $E0;         // Port-A 8255#3
Const  PB3 = $E4;         // Port-B 8255#3
Const  PC3 = $E8;         // Port-C 8255#3
Const  PCC3 = $EC;        // Port Control 8255#3

Const  ON_Bit0 = $01;      // XXXX XXXX OR 0000 0001 = XXXX XXX1
Const  OFF_Bit0 = $FE;     // XXXX XXXX AND 1111 1110 = XXXX XXX0
Const  ON_Bit1 = $02;      // XXXX XXXX OR 0000 0010 = XXXX XX1X
Const  OFF_Bit1 = $FD;     // XXXX XXXX AND 1111 1101 = XXXX XX0X
Const  ON_Bit2 = $04;      // XXXX XXXX OR 0000 0100 = XXXX X1XX
Const  OFF_Bit2 = $FB;     // XXXX XXXX AND 1111 1011 = XXXX X0XX
Const  ON_Bit3 = $08;      // XXXX XXXX OR 0000 1000 = XXXX 1XXX
Const  OFF_Bit3 = $F7;     // XXXX XXXX AND 1111 0111 = XXXX 0XXX
Const  ON_Bit4 = $10;      // XXXX XXXX OR 0001 0000 = XXX1 XXXX
Const  OFF_Bit4 = $EF;     // XXXX XXXX AND 1110 1111 = XXX0 XXXX
Const  ON_Bit5 = $20;      // XXXX XXXX OR 0010 0000 = XX1X XXXX
Const  OFF_Bit5 = $DF;     // XXXX XXXX AND 1101 1111 = XX0X XXXX
Const  ON_Bit6 = $40;      // XXXX XXXX OR 0100 0000 = X1XX XXXX
Const  OFF_Bit6 = $BF;     // XXXX XXXX AND 1011 1111 = X0XX XXXX
Const  ON_Bit7 = $80;      // XXXX XXXX OR 1000 0000 = 1XXX XXXX
Const  OFF_Bit7 = $7F;     // XXXX XXXX AND 0111 1111 = 0XXX XXXX

Const  OutputOFF = CIMenu; // Color of Output OFF ("0") Status
Const  OutputON = CIRed;   // Color of Output ON ("1") Status

{$R *.DFM}

Procedure TForm1.SETUP_8255Click(Sender: TObject);
Var  SetupData : Byte;

Begin
  BaseAddress := IO_BaseAddress;           // Set I/O Base Address

  // Initial Reset and 8255 Bus Cycle //
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
SetupData := Inp32(BaseAddress+PIB);           // Read PIB Reset Port
SetupData := SetupData AND OFF_BIT0;          // Bit0 = EXTRST# = "0" (Reset:RES#)
SetupData := SetupData OR ON_BIT5;           // Bit5:4 = 11 = PIB Cycle Slowest
SetupData := SetupData OR ON_BIT4;
Out32(BaseAddress+PIB,SetupData);             // Active RES# & Relay

// Initial Data (AUX) For CS# and Relay //
SetupData := Inp32(BaseAddress+AUXD);         // Read Aux Data Port
SetupData := SetupData AND OFF_BIT0;         // Bit0 = Aux0 = "0" (Enable CS)
SetupData := SetupData OR ON_BIT4;           // Bit4 = Aux4 = "1" (Relay OFF)
Out32(BaseAddress+AUXD,SetupData);           // Active Chips Select & Relay
LEDRelay.Brush.Color := OutputOFF;          // LED Relay = OFF

// Initial Direction (AUX) For CS# and Relay //
SetupData := Inp32(BaseAddress+AUXC);         // Read Aux Port Direction
SetupData := SetupData OR ON_BIT4;           // Aux4 = "1" = Output
SetupData := SetupData OR ON_BIT0;           // Aux0 = "1" = Output
Out32(BaseAddress+AUXC,SetupData);           // Setup Aux Direction

// Initial 8255#1 = All Output Port //
Setup_8255.Enabled := False;                 // Disable Setup After Setup
Out32(BaseAddress+PCC1,$80);                 // Write Control Port 8255#1
RELAY.Enabled := True;                       // Enable Relay Control
PA01.Enabled := True;                        // Enable Output Control
PA11.Enabled := True;
PA21.Enabled := True;
PA31.Enabled := True;
PA41.Enabled := True;
PA51.Enabled := True;
PA61.Enabled := True;
PA71.Enabled := True;
End;

Procedure TForm1.RelayClick(Sender: TObject);
Var InOutData : Byte;
    CheckData : Byte;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Begin
  InOutData := Inp32(BaseAddress+AuxD);           // Read Output Latch AUX
  CheckData := InOutData AND ON_BIT4;           // Check Last AUX4 Status
  If CheckData = $10 Then
  Begin
    InOutData := InOutData AND OFF_BIT4;        // AUX4 = "0" (1110 1111)
    LEDRelay.Brush.Color := OutputON;          // LED RElay = ON
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT4;          // AUX4 = "1" (0001 0000)
    LEDRelay.Brush.Color := OutputOFF;         // LED Relay = OFF
  End;
  Out32(BaseAddress+AuxD,InOutData);           // Update AuxD
End;

Procedure TForm1.PA01Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);          // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT0;          // Check Last PA0 Status
  If CheckData = $01 Then
  Begin
    InOutData := InOutData AND OFF_BIT0;        // PA0 = "0" (1111 1110)
    LEDPA01.Brush.Color := OutputOFF;          // LED PA0 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT0;          // PA0 = "1" (0000 0001)
    LEDPA01.Brush.Color := OutputON;           // LED PA0 of 8255#1 = ON
  End;
  Out32(BaseAddress+PA1,InOutData);           // Update Port-A
End;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure TForm1.PA11Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT1;           // Check Last PA1 Status
  If CheckData = $02 Then
  Begin
    InOutData := InOutData AND OFF_BIT1;        // PA1 = "0" (1111 1101)
    LEDPA11.Brush.Color := OutputOFF;          // LED PA1 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT1;          // PA1 = "1" (0000 0010)
    LEDPA11.Brush.Color := OutputON;           // LED PA1 of 8255#1 = ON
  End;
  Out32(BaseAddress+PA1,InOutData);             // Update Port-A
End;

Procedure TForm1.PA21Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT2;           // Check Last PA2 Status
  If CheckData = $04 Then
  Begin
    InOutData := InOutData AND OFF_BIT2;        // PA2 = "0" (1111 1011)
    LEDPA21.Brush.Color := OutputOFF;          // LED PA2 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT2;          // PA2 = "1" (0000 0100)
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
    LEDPA21.Brush.Color := OutputON;           // LED PA2 of 8255#1 = ON
End;
    Out32(BaseAddress+PA1,InOutData);         // Update Port-A
End;

Procedure TForm1.PA31Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+PA1);      // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT3;      // Check Last PA3 Status
    If CheckData = $08 Then
    Begin
        InOutData := InOutData AND OFF_BIT3; // PA3 = "0" (1111 0111)
        LEDPA31.Brush.Color := OutputOFF;    // LED PA3 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT3;   // PA3 = "1" (0000 1000)
        LEDPA31.Brush.Color := OutputON;     // LED PA3 of 8255#1 = ON
    End;
    Out32(BaseAddress+PA1,InOutData);        // Update Port-A
End;

Procedure TForm1.PA41Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+PA1);      // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT4;      // Check Last PA4 Status
    If CheckData = $10 Then
    Begin
        InOutData := InOutData AND OFF_BIT4; // PA4 = "0" (1110 1111)
        LEDPA41.Brush.Color := OutputOFF;    // LED PA4 of 8255#1 = OFF
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
End
Else
Begin
    InOutData := InOutData OR ON_BIT4;           // PA4 = "1" (0001 0000)
    LEDPA41.Brush.Color := OutputON;           // LED PA4 of 8255#1 = ON
End;
Out32(BaseAddress+PA1,InOutData);           // Update Port-A
End;

Procedure TForm1.PA51Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+PA1);       // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT5;       // Check Last PA5 Status
    If CheckData = $20 Then
    Begin
        InOutData := InOutData AND OFF_BIT5;  // PA5 = "0" (1101 1111)
        LEDPA51.Brush.Color := OutputOFF;     // LED PA5 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT5;    // PA5 = "1" (0010 0000)
        LEDPA51.Brush.Color := OutputON;     // LED PA5 of 8255#1 = ON
    End;
    Out32(BaseAddress+PA1,InOutData);         // Update Port-A
End;

Procedure TForm1.PA61Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+PA1);       // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT6;       // Check Last PA6 Status
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
If CheckData = $40 Then
Begin
  InOutData := InOutData AND OFF_BIT6;           // PA6 = "0" (1011 1111)
  LEDPA61.Brush.Color := OutputOFF;             // LED PA6 of 8255#1 = OFF
End
Else
Begin
  InOutData := InOutData OR ON_BIT6;            // PA6 = "1" (0100 0000)
  LEDPA61.Brush.Color := OutputON;             // LED PA6 of 8255#1 = ON
End;
Out32(BaseAddress+PA1,InOutData);              // Update Port-A
End;

Procedure TForm1.PA71Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);          // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT7;          // Check Last PA7 Status
  If CheckData = $80 Then
  Begin
    InOutData := InOutData AND OFF_BIT7;       // PA7 = "0" (0111 1111)
    LEDPA71.Brush.Color := OutputOFF;         // LED PA7 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT7;         // PA7 = "1" (1000 0000)
    LEDPA71.Brush.Color := OutputON;          // LED PA7 of 8255#1 = ON
  End;
  Out32(BaseAddress+PA1,InOutData);            // Update Port-A
End;
End.
```

แสดง Code โปรแกรมตัวอย่างที่ 1.1

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

อธิบายการทำงานของคำสั่งในโปรแกรม ตัวอย่างที่ 1.1

เพื่อให้ง่ายต่อการทำความเข้าใจโปรแกรม ผู้เขียนจะขอแยกอธิบายหลักการทำงานของโปรแกรม เฉพาะบางส่วนที่น่าสนใจเพื่อที่ผู้อ่านจะได้ใช้อ้างอิงประกอบความเข้าใจ อันจะเป็นแนวทางในการนำตัวอย่างโปรแกรมไปประยุกต์ หรือดัดแปลงใช้งานในอนาคตต่อไปได้ โดยจะขอแยกอธิบายการทำงานของโปรแกรมเป็นส่วนตัวดังนี้

```
Procedure Out32(Port:Word; Data:Byte); Stdcall; External 'InpOut32.DLL';  
Function Inp32(Port:Word):Byte; Stdcall; External 'InpOut32.DLL';
```

โดยในส่วนนี้จะเป็นการประกาศเรียกใช้ฟังก์ชัน Inp32 และ Out32 จากไฟล์ Library ชื่อ InpOut32.dll เพื่อใช้ในการควบคุมการ In/Out Port สำหรับติดต่อกับการ์ด

```
Const IO_BaseAddress = $E800; // I/O Base Address  
Const PIB = $00; // Reset & PIB Cycle  
Const AUXC = $02; // Aux Direction Port  
Const AUXD = $03; // Aux Data Port  
Const PA1 = $C0; // Port-A 8255#1  
Const PB1 = $C4; // Port-B 8255#1  
Const PC1 = $C8; // Port-C 8255#1  
Const PCC1 = $CC; // Port Control 8255#1  
Const PA2 = $D0; // Port-A 8255#2  
Const PB2 = $D4; // Port-B 8255#2  
Const PC2 = $D8; // Port-C 8255#2  
Const PCC2 = $DC; // Port Control 8255#2  
Const PA3 = $E0; // Port-A 8255#3  
Const PB3 = $E4; // Port-B 8255#3  
Const PC3 = $E8; // Port-C 8255#3  
Const PCC3 = $EC; // Port Control 8255#3
```

ในส่วนนี้จะเป็นการประกาศตัวแปรต่างๆ ในการเขียนโปรแกรม สิ่งที่สำคัญมากในการประกาศตัวแปร คือ การกำหนด Base Address ซึ่งตัวอย่างใช้ค่า &HE800 ซึ่งตรงจุดนี้ ในคอมพิวเตอร์แต่ละเครื่องที่ทำการติดตั้งการ์ด ET-PCI8255 V3 ไปแล้ว อาจมีความแตกต่างกัน ซึ่งผู้ใช้งานจะต้องทำการตรวจสอบค่าของตำแหน่ง Base Address ของการ์ด ET-PCI8255 V3 ในเครื่องคอมพิวเตอร์เองจาก Control Panel แล้วแก้ไขค่าให้ตรงกันเสียก่อน หลังจากนั้นก็เป็นส่วนของการประกาศค่าคงที่สำหรับใช้อ้างอิงตำแหน่งของ รีจิสเตอร์ใช้งานของชิพ TIGER320 และพอร์ต

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

8255 ทั้งหมด โดยค่าที่ประกาศไว้นี้เป็นค่า Offset ของตำแหน่งที่จะใช้สำหรับอ้างอิง โดยในการใช้งานจะต้องนำไปบวกเข้ากับค่าของ Base Address ของการ์ด ET-PCI8255 V3 อีกครั้งหนึ่ง ซึ่งการทำเช่นนี้ก็เพื่อให้โปรแกรมมีความอ่อนตัว และง่ายต่อการแก้ไขตำแหน่งแอดเดรสต่างๆ เพราะจะทำให้เราสามารถเปลี่ยนค่าตำแหน่งแอดเดรสทั้งหมดจากจุดที่ใช้ประกาศค่า Base Address เพียงจุดเดียวเท่านั้น ตัวอย่างเช่น ถ้าค่า Base Address ของการ์ดมีค่าเป็น E800H-E8FFH เมื่อเราต้องการอ้างอิงพอร์ต PA ของ 8255 ตัวที่ 1 ก็จะต้องอ้างตำแหน่งพอร์ตเป็น E8C0H แต่ถ้าค่า Base Address ของการ์ดมีค่าเป็น 1400H-14FFH เราก็จะต้องเปลี่ยนการอ้างตำแหน่งของพอร์ตเป็น 14C0H แทน ซึ่งถ้าเราใช้วิธีการอ้างตำแหน่งแอดเดรสตรงๆนั้น จะเห็นได้ว่าเมื่อมีการเปลี่ยนแปลงค่า Base Address ไปจากเดิมเราก็จะต้องทำการแก้ไข Code ของโปรแกรมในส่วนของการอ้างตำแหน่งแอดเดรสทุกจุดในโปรแกรม ซึ่งถ้ามีการอ้างตำแหน่งไว้เป็นจำนวนมากก็จะทำให้เสียเวลาในการแก้ไข Code ของโปรแกรมเป็นอย่างมาก แต่เมื่อเราเปลี่ยนมาใช้วิธีการประกาศเป็นค่าตัวแปรแทนดังตัวอย่างข้างต้นนั้น เมื่อต้องการอ้างตำแหน่งแอดเดรสก็ใช้การอ้างผ่านชื่อนั้นๆได้ทันที เช่น ถ้าต้องการอ้างตำแหน่งแอดเดรสถึงพอร์ต PA ของ 8255 ตัวที่ 1 ก็ใช้วิธีการอ้างตำแหน่งแอดเดรสเป็น BaseAddress+PA1 ซึ่งก็จะมีความหมายเหมือนกันการอ้างตำแหน่งแอดเดรสเป็น E8C0H เนื่องจากเราประกาศค่า Base Address ไว้เท่ากับ E800H และประกาศค่า PA1 ไว้เท่ากับ C0H ซึ่งเมื่อมีการเปลี่ยนค่าตำแหน่ง Base Address ไปจากเดิมเราก็เพียงแต่เข้าไปแก้ไขค่าของ Base Address เพียงจุดเดียวเท่านั้น เนื่องจากตำแหน่งพอร์ต Address ต่างๆจะยังคงมีค่าตำแหน่ง Offset เหมือนเดิมไม่เปลี่ยนแปลง

```
Const ON_Bit0 = $01;           // XXXX XXXX OR 0000 0001 = XXXX XXX1
Const OFF_Bit0 = $FE;          // XXXX XXXX AND 1111 1110 = XXXX XXX0
Const ON_Bit1 = $02;           // XXXX XXXX OR 0000 0010 = XXXX XX1X
Const OFF_Bit1 = $FD;          // XXXX XXXX AND 1111 1101 = XXXX XX0X
Const ON_Bit2 = $04;           // XXXX XXXX OR 0000 0100 = XXXX X1XX
Const OFF_Bit2 = $FB;          // XXXX XXXX AND 1111 1011 = XXXX X0XX
Const ON_Bit3 = $08;           // XXXX XXXX OR 0000 1000 = XXXX 1XXX
Const OFF_Bit3 = $F7;          // XXXX XXXX AND 1111 0111 = XXXX 0XXX
Const ON_Bit4 = $10;           // XXXX XXXX OR 0001 0000 = XXX1 XXXX
Const OFF_Bit4 = $EF;          // XXXX XXXX AND 1110 1111 = XXX0 XXXX
Const ON_Bit5 = $20;           // XXXX XXXX OR 0010 0000 = XX1X XXXX
Const OFF_Bit5 = $DF;          // XXXX XXXX AND 1101 1111 = XX0X XXXX
Const ON_Bit6 = $40;           // XXXX XXXX OR 0100 0000 = X1XX XXXX
Const OFF_Bit6 = $BF;          // XXXX XXXX AND 1011 1111 = X0XX XXXX
Const ON_Bit7 = $80;           // XXXX XXXX OR 1000 0000 = 1XXX XXXX
Const OFF_Bit7 = $7F;          // XXXX XXXX AND 0111 1111 = 0XXX XXXX
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ในส่วนนี้เป็นการประกาศค่าคงที่สำหรับ ใช้ควบคุมการ ON/OFF ของ Output ต่างๆ โดยจะประกาศเป็นค่าคงที่สำหรับใช้ ON และ OFF ข้อมูลในตำแหน่งบิตต่างๆ จาก 0-7 โดยจะประกาศเป็นค่าสำหรับใช้ตั้ง ON หรือ Set ข้อมูลให้เป็น "1" และสั่ง OFF หรือ Reset ข้อมูลให้เป็น "0"

ทั้งนี้ก็เนื่องมาจากว่าในการอ่านเขียนข้อมูลกับพอร์ต I/O นั้นจะต้องกระทำแบบ Byte เสมอ แต่สำหรับในการใช้งานบางกรณีนั้น อาจมีความจำเป็นต้องแยกการทำงานของ Input / Output ออกเป็นบิตๆ แบบอิสระต่อกัน แต่ก็ยังต้องใช้การอ่าน และ เขียน ข้อมูลครั้งละ 1 Byte เหมือนเดิม ตัวอย่างเช่น ถ้าต้องการให้ Output บิต 0 ของ PA มีค่าเป็น "1" เราก็จะต้องทำการส่งเขียนข้อมูลออกไปยัง Port-PA ด้วยค่าข้อมูลที่ บิต0 มีค่าเป็น "1" แต่ในการเขียนข้อมูลนั้นขนาดของข้อมูลจะต้องมีขนาด 8 บิต หรือ 1 Byte ซึ่งถ้าสั่ง เขียนข้อมูลออกไปโดยไม่คำนึงถึงข้อมูลบิตอื่นๆ เลย ก็อาจสั่งเขียนข้อมูลด้วยค่า 01H หรือ FFH ก็จะมีผลทำให้ Output บิต 0 ของ PA มีค่าเป็น "1" เหมือนกัน เพียงแต่ว่า Output บิตอื่นๆ คือ บิต1 ถึง บิต7 อาจถูกเปลี่ยนแปลงสถานะทาง Output ไปจากเดิมได้ ดังนั้นการส่งเปลี่ยนแปลงค่าสถานะทาง Output ของบิตใดบิตหนึ่งเพียงบิตเดียว โดยไม่คำนึงถึงสถานะเดิมของบิตอื่นๆ นั้น ก็อาจส่งผลให้การทำงานของโปรแกรมเกิดความผิดพลาดขึ้นได้โดยง่าย

ด้วยเหตุผลอันนี้ เราจึงจำเป็นต้องใช้กระบวนการทางลอจิก เข้ามาช่วยแก้ปัญหานี้ โดยใช้เทคนิคของการ OR และ AND เข้ามาช่วย โดยถ้าเราต้องการสั่งให้ Output ใด ON ก็จะต้องนำค่าสถานะ Output ครึ่งสุดท้ายมาทำการ OR กับค่า BITX_ON เช่น ถ้าต้องการให้ Output Port-PA บิต 0 มีสถานะเป็น ON ก็จะต้องนำค่าสถานะเก่าของ Output ของ Port-PA มาทำการ OR กับค่า BIT0_ON หรือ 00000001B (01H) จากนั้นจึงนำค่าผลลัพธ์ที่ได้จากการ OR กันเขียนกลับออกไปยัง Port-PA ใหม่อีกครั้งหนึ่ง ซึ่งการกระทำเช่นนี้จะทำให้ค่า Output ของ PA0 มีสถานะเป็น ON หรือ "1" ส่วน PA1..PA7 จะยังคงมีค่าเหมือนเดิมไม่เปลี่ยนแปลง

และสำหรับกรณีที่ต้องการให้สถานะของ Output มีค่าเป็น OFF หรือ "0" ก็จะต้องนำค่าสถานะเก่าของ Output มาทำการ AND กับค่า BITX_OFF แทน เช่น ถ้าต้องการให้ Output บิต PA0 มีสถานะเป็น OFF ก็จะต้องนำค่าสถานะเก่าของ Output ของ Port-PA มาทำการ AND กับค่า BIT0_OFF หรือ 11111110B หรือ FEH จากนั้นจึงนำค่าผลลัพธ์ที่ได้จากการ AND กันเขียนกลับออกไปยัง Port-PA ใหม่อีกครั้งหนึ่ง ซึ่งการกระทำเช่นนี้จะทำให้ค่า Output ของ PA0 มีสถานะเป็น OFF หรือ "0" ส่วน PA1..PA7 จะยังคงมีค่าเหมือนเดิมไม่เปลี่ยนแปลง

```
Const OutputOFF = CIMenu; // Color of Output OFF ("0") Status
Const OutputON = CIRed; // Color of Output ON ("1") Status
```

ในส่วนนี้จะเป็นการประกาศสีของ Shape ซึ่งทำหน้าที่เหมือน LED สำหรับใช้แสดงผลการทำงานของ Relay และ Output โดยเราจะแทนการทำงาน (ON) ของ Output ด้วยสีแดง และแสดงการหยุดทำงาน (OFF) ของ Output ด้วยสีพื้นเหมือนสีของเมนู

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

การ OR เป็นการกระทำทางลอจิก ซึ่งจะให้ผลลัพธ์มีค่าเป็น "1" เมื่อตัวตั้งหรือตัวกระทำมีค่าเป็น "1" เพียงตัวใดตัวหนึ่งหรือเป็น "0" ทั้งคู่ ดังนั้นเมื่อต้องการให้ผลลัพธ์ของบิตใดมีค่าเป็น "1" ก็ให้นำข้อมูลที่บิตข้อมูลนั้นๆ มีค่าเป็น "1" ไปทำการ OR กับค่าตัวตั้งก็จะได้ผลลัพธ์เป็น "1" ตามต้องการดังตัวอย่าง

BIT ข้อมูล	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ข้อมูลเก่า	?	?	?	?	?	?	?	?
ค่าการ OR	0	0	0	0	0	0	0	<u>1</u>
ค่าผลลัพธ์	?	?	?	?	?	?	?	<u>1</u>



BIT ข้อมูล	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ข้อมูลเก่า	1	0	1	0	1	0	1	<u>0</u>
ค่าการ OR	0	0	0	0	0	0	0	<u>1</u>
ค่าผลลัพธ์	1	0	1	0	1	0	1	<u>1</u>

ตารางแสดง คุณสมบัติการการกระทำทางลอจิกแบบ OR

การ AND เป็นการกระทำทางลอจิก ซึ่งจะให้ผลลัพธ์มีค่าเป็น "0" เมื่อตัวตั้งหรือตัวกระทำเพียงตัวใดตัวหนึ่งมีค่าเป็น "0" ดังนั้นเมื่อต้องการให้ผลลัพธ์ของบิตใดมีค่าเป็น "0" ก็ให้นำข้อมูลที่บิตข้อมูลนั้นๆ มีค่าเป็น "0" ไปทำการ AND กับค่าตัวตั้งก็จะได้ผลลัพธ์เป็น "0" ตามต้องการดังตัวอย่าง

BIT ข้อมูล	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ข้อมูลเก่า	?	?	?	?	?	?	?	?
ค่าการ AND	1	1	1	1	1	1	1	<u>0</u>
ค่าผลลัพธ์	?	?	?	?	?	?	?	<u>0</u>



BIT ข้อมูล	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ข้อมูลเก่า	1	0	1	0	1	0	1	<u>1</u>
ค่าการ AND	1	1	1	1	1	1	1	<u>0</u>
ค่าผลลัพธ์	1	0	1	0	1	0	1	<u>0</u>

ตารางแสดง คุณสมบัติการการกระทำทางลอจิกแบบ AND

```

Procedure TForm1.SETUP_8255Click(Sender: TObject);
Var  SetupData : Byte;
Begin
  BaseAddress := IO_BaseAddress;           // Set I/O Base Address
  // Initial Reset and 8255 Bus Cycle //
  SetupData := Inp32(BaseAddress+PIB);      // Read PIB Reset Port
  SetupData := SetupData AND OFF_BIT0;     // Bit0 = EXTRST# = "0" (Reset:RES#)
  SetupData := SetupData OR  ON_BIT5;      // Bit5:4 = 11 = PIB Cycle Slowest
  SetupData := SetupData OR  ON_BIT4;
  Out32(BaseAddress+PIB,SetupData);        // Active RES# & Relay
  // Initial Data (AUX) For CS# and Relay //
  SetupData := Inp32(BaseAddress+AUXD);    // Read Aux Data Port
  SetupData := SetupData AND OFF_BIT0;    // Bit0 = Aux0 = "0" (Enable CS)
  SetupData := SetupData OR  ON_BIT4;     // Bit4 = Aux4 = "1" (Relay OFF)
  Out32(BaseAddress+AUXD,SetupData);      // Active Chips Select & Relay
  LEDRelay.Brush.Color := OutputOFF;     // LED Relay = OFF
  // Initial Direction (AUX) For CS# and Relay //
  SetupData := Inp32(BaseAddress+AUXC);    // Read Aux Port Direction
  SetupData := SetupData OR ON_BIT4;      // Aux4 = "1" = Output
  SetupData := SetupData OR ON_BIT0;     // Aux0 = "1" = Output
  Out32(BaseAddress+AUXC,SetupData);      // Setup Aux Direction
  // Initial 8255#1 = All Output Port //
  Setup_8255.Enabled := False;           // Disable Setup After Setup
  Out32(BaseAddress+PCC1,$80);           // Write Control Port 8255#1
  RELAY.Enabled := True;                 // Enable Relay Control
  PA01.Enabled := True;                  // Enable Output Control
  PA11.Enabled := True;
  PA21.Enabled := True;
  PA31.Enabled := True;
  PA41.Enabled := True;
  PA51.Enabled := True;
  PA61.Enabled := True;
  PA71.Enabled := True;
End;

```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ในส่วนนี้เป็นการทำงานเมื่อทำการคลิกเมาส์ที่ปุ่ม Setup ซึ่งโปรแกรมจะเริ่มต้นด้วยการส่งอ่านค่าสถานะเดิมจากรีจิสเตอร์ PIB ของ TIGER320 จากนั้นจึงส่งเปลี่ยนข้อมูลโดยกำหนดให้ บิต 5 และบิต 4 มีค่าเป็น "1" ทั้งคู่ และให้บิต 0 มีค่าเป็น "0" ส่วนบิตอื่นๆที่เหลือให้มีค่าคงเดิมไม่เปลี่ยนแปลง แล้วจึงเขียนค่าข้อมูลกลับไปยัง PIB ซึ่งตรงส่วนนี้เป็นการสั่งให้ชิพ TRGER320 สร้างสัญญาณรีเซ็ตแบบ Active "0" พร้อมกับกำหนดให้สัญญาณ Bus มีค่าความเร็วเท่ากับ 15 Cycle ของความเร็วจาก PCI Bus

จากนั้นเป็นการส่งอ่านค่าสถานะเดิมจากรีจิสเตอร์ AUXD ของ TIGER320 ซึ่งเป็นรีจิสเตอร์สำหรับควบคุมการทำงานของสัญญาณ AUX0..7 โดยจะทำการสั่งให้ บิต 0 (AUX0) มีค่าเป็น "0" เพื่อเปิดการทำงานของสัญญาณ Chip Select หรือ CS# และกำหนดให้ บิต 4 (AUX4) มีค่าเป็น "1" เพื่อสั่ง OFF การทำงานของ Relay ส่วนข้อมูลบิตอื่นๆให้มีค่าเหมือนเดิมไม่เปลี่ยนแปลง แล้วจึงเขียนข้อมูลกลับคืนไปยังรีจิสเตอร์ AUXD

ส่วนถัดไปเป็นการส่งอ่านค่าสถานะเดิมจากรีจิสเตอร์ AUXC ของ TIGER320 ซึ่งเป็นรีจิสเตอร์สำหรับกำหนดหน้าที่การทำงานของสัญญาณ AUX0..7 ว่าต้องการให้เป็น Input หรือ Output โดยจะทำการสั่งให้ บิต 4 และบิต 0 มีค่าเป็น "1" ทั้ง 2 บิต เพื่อเป็นการกำหนดให้สัญญาณ AUX4 และ AUX0 ทำหน้าที่เป็น Output ส่วนบิตอื่นๆยังคงให้มีค่าคงเดิมไม่มีการเปลี่ยนแปลง

ซึ่งในส่วนของชิพ TIGER320 นั้นตามปรกติแล้วจะมีรีจิสเตอร์สำหรับแสดงสถานะการทำงานและรีจิสเตอร์สำหรับควบคุมการทำงานต่างๆทั้งหมดรวมกันมากถึง 48 รีจิสเตอร์ แต่ส่วนที่จำเป็นและเกี่ยวข้องกับฟังก์ชัน PIB ที่นำมาใช้กับการ์ด ET-PCI8255 V3.0 นั้น จะมีเพียง 3 รีจิสเตอร์ดังกล่าวมาแล้วเท่านั้น ส่วนรีจิสเตอร์อื่นๆที่เหลือจะกำหนดให้มีค่าตามมาตรฐานหรือ Default ไว้ไม่เปลี่ยนแปลงดังนั้นเราจึงไม่เข้าไปเปลี่ยนแปลงแก้ไขข้อมูลใดๆกับรีจิสเตอร์อื่นๆที่เหลืออยู่นอกเหนือจากรีจิสเตอร์จำนวน 3 รีจิสเตอร์ที่กล่าวถึงไปแล้วเท่านั้น

หลังจากนั้นจะเป็นการส่งควบคุมการทำงานของ 8255 ให้ทำหน้าที่เป็น Output ทั้งหมดทุกพอร์ต โดยการส่งเขียนคำสั่ง Control Word ค่า 80H ไปยังพอร์ตควบคุมของ 8255 ตัวที่ 1 ส่วน 8255 อีก 2 ตัวจะยังไม่เข้าไปยุ่งเกี่ยวใดๆ จากนั้นโปรแกรมก็จะสั่งให้ค่าสถานะ Output ของพอร์ต PA0..7 มีค่าเริ่มต้นเป็น "0" ทั้งหมด

สำหรับส่วนสุดท้ายเป็นการสั่งปิดการทำงานของปุ่ม Setup ไว้เนื่องจากไม่ต้องการให้สั่ง Setup ซ้ำใหม่อีก เพราะการสั่ง Setup ชิพ TIGER320 และ 8255 นั้นควรกระทำเพียงครั้งเดียวเท่านั้น ซึ่งหลังจากสั่งปิดการทำงานของปุ่ม Setup แล้วก็จะสั่งเปิดการทำงานของปุ่มควบคุม Output ต่างๆ ทั้งของ Relay และ PA0..PA7 เพื่อให้ผู้ใช้สามารถสั่งงาน Output ต่างๆได้ตามต้องการ

```

Procedure TForm1.RelayClick(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+AuxD);           // Read Output Latch AUX
  CheckData := InOutData AND ON_BIT4;           // Check Last AUX4 Status
  If CheckData = $10 Then
  Begin
    InOutData := InOutData AND OFF_BIT4;        // AUX4 = "0" (1110 1111)
    LEDRelay.Brush.Color := OutputON;          // LED RELay = ON
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT4;          // AUX4 = "1" (0001 0000)
    LEDRelay.Brush.Color := OutputOFF;         // LED Relay = OFF
  End;
  Out32(BaseAddress+AuxD,InOutData);           // Update AuxD
End;
    
```

ในส่วนนี้เป็นการทำงานของโปรแกรมเมื่อทำการตั้งคลิกเมาส์ที่ปุ่ม Relay แต่ละครั้ง ซึ่งจะเป็นการตั้ง เปิด หรือ ปิด การทำงานของ Relay โดยถ้า Relay อยู่ในสถานะ ON เมื่อตั้งคลิกเมาส์ที่ปุ่ม Relay จะเป็นการตั้งให้ Relay OFF แต่ถ้า Relay อยู่ในสถานะ OFF ก็จะเป็นการตั้งให้ Relay กลับมาอยู่ในสถานะ ON ใหม่

โดยการทำงานของโปรแกรมส่วนนี้จะเริ่มต้นด้วยการส่งอ่านค่าสถานะของรีจิสเตอร์ AUXD ซึ่งเก็บค่าสถานะทาง Output ของสัญญาณ AUX0..7 ไว้ จากนั้นจึงส่งตรวจสอบเฉพาะ บิต 4 (AUX4) ซึ่งเป็นสัญญาณที่ใช้ในการควบคุมการทำงานของ Relay โดยถ้าสัญญาณ AUX4 มีค่าเป็น "0" จะหมายถึง Relay อยู่ในสถานะ ON แต่ถ้าสัญญาณ AUX4 มีค่าเป็น "1" จะหมายถึง Relay อยู่ในสถานะ OFF อยู่ โดยโปรแกรมจะทำการนำค่าข้อมูลที่ส่งอ่านมาได้จากรีจิสเตอร์ AUXD มาทำการ AND ด้วยค่าของบิตข้อมูลที่ทำให้ ผลลัพธ์ บิตอื่น ๆ มีค่าเป็น "0" ทั้งหมด ยกเว้น บิต 4 (00010000B) จากนั้นจึงตรวจสอบผลลัพธ์ที่ได้ว่ามีค่าเท่ากับค่าที่นำไป AND (00010000B) อยู่หรือไม่ ซึ่งถ้าผลลัพธ์มีค่าเหมือนกับค่าที่นำไป AND ก็แสดงว่า บิต 4 มีค่าเป็น "1" อยู่ แต่ถ้าไม่เท่าก็แสดงว่า บิต 4 มีค่าเป็น "0" อยู่ โดยถ้าพบว่าค่า บิต 4 มีค่าเป็น "1" ก็จะตั้งเปลี่ยนข้อมูลเฉพาะ บิต 4 ให้กลับเป็น "0" พร้อมกับตั้งเปลี่ยนค่าสีของ Shape ที่ใช้แสดงสถานะการทำงานของ Relay ให้อยู่ในสถานะ ON ("0" หมายถึง Relay ON)

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

แต่ถ้าพบว่าค่าของ บิต 4 มีค่าเป็น “0” อยู่ก็จะสั่งเปลี่ยนข้อมูลเฉพาะ บิต 4 ให้มีค่าเป็น “1” พร้อมกับสั่งเปลี่ยนสีของ LED ที่ใช้แสดงสถานการณ์การทำงานของ Relay ให้มีค่าเป็น OFF (“1” หมายถึง Relay OFF) แล้วจึงส่งเขียนข้อมูลที่ได้สั่งเปลี่ยนแปลงค่าใน บิต 4 แล้วกลับออกไปยังรีจิสเตอร์ AUXD ซึ่งก็จะทำให้สถานการณ์การทำงานของ Relay เปลี่ยนแปลงไปตามการคลิกเมาส์แต่ละครั้ง

```
Procedure TForm1.PA01Click(Sender: TObject);
Var  InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT0;           // Check Last PA0 Status
  If CheckData = $01 Then
  Begin
    InOutData := InOutData AND OFF_BIT0;        // PA0 = "0" (1111 1110)
    LEDPA01.Brush.Color := OutputOFF;           // LED PA0 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT0;          // PA0 = "1" (0000 0001)
    LEDPA01.Brush.Color := OutputON;           // LED PA0 of 8255#1 = ON
  End;
  Out32(BaseAddress+PA1,InOutData);             // Update Port-A
End;
```

สำหรับส่วน สุดท้ายนี้เป็นส่วนของการควบคุมการทำงานของ Output ของ Port-PA ทั้ง 8 บิต คือ PA0..7 โดย จะแสดงให้เห็นเพียงเฉพาะส่วนของ PA0 เท่านั้น ส่วน Output อีก 7 บิต จะมี หลักการทำงานของโปรแกรมที่เหมือนกันทุกประการ ต่างกันที่ตำแหน่งของบิตข้อมูลเท่านั้นเอง ซึ่งความจริงแล้วการทำงานของโปรแกรมน้อยในส่วนนี้จะคล้ายกับโปรแกรมน้อยที่ใช้ในการควบคุมการทำงานของ Relay ทุกอย่าง โดยการทำงานของโปรแกรมส่วนนี้จะเริ่มต้นด้วยการสั่งอ่านค่าสถานะ Output ของ Port-PA จากนั้นจึงส่งตรวจสอบเฉพาะ บิต 0 (PA0) โดยถ้าสัญญาณ PA0 มีค่าเป็น “1” จะหมายถึง Output อยู่ในสถานะ ON แต่ถ้าสัญญาณ PA0 มีค่าเป็น “0” จะหมายถึง Output อยู่ในสถานะ OFF อยู่ โดยโปรแกรมจะทำการนำค่าข้อมูลที่สั่งอ่านมาได้จาก Port-PA มาทำการ AND ด้วยค่าของบิตข้อมูลที่ทำให้ ผลลัพธ์ บิตอื่นๆมีค่าเป็น “0” ทั้งหมดยกเว้น บิต 0 (00000001B) จากนั้นจึงตรวจสอบ

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ผลลัพธ์ที่ได้ว่ามีค่าเท่ากับค่าที่นำไป AND (00000001B) อยู่หรือไม่ ซึ่งถ้าผลลัพธ์มีค่าเหมือนกับค่าที่นำไป AND ก็แสดงว่า บิต 0 หรือ PA0 มีค่าเป็น "1" อยู่ แต่ถ้าไม่เท่าก็แสดงว่า บิต 0 หรือ PA0 มีค่าเป็น "0" อยู่ ซึ่งเมื่อพบว่าค่าสถานะ Output เก้าของ PA0 มีค่าเป็น "1" ก็จะสั่งเปลี่ยนข้อมูลเฉพาะ บิต 0 ให้กลับเป็น "0" พร้อมกับสั่งเปลี่ยนค่าสีของ Shape ที่ใช้แสดงสถานะการทำงานของ Output ให้อยู่ในสถานะ OFF ("0" หมายถึง Output OFF)

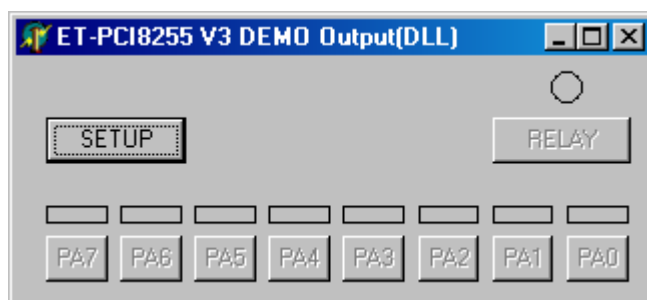
แต่ถ้าพบว่าค่าของ PA0 มีค่าเป็น "0" อยู่ก็จะสั่งเปลี่ยนข้อมูลเฉพาะ บิต 0 ให้มีค่าเป็น "1" พร้อมกับสั่งเปลี่ยนสีของ LED ที่ใช้แสดงสถานะการทำงานของ Output ให้มีค่าเป็น ON ("1" หมายถึง Output ON) แล้วจึงสั่งเขียนข้อมูลที่ได้สั่งเปลี่ยนแปลงค่าใน บิต 0 แล้วกลับออกไป Port-PA ซึ่งก็จะทำให้สถานะการทำงานของ Output เปลี่ยนแปลงไปตามการคลิกเมาส์แต่ละครั้ง

สำหรับโปรแกรมย่อยที่ใช้ควบคุมการทำงานของ Output ของ Port-PA บิตอื่นๆ ก็จะมีลักษณะของคำสั่ง และการทำงานของ โปรแกรมที่คล้ายกันนี้ทุกประการ แต่จะมีความแตกต่างกันเฉพาะส่วนของข้อมูลที่ใช้ในการตรวจสอบและสั่ง ON/OFF การทำงานของ Output ของแต่ละบิตเท่านั้นเอง ตัวอย่าง เช่นถ้าเป็นของ PA1 ก็จะใช้ค่าข้อมูลที่ได้ประกาศไว้สำหรับใช้กับ บิต1 คือ BIT1_ON และ BIT1_OFF ส่วน LED สถานะก็เปลี่ยนเป็น LEDPA11 แทนดังนี้ เป็นต้น

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

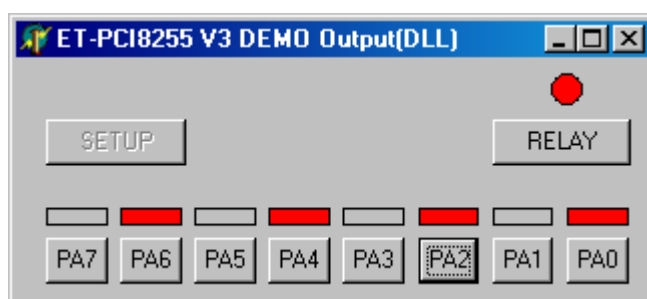
การทดสอบการทำงานของโปรแกรมตัวอย่างที่ 1.1

1. ก่อนทำการทดสอบให้ต่อบอร์ด ET-TEST I/O เข้ากับพอร์ต PA ของ 8255#1 เพื่อใช้แสดงการทำงานของ Output จาก 8255 แต่ถ้าไม่มีก็อาจไม่ต้องต่อก็ได้
2. สั่ง Run โปรแกรมที่เขียนเสร็จแล้วโดยกด F5 ซึ่งในครั้งแรกจะเห็นว่าปุ่มควบคุมการทำงานของ RELAY และ PA0 ถึง PA7 จะถูกปิดการทำงานไว้ จะมีเฉพาะปุ่ม SETUP เท่านั้นที่ถูกเปิดการทำงานอยู่ ทั้งนี้ก็เพื่อเป็นการบังคับให้ผู้ใช้ต้องสั่ง Setup การทำงานของการ์ดก่อนเป็นอันดับแรกดังรูป



แสดงรูป ผลการ Run โปรแกรมตัวอย่างในครั้งแรก

3. เริ่มต้นทดสอบการทำงานของโปรแกรมโดยการ คลิกเมาส์ที่ปุ่ม SETUP เพื่อสั่ง Setup การทำงานของชิพ TIGER320 และ 8255 ตัวที่ 1 ซึ่งจะทำให้ปุ่ม SETUP ถูกปิดการทำงาน ขณะเดียวกันปุ่ม ควบคุมการทำงานของ RELAY และ PA0 ถึง PA7 จะถูกเปิดการทำงานขึ้นมาทันที ให้ทดสอบคลิกเมาส์ที่ปุ่ม RELAY สลับไปมา ซึ่งจะต้องเห็นการทำงานของ Relay สลับการทำงานระหว่าง ON/OFF ตามการคลิกเมาส์แต่ละครั้ง โดยจะสังเกตเห็น LED แสดงสถานะการทำงานของ Relay สลับสีไปมาระหว่างสีแดง กับสีพื้นของเมนูตามจังหวะการ ON/OFF ด้วยทุกครั้ง และถ้าต้องการทดสอบการทำงานของ Output จาก Port-PA ก็ให้ทดลองคลิกเมาส์ที่ปุ่มควบคุมการทำงานของ PA0 ถึง PA7 คุณจะได้ผลการทำงานในทำนองเดียวกัน ซึ่งถ้ามีการต่อบอร์ด ET-TEST I/O ไว้ที่ Port-PA ของ 8255#1 ด้วยก็จะเห็น LED ติดและดับด้วยทันที ดังรูป



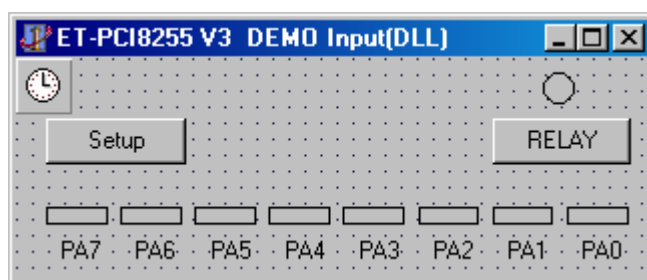
แสดงรูป ผลการทดสอบโปรแกรมของ ตัวอย่างที่ 1.1

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ตัวอย่างที่1.2 DEMO Input (เรียกใช้ Library ไฟล์จาก InpOut32.DLL)

สำหรับโปรแกรม ตัวอย่างที่1.2 จะเป็นการแสดงให้เห็นการประยุกต์ใช้งาน 8255 ให้ทำหน้าที่เป็น Input โดยจะแสดงให้เห็นการใช้พอร์ต PA ในการตรวจสอบค่าสถานะการทำงานของ Input จากภายนอก สำหรับขั้นตอนการเขียนโปรแกรมของการ์ด ET-PCI8255 V3 มีรายละเอียดดังนี้

1. เปิด Project ใหม่โดยไปที่ File > New Application จะได้ฟอร์มเปล่าขึ้นมา 1 ฟอร์ม
2. เมื่อฟอร์มขึ้นมา ให้นำคอนโทรลชนิดต่างๆ มาจัดวางดังรูป



รูปแสดง การจัดวาง Component บน Form ของโปรแกรม ตัวอย่างที่1.2

โดยในตัวอย่างนี้จะใช้ Component ทั้งหมด 4 ชนิดด้วยกัน คือ Command Button, Shape ,Label และTimer ซึ่ง Command Button จะมีทั้งหมด 2 ชุด ใช้สำหรับ Setup การทำงานของการ์ด ET-PCI8255 V3 จำนวน 1 ชุด โดยเมื่อกดปุ่ม SETUP จะเป็นการสั่ง Initial การทำงานของการ์ด ET-PCI8255 V3 โดยจะเริ่มต้นด้วยการเปิดการทำงานของสัญญาณ Chip Select และปิดการทำงานของ Relay ไว้ พร้อมกับ Initial การทำงานของ 8255 ตัวที่ 1 ให้ทำหน้าที่เป็น Input Port ทั้ง 3 พอร์ต (PA,PB และ PC ทำหน้าที่เป็น Input) ส่วน Command Button อีก 1 ตัว คือ RELAY จะใช้สำหรับสั่ง ON และ OFF การทำงานของ Relay

สำหรับ Shape นั้นจะใช้สำหรับทำหน้าที่เป็นเสมือน หลอด LED หรือไฟแสดง สถานะ การทำงานของ Relay และ Input ของ PA0..PA7 โดยให้ LED ของ Relay มีลักษณะเป็นทรงกลม และให้ LED ของ Input 8255 มีลักษณะเป็นสี่เหลี่ยม ส่วนการแสดงผลของ LED ในกรณีของ LED Output Relay ถ้า ON จะให้สถานะเป็นสีแดง แต่ถ้า Relay OFF จะให้สถานะเป็นสีพื้นของ Form แทน สำหรับในกรณีของ LED Input นั้นจะให้การทำงานเป็นสีเขียว สลับ กับ สีพื้นของเมนู แทน ส่วน Label นั้นจะใช้สำหรับแสดงชื่อและตำแหน่งของ Input จาก PA0 ถึง PA7 ของ 8255 บน Form เท่านั้น

สำหรับ Timer นั้นจะถูกใช้สำหรับทำหน้าที่เป็นตัวนับเวลาสำหรับการอ่านค่าสถานะ Input จาก Port-PA มาแสดงผลการทำงานให้เห็น ที่หลอดแสดงผล LED บนฟอร์มของโปรแกรม เพื่อให้การทำงานของโปรแกรมเป็นไปอย่างอัตโนมัติ ไม่ต้องเสียเวลามากมายสั่งอ่านค่า Input ด้วยการกดปุ่ม Button เหมือนการสั่งงาน Output

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

3. ทำการกำหนดค่า Properties ให้กับคอนโทรลต่างๆ ดังนี้

ชื่อ	คอนโทรล	พรีอเพอร์ตี้	ค่าที่กำหนด
Form1	Form	Name	Form1
		Border Style	bsSizeable
		Caption	ET-PCI8255 V3 DEMO Input(DLL)
SETUP	Command Button	Name	SETUP
		Enable	True
		Caption	SETUP
PA0 . . PA7	Label	Name	PA01Name,PA11Name,...PA71Name
		Enable	True
		Caption	PA0,PA1,PA2,...PA7
PA7	Shape	Name	LEDPA01,LEDPA11,LEDPA21,...LEDPA71
		Fill Style	Solid
		Shape	Rectangle
RELAY	Command Button	Name	RELAY
		Enable	False
		Caption	RELAY
	Shape	Name	LEDRelay
		Fill Style	Solid
		Shape	Circle
Timer1	Timer	Name	Timer1
		Enable	False
		Interval	100

สำหรับการกำหนดค่า พรีอเพอร์ตี้ ของคอนโทรลต่างๆนั้น จะแสดงเฉพาะค่าที่เปลี่ยนแปลงไปจากค่า Default เดิมเท่านั้น โดยในส่วนของ Command Button นั้นจะเปิดการทำงาน (Enabled) ไว้เฉพาะที่เป็นของ Setup เพียงตัวเดียวเท่านั้น ส่วน Command Button ของ Relay และ Timer ต้องปิด(Disable) ไว้ก่อน เพื่อบังคับให้ ผู้ใช้งานต้องทำการ Setup การทำงานให้กับการ์ดเสียก่อน จากนั้นจึงจะสามารถสั่งงาน Output Relay ได้ โดยการ ทำงานของ Command Button ของ Relay และ Timer1 นั้นจะถูกเปิดการทำงานเมื่อทำการสั่ง Setup แล้ว

4. เขียนโค้ดคำสั่งเพื่อความคุมการทำงานของโปรแกรกดังต่อไปนี้

```
{ Demo2 Input Demo Access Input/Output Library "INPOUT32.DLL" }  
Unit Unit1;  
Interface  
Uses  Windows,Messages,SysUtils,Classes,Graphics,Controls,Forms,Dialogs,StdCtrls,ExtCtrls;  
Type  TForm1 = class(TForm)  
      SETUP_8255: TButton;  
      RELAY: TButton;  
      LEDRELAY: TShape;  
      LEDPA01: TShape;  
      LEDPA11: TShape;  
      LEDPA21: TShape;  
      LEDPA31: TShape;  
      LEDPA41: TShape;  
      LEDPA51: TShape;  
      LEDPA61: TShape;  
      LEDPA71: TShape;  
      PA0Name: TLabel;  
      PA1Name: TLabel;  
      PA2Name: TLabel;  
      PA3Name: TLabel;  
      PA4Name: TLabel;  
      PA5Name: TLabel;  
      PA6Name: TLabel;  
      PA7Name: TLabel;  
      Timer1: TTimer;  
      Procedure SETUP_8255Click(Sender: TObject);  
      Procedure RELAYClick(Sender: TObject);  
      Procedure Timer1Timer(Sender: TObject);  
end;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Private
  { Private declarations }
Public
  { Public declarations }
End;

Var
  Form1      : TForm1;
  BaseAddress : Word;

  Procedure Out32(Port:Word; Data:Byte); Stdcall; External 'InpOut32.DLL';
  Function  Inp32(Port:Word):Byte; Stdcall; External 'InpOut32.DLL';

Implementation

Const IO_BaseAddress = $E800; // I/O Base Address
Const PIB = $00; // Reset & PIB Cycle
Const AUXC = $02; // Aux Direction Port
Const AUXD = $03; // Aux Data Port

Const PA1 = $C0; // Port-A 8255#1
Const PB1 = $C4; // Port-B 8255#1
Const PC1 = $C8; // Port-C 8255#1
Const PCC1 = $CC; // Port Control 8255#1
Const PA2 = $D0; // Port-A 8255#2
Const PB2 = $D4; // Port-B 8255#2
Const PC2 = $D8; // Port-C 8255#2
Const PCC2 = $DC; // Port Control 8255#2
Const PA3 = $E0; // Port-A 8255#3
Const PB3 = $E4; // Port-B 8255#3
Const PC3 = $E8; // Port-C 8255#3
Const PCC3 = $EC; // Port Control 8255#3

Const ON_Bit0 = $01; // XXXX XXXX OR 0000 0001 = XXXX XXX1
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Const OFF_Bit0 = $FE;           // XXXX XXXX AND 1111 1110 = XXXX XXX0
Const ON_Bit1 = $02;           // XXXX XXXX OR 0000 0010 = XXXX XX1X
Const OFF_Bit1 = $FD;         // XXXX XXXX AND 1111 1101 = XXXX XX0X
Const ON_Bit2 = $04;           // XXXX XXXX OR 0000 0100 = XXXX X1XX
Const OFF_Bit2 = $FB;         // XXXX XXXX AND 1111 1011 = XXXX X0XX
Const ON_Bit3 = $08;           // XXXX XXXX OR 0000 1000 = XXXX 1XXX
Const OFF_Bit3 = $F7;         // XXXX XXXX AND 1111 0111 = XXXX 0XXX
Const ON_Bit4 = $10;           // XXXX XXXX OR 0001 0000 = XXX1 XXXX
Const OFF_Bit4 = $EF;         // XXXX XXXX AND 1110 1111 = XXX0 XXXX
Const ON_Bit5 = $20;           // XXXX XXXX OR 0010 0000 = XX1X XXXX
Const OFF_Bit5 = $DF;         // XXXX XXXX AND 1101 1111 = XX0X XXXX
Const ON_Bit6 = $40;           // XXXX XXXX OR 0100 0000 = X1XX XXXX
Const OFF_Bit6 = $BF;         // XXXX XXXX AND 1011 1111 = X0XX XXXX
Const ON_Bit7 = $80;           // XXXX XXXX OR 1000 0000 = 1XXX XXXX
Const OFF_Bit7 = $7F;         // XXXX XXXX AND 0111 1111 = 0XXX XXXX

Const OutputOFF = CIMenu;      // Color of Output OFF ("0") Status
Const OutputON = CIRed;        // Color of Output ON ("1") Status
Const InputOFF = Clmenu;       // Color of Input OFF ("1") Status
Const InputON = CILime;        // Color of Input ON ("0") Status
{$R *.DFM}

Procedure TForm1.SETUP_8255Click(Sender: TObject);
Var SetupData : Byte;

Begin
  BaseAddress := IO_BaseAddress;           // Set I/O Base Address

  // Initial Reset and 8255 Bus Cycle //
  SetupData := Inp32(BaseAddress+PIB);      // Read PIB Reset Port
  SetupData := SetupData AND OFF_BIT0;     // Bit0 = EXTRST# = "0" (Reset:RES#)
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
SetupData := SetupData OR ON_BIT5;           // Bit5:4 = 11 = PIB Cycle Slowest
SetupData := SetupData OR ON_BIT4;
Out32(BaseAddress+PIB,SetupData);           // Active RES# & Relay

// Initial Data (AUX) For CS# and Relay //
SetupData := Inp32(BaseAddress+AUXD);       // Read Aux Data Port
SetupData := SetupData AND OFF_BIT0;       // Bit0 = Aux0 = "0" (Enable CS)
SetupData := SetupData OR ON_BIT4;        // Bit4 = Aux4 = "1" (Relay OFF)
Out32(BaseAddress+AUXD,SetupData);        // Active Chips Select & Relay
LEDRelay.Brush.Color := OutputOFF;        // LED Relay = OFF

// Initial Direction (AUX) For CS# and Relay //
SetupData := Inp32(BaseAddress+AUXC);       // Read Aux Port Direction
SetupData := SetupData OR ON_BIT4;        // Aux4 = "1" = Output
SetupData := SetupData OR ON_BIT0;        // Aux0 = "1" = Output
Out32(BaseAddress+AUXC,SetupData);        // Setup Aux Direction

// Initial 8255#1 = All Input Port //
Setup_8255.Enabled := False;              // Disable Setup After Setup
RELAY.Enabled := True;                    // Enable Relay Control
Out32(BaseAddress+PCC1,$9B);              // Write Control Port 8255#1
Timer1.Enabled := True;                   // Start Timer For Auto Read
End;

Procedure TForm1.RELAYClick(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+AuxD);    // Read Output Latch AUX
  CheckData := InOutData AND ON_BIT4;     // Check Last AUX4 Status
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
If CheckData = $10 Then
Begin
    InOutData := InOutData AND OFF_BIT4;           // AUX4 = "0" (1110 1111)
    LEDRelay.Brush.Color := OutputON;             // LED Relay = ON
End
Else
Begin
    InOutData := InOutData OR ON_BIT4;           // AUX4 = "1" (0001 0000)
    LEDRelay.Brush.Color := OutputOFF;           // LED Relay = OFF
End;
Out32(BaseAddress+AuxD,InOutData);               // Update AuxD
End;

Procedure TForm1.Timer1Timer(Sender: TObject);
Var  InputData : Byte;
     CheckData : Byte;
Begin
    InputData := Inp32(BaseAddress+PA1);          // Read Input/Output Latch of PA-8255#1

    CheckData := InputData AND ON_BIT0;          // Check PA0 Status
    If CheckData = ON_BIT0 Then
        LEDPA01.Brush.Color := InputOFF         // PA0 = ("1")
    Else LEDPA01.Brush.Color := InputON;        // PA0 = ("0")

    CheckData := InputData AND ON_BIT1;          // Check PA1 Status
    If CheckData = ON_BIT1 Then
        LEDPA11.Brush.Color := InputOFF         // PA1 = ("1")
    Else LEDPA11.Brush.Color := InputON;        // PA1 = ("0")

    CheckData := InputData AND ON_BIT2;          // Check PA2 Status
    If CheckData = ON_BIT2 Then
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
LEDPA21.Brush.Color := InputOFF           // PA2 = ("1")
Else LEDPA21.Brush.Color := InputON;      // PA2 = ("0")

CheckData := InputData AND ON_BIT3;      // Check PA3 Status
If CheckData = ON_BIT3 Then
    LEDPA31.Brush.Color := InputOFF       // PA3 = ("1")
Else LEDPA31.Brush.Color := InputON;      // PA3 = ("0")

CheckData := InputData AND ON_BIT4;      // Check PA4 Status
If CheckData = ON_BIT4 Then
    LEDPA41.Brush.Color := InputOFF       // PA4 = ("1")
Else LEDPA41.Brush.Color := InputON;      // PA4 = ("0")

CheckData := InputData AND ON_BIT5;      // Check PA5 Status
If CheckData = ON_BIT5 Then
    LEDPA51.Brush.Color := InputOFF       // PA5 = ("1")
Else LEDPA51.Brush.Color := InputON;      // PA5 = ("0")

CheckData := InputData AND ON_BIT6;      // Check PA6 Status
If CheckData = ON_BIT6 Then
    LEDPA61.Brush.Color := InputOFF       // PA6 = ("1")
Else LEDPA61.Brush.Color := InputON;      // PA6 = ("0")

CheckData := InputData AND ON_BIT7;      // Check PA7 Status
If CheckData = ON_BIT7 Then
    LEDPA71.Brush.Color := InputOFF       // PA7 = ("1")
Else LEDPA71.Brush.Color := InputON;      // PA7 = ("0")
End;

End.
```

อธิบายการทำงานของคำสั่งในโปรแกรม ตัวอย่างที่ 1.2

สำหรับการทำงานของตัวอย่างโปรแกรมนี้ โค้ดของโปรแกรมส่วนมากจะยังคงเหมือนและคล้ายกันกับตัวอย่างโปรแกรมที่ 1.1 ทุกประการ โดยในส่วนของโปรแกรมที่ใช้ควบคุมการทำงานของ Relay จะเหมือนเดิมไม่มีการเปลี่ยนแปลง สำหรับส่วนของโปรแกรมที่ใช้ในการสั่ง Setup นั้นจะมีความแตกต่างกับตัวอย่างที่ 1.1 เฉพาะส่วนท้าย คือส่วนที่สั่ง Initial การทำงานของ 8255 โดยจะเปลี่ยนค่า Control Word จากรหัส 80H ซึ่งเป็นรหัสที่ใช้ควบคุมการทำงานของ 8255 ให้ทำหน้าที่เป็น Output ทั้ง 3 พอร์ต มาเป็นรหัส 9BH ซึ่งเป็นรหัสที่ใช้ควบคุมให้ 8255 ทำหน้าที่เป็น Input ทั้ง 3 พอร์ตแทนเท่านั้นเอง

โดยในส่วนท้ายของการ Setup นั้นจะมีการกำหนดค่า Time-Out ของ Timer1 (Timer1.Interval) ให้มีค่าเท่ากับ 100 ซึ่งหมายถึง 100 millisecond หรือ 100/1000 ของวินาที จากนั้นจึงสั่งเปิดการทำงานของ Timer1 ซึ่งในตัวอย่างนี้เราจะใช้ Timer1 เป็นตัวควบคุมการสั่งอ่านค่า Input จาก Port-PA ของ 8255 ตัวที่ 1 โดยที่โปรแกรมจะทำการอ่านค่า Input ทุกๆช่วงเวลา Time-Out ของ Timer1 หรือ ทุกๆ 100 millisecond นั้นเอง ซึ่งการทำงานของ Timer1 นั้นจะทำหน้าที่นับเวลาถอยหลังจากค่าที่เรากำหนดไว้ใน Interval โดยค่าที่กำหนดไว้จะถูกลดค่าลงครั้งละ 1 ค่า ในทุกๆ 1 millisecond ซึ่งเมื่อค่าการนับถูกลดลงจนเป็นศูนย์แล้วก็จะทำให้โปรแกรมกระโดดเข้าไปทำงานยังโปรแกรมย่อย Procedure TForm1.Timer1Timer(); ในทันที ซึ่งเราจะเข้าไปเขียนคำสั่งในโปรแกรมย่อยนี้เพื่อสั่งให้โปรแกรมทำการอ่านค่า Input จาก Port-PA มาแสดงผลด้วยสีของ LED บนฟอร์มของโปรแกรม

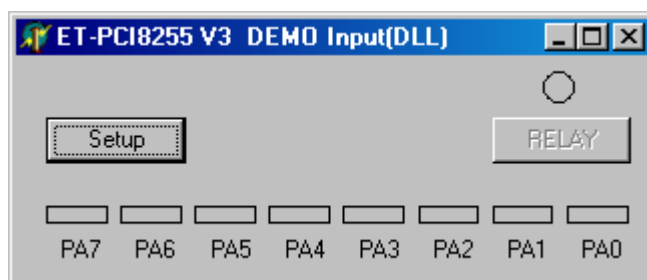
ดังนั้นหลังจากสั่ง Setup การทำงานของโปรแกรมนี้แล้วจะทำให้ Timer1 เริ่มต้นทำงานทันที ซึ่งเมื่อค่าสถานะทาง Input ที่เชื่อมต่อมายัง Port-PA เปลี่ยนไปอย่างไรก็จะทำให้การแสดงผลการทำงานของ Input เปลี่ยนตามไปในทันทีโดยอัตโนมัติ โดยไม่ต้องมีการคลิกเมาส์เพื่อสั่งอ่านค่า Input เองแต่อย่างใด

สำหรับการทำงานของโปรแกรมในส่วนของ Timer1 นั้น ให้ดูจากโค้ดของโปรแกรมในส่วนของโปรแกรมย่อย Procedure TForm1.Timer1Timer(); ซึ่งเป็นส่วนของโปรแกรมย่อยที่ถูกเรียกใช้งานทุกๆครั้งที่ Timer นับจนค่าเป็นศูนย์ โดยการทำงานของโปรแกรมจะเริ่มต้นด้วยการสั่งอ่านค่า Input จาก Port-PA เข้ามาเก็บไว้ในตัวแปร InOutData จากนั้นจึงทำการส่งตรวจสอบค่าข้อมูลที่อ่านได้นี้ทีละบิต โดยใช้ตัวแปร CheckData เป็นตัวทดสอบ ทั้งนี้ก็เพื่อให้ค่าของตัวแปร InOutData ซึ่งใช้เก็บค่าสถานะปัจจุบันของ Input จาก Port-PA มีค่าเหมือนเดิมไม่เปลี่ยนแปลง โดยการตรวจสอบค่า Input ของ Port-PA นั้นจะเริ่มจาก บิต 0 ไปจนถึง บิต 7 โดยในแต่ละบิตนั้น เมื่อตรวจสอบว่าข้อมูลในบิตใดมีค่าเป็น "1" ก็แสดงว่า Input ของบิตนั้นอยู่ในสถานะของการ OFF อยู่ ก็จะสั่งให้ LED ที่ใช้แสดงสถานะของ Input บิตนั้นๆ มีสีเป็นสีพื้นของฟอร์มเพื่อแสดงว่า Input OFF แต่ถ้าพบว่า ค่าข้อมูลของ บิตใดมีค่าเป็น "0" ก็แสดงว่า สถานะ Input ของบิตนั้นๆอยู่ในสถานะ ON ก็จะสั่งให้เปลี่ยนสีของ LED ที่ใช้แสดงค่า Input ของบิตนั้นๆมีค่าเป็นสีเขียวแทน โดยโปรแกรมจะวนรอบทำงานซ้ำอยู่อย่างนี้ในทุกๆ 100 millisecond

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

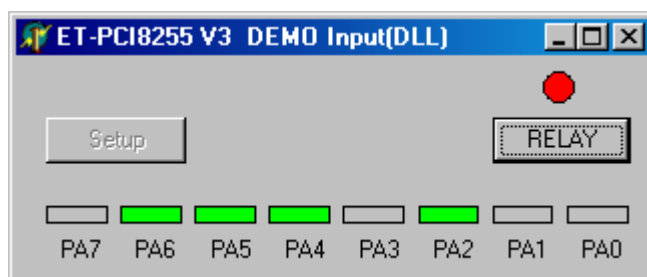
การทดสอบการทำงานของโปรแกรม ตัวอย่างที่ 1.2

1. ก่อนทำการทดสอบให้ต่อบอร์ด ET-TEST I/O เข้ากับพอร์ต PA ของ 8255#1 เพื่อใช้แสดงการทำงานของ Output จาก 8255 แต่ถ้าไม่มีก็อาจไม่ต้องต่อก็ได้
2. สั่ง Run โปรแกรมที่เขียนเสร็จแล้วโดยกด F5 ซึ่งในครั้งแรกจะเห็นว่าปุ่มควบคุมการทำงานของ RELAY จะถูกปิดการทำงานไว้ จะมีเฉพาะปุ่ม SETUP เท่านั้นที่ถูกเปิดการทำงานอยู่ ทั้งนี้ก็เพื่อเป็นการบังคับให้ผู้ใช้ต้องสั่ง Setup การทำงานของการ์ดก่อนเป็นอันดับแรกดังรูป



แสดงรูป ผลการ Run โปรแกรมตัวอย่างในครั้งแรก

3. เริ่มต้นทดสอบการทำงานของโปรแกรมโดยการ คลิกเมาส์ที่ปุ่ม SETUP เพื่อสั่ง Setup การทำงานของชิพ TIGER320 และ 8255 ตัวที่ 1 ซึ่งจะทำให้ปุ่ม SETUP ถูกปิดการทำงาน ขณะเดียวกันปุ่ม ควบคุมการทำงานของ RELAY จะถูกเปิดการทำงานขึ้นมาทันที ให้ทดสอบคลิกเมาส์ที่ปุ่ม RELAY สลับไปมา ซึ่งจะต้องเห็นการทำงานของ Relay สลับการทำงานระหว่าง ON/OFF ตามการคลิกเมาส์แต่ละครั้ง โดยจะสังเกตเห็น LED แสดงสถานะการทำงานของ Relay สลับสีไปมาระหว่างสีแดง กับสีพื้นของเมนู ตามจังหวะการ ON/OFF ด้วยทุกครั้ง และถ้าต้องการทดสอบการทำงานของ Input จาก Port-PA ก็ให้ทดลองปรับตำแหน่งของ Dip-Switch ที่ต่อไว้กับ Port-PA ดู ซึ่งจะสังเกตเห็นหลอดแสดงผล LED ของ Input แต่ละบิตเปลี่ยนแปลงไปตามการทำงานของ Input ด้วยทันที ดังรูป



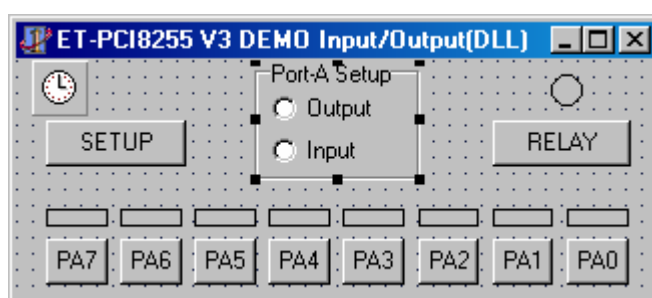
แสดงรูป ผลการทดสอบโปรแกรมของ ตัวอย่างที่ 1.2

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ตัวอย่างที่ 1.3 DEMO Input/Output (เรียกใช้ Library ไฟล์จาก InpOut32.DLL)

สำหรับโปรแกรม ตัวอย่างที่1.3 จะเป็นการแสดงให้เห็นการประยุกต์ใช้งาน 8255 ให้ทำหน้าที่เป็น Input / Output แบบอัตโนมัติ โดยสามารถเลือกได้ว่าจะให้พอร์ตของ 8255 ทำหน้าที่เป็น Input หรือ Output สำหรับขั้นตอนการเขียนโปรแกรมของการ์ด ET-PCI8255 V3 มีรายละเอียดดังนี้

1. เปิด Project ใหม่โดยไปที่ File > New Application จะได้ฟอร์มเปล่าขึ้นมา 1 ฟอร์ม
2. เมื่อฟอร์มขึ้นมาให้นำคอนโทรลชนิดต่างๆ มาจัดวางดังรูป



รูปแสดง การจัดวาง Component บน Form ของโปรแกรม ตัวอย่างที่1.3

โดยในตัวอย่างนี้จะใช้ Component ทั้งหมด 4 ชนิดด้วยกัน คือ Command Button, Shape ,Radio Group และ Timer ซึ่ง Command Button จะมีทั้งหมด 10 ชุด ใช้สำหรับ Setup การทำงานของการ์ด ET-PCI8255 V3 จำนวน 1 ชุด โดยเมื่อกดปุ่ม SETUP จะเป็นการตั้ง Initial การทำงานของการ์ด ET-PCI8255 V3

สำหรับ Shape นั้นจะใช้สำหรับทำหน้าที่เป็นเสมือน หลอด LED หรือไฟแสดง สถานะ การทำงานของ Relay และ Input / Output ของ PA0..PA7 โดยให้ LED ของ Relay มีลักษณะเป็นทรงกลม และให้ LED การแสดงผลของ Input / Output 8255 มีลักษณะเป็นสี่เหลี่ยม ส่วนการแสดงผลของ LED ในกรณีของ LED Relay ถ้า ON จะให้สถานะเป็นสีแดง แต่ถ้า Relay OFF จะให้สถานะเป็นสีพื้นของ Form แทน

สำหรับในกรณีของ LED Input / Output นั้นจะให้การทำงานเป็นสีเขียวเมื่อเป็น Input ON และเป็นสีแดงเมื่อเป็น Output ON สลับ กับ สีพื้นของเมนูเมื่อเป็น Input / Output OFF

ส่วน Radio Group นั้นจะใช้สำหรับเลือกกำหนดหน้าที่ของ Port-PA ของ 8255 ว่าจะกำหนดหน้าที่การทำงานเป็นแบบ Input หรือ Output

สำหรับ Timer นั้นจะถูกใช้สำหรับทำหน้าที่เป็นตัวนับเวลาสำหรับการอ่านค่าสถานะ Input จาก Port-PA มาแสดงผลการทำงานให้เห็น ที่หลอดแสดงผล LED บนฟอร์มของโปรแกรม เพื่อให้การทำงานของโปรแกรมเป็นไปอย่างอัตโนมัติ ไม่ต้องเสียเวลามาคอยสั่งอ่านค่า Input ด้วยการกดปุ่ม Button เหมือนการสั่งงาน Output

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

3 ทำการกำหนดค่า Properties ให้กับคอนโทรลต่างๆ ดังนี้

ชื่อ	คอนโทรล	พรีอเพอร์ตี้	ค่าที่กำหนด
Form1	Form	Name	Form1
		Border Style	bsSizeable
		Caption	ET-PCI8255 V3 DEMO Input/Output(DLL)
SETUP	Command Button	Name	SETUP
		Enable	True
		Caption	SETUP
PA0 . . PA7	Command Button	Name	PA01,PA11,PA21,...PA71
		Enable	False
		Caption	PA0,PA1,PA2,...PA7
	Shape	Name	LEDPA01,LEDPA11,LEDPA21,...LEDPA71
		Fill Style	Solid
		Shape	Rectangle
RELAY	Command Button	Name	RELAY
		Enable	False
		Caption	RELAY
	Shape	Name	LEDRelay
		Fill Style	Solid
		Shape	Circle
Timer1	Timer	Name	Timer1
		Enable	False
		Interval	100
Option Button1		Name	SelectPA1_Input
		Caption	Input
		Enabled	False
Option Button2		Name	SelectPA1_Output
		Caption	Output
		Enabled	False

4. เขียนโค้ดคำสั่งเพื่อควบคุมการทำงานของโปรแกรกดังต่อไปนี้

```
{ Demo3 Input/Output Demo Access Input/Output Library "INPOUT32.DLL" }  
Unit Unit1;  
Interface  
Uses Windows,Messages,SysUtils,Classes,Graphics,Controls,Forms,Dialogs,StdCtrls,ExtCtrls;  
  
Type TForm1 = class(TForm)  
    SETUP_8255: TButton;  
    RELAY: TButton;  
    PA01: TButton;  
    PA11: TButton;  
    PA21: TButton;  
    PA31: TButton;  
    PA41: TButton;  
    PA51: TButton;  
    PA61: TButton;  
    PA71: TButton;  
    LEDRELAY: TShape;  
    LEDPA01: TShape;  
    LEDPA11: TShape;  
    LEDPA21: TShape;  
    LEDPA31: TShape;  
    LEDPA41: TShape;  
    LEDPA51: TShape;  
    LEDPA61: TShape;  
    LEDPA71: TShape;  
    Control_PA8255_1: TRadioGroup;  
    Timer1: TTimer;  
    Procedure SETUP_8255Click(Sender: TObject);  
    Procedure RELAYClick(Sender: TObject);  
end;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure PA01Click(Sender: TObject);  
Procedure PA11Click(Sender: TObject);  
Procedure PA21Click(Sender: TObject);  
Procedure PA31Click(Sender: TObject);  
Procedure PA41Click(Sender: TObject);  
Procedure PA51Click(Sender: TObject);  
Procedure PA61Click(Sender: TObject);  
Procedure PA71Click(Sender: TObject);  
Procedure Control_PA8255_1Click(Sender: TObject);  
Procedure Timer1Timer(Sender: TObject);
```

Private

```
{ Private declarations }
```

Public

```
{ Public declarations }
```

End;

```
Var Form1 : TForm1;
```

```
BaseAddress : Word;
```

```
Procedure Out32(Port:Word; Data:Byte); Stdcall;External'InpOut32.DLL';
```

```
Function Inp32(Port:Word):Byte; Stdcall; External 'InpOut32.DLL';
```

Implementation

```
Const IO_BaseAddress = $E800; // I/O Base Address
```

```
Const PIB = $00; // Reset & PIB Cycle
```

```
Const AUXC = $02; // Aux Direction Port
```

```
Const AUXD = $03; // Aux Data Port
```

```
Const PA1 = $C0; // Port-A 8255#1
```

```
Const PB1 = $C4; // Port-B 8255#1
```

```
Const PC1 = $C8; // Port-C 8255#1
```

```
Const PCC1 = $CC; // Port Control 8255#1
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```

Const PA2 = $D0;           // Port-A 8255#2
Const PB2 = $D4;           // Port-B 8255#2
Const PC2 = $D8;           // Port-C 8255#2
Const PCC2 = $DC;          // Port Control 8255#2
Const PA3 = $E0;           // Port-A 8255#3
Const PB3 = $E4;           // Port-B 8255#3
Const PC3 = $E8;           // Port-C 8255#3
Const PCC3 = $EC;          // Port Control 8255#3
Const ON_Bit0 = $01;        // XXXX XXXX OR 0000 0001 = XXXX XXX1
Const OFF_Bit0 = $FE;       // XXXX XXXX AND 1111 1110 = XXXX XXX0
Const ON_Bit1 = $02;        // XXXX XXXX OR 0000 0010 = XXXX XX1X
Const OFF_Bit1 = $FD;       // XXXX XXXX AND 1111 1101 = XXXX XX0X
Const ON_Bit2 = $04;        // XXXX XXXX OR 0000 0100 = XXXX X1XX
Const OFF_Bit2 = $FB;       // XXXX XXXX AND 1111 1011 = XXXX X0XX
Const ON_Bit3 = $08;        // XXXX XXXX OR 0000 1000 = XXXX 1XXX
Const OFF_Bit3 = $F7;       // XXXX XXXX AND 1111 0111 = XXXX 0XXX
Const ON_Bit4 = $10;        // XXXX XXXX OR 0001 0000 = XXX1 XXXX
Const OFF_Bit4 = $EF;       // XXXX XXXX AND 1110 1111 = XXX0 XXXX
Const ON_Bit5 = $20;        // XXXX XXXX OR 0010 0000 = XX1X XXXX
Const OFF_Bit5 = $DF;       // XXXX XXXX AND 1101 1111 = XX0X XXXX
Const ON_Bit6 = $40;        // XXXX XXXX OR 0100 0000 = X1XX XXXX
Const OFF_Bit6 = $BF;       // XXXX XXXX AND 1011 1111 = X0XX XXXX
Const ON_Bit7 = $80;        // XXXX XXXX OR 1000 0000 = 1XXX XXXX
Const OFF_Bit7 = $7F;       // XXXX XXXX AND 0111 1111 = 0XXX XXXX

Const OutputOFF = CIMenu;   // Color of Output OFF ("0") Status
Const OutputON = CIRed;     // Color of Output ON ("1") Status
Const InputOFF = Clmenu;    // Color of Input OFF ("1") Status
Const InputON = CILime;     // Color of Input ON ("0") Status

{$R *.DFM}

```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure TForm1.SETUP_8255Click(Sender: TObject);
Var  SetupData : Byte;

Begin
  BaseAddress := IO_BaseAddress;           // Set I/O Base Address

  // Initial Reset and 8255 Bus Cycle //
  SetupData := Inp32(BaseAddress+PIB);     // Read PIB Reset Port
  SetupData := SetupData AND OFF_BIT0;    // Bit0 = EXTRST# = "0" (Reset:RES#)
  SetupData := SetupData OR  ON_BIT5;     // Bit5:4 = 11 = PIB Cycle Slowest
  SetupData := SetupData OR  ON_BIT4;
  Out32(BaseAddress+PIB,SetupData);       // Active RES# & Relay

  // Initial Data (AUX) For CS# and Relay //
  SetupData := Inp32(BaseAddress+AUXD);    // Read Aux Data Port
  SetupData := SetupData AND OFF_BIT0;    // Bit0 = Aux0 = "0" (Enable CS)
  SetupData := SetupData OR  ON_BIT4;     // Bit4 = Aux4 = "1" (Relay OFF)
  Out32(BaseAddress+AUXD,SetupData);      // Active Chips Select & Relay
  LEDRelay.Brush.Color := OutputOFF;     // LED Relay = OFF

  // Initial Direction (AUX) For CS# and Relay //
  SetupData := Inp32(BaseAddress+AUXC);    // Read Aux Port Direction
  SetupData := SetupData OR ON_BIT4;     // Aux4 = "1" = Output
  SetupData := SetupData OR ON_BIT0;     // Aux0 = "1" = Output
  Out32(BaseAddress+AUXC,SetupData);     // Setup Aux Direction

  Setup_8255.Enabled := False;           // Disable Setup After Setup
  RELAY.Enabled := True;                 // Enable Relay Control
  Control_PA8255_1.Enabled := True;     // Enable Select Control Port-PA 8255#1
  Control_PA8255_1.ItemIndex := 0;      // Default Port-PA = Output
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Timer1.Interval := 100; // Setup Input Scan Time = 100mS
Timer1.Enabled := True; // Start Auto Scan Input
End;

Procedure TForm1.RelayClick(Sender: TObject);
Var InOutData : Byte;
    CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+AuxD); // Read Output Latch AUX
    CheckData := InOutData AND ON_BIT4; // Check Last AUX4 Status
    If CheckData = $10 Then
    Begin
        InOutData := InOutData AND OFF_BIT4; // AUX4 = "0" (1110 1111)
        LEDRelay.Brush.Color := OutputON; // LED RELay = ON
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT4; // AUX4 = "1" (0001 0000)
        LEDRelay.Brush.Color := OutputOFF; // LED Relay = OFF
    End;
    Out32(BaseAddress+AuxD,InOutData); // Update AuxD
End;

Procedure TForm1.PA01Click(Sender: TObject);
Var InOutData : Byte;
    CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+PA1); // Read Output Latch Port-A
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
CheckData := InOutData AND ON_BIT0;           // Check Last PA0 Status
If CheckData = $01 Then
Begin
    InOutData := InOutData AND OFF_BIT0;       // PA0 = "0" (1111 1110)
    LEDPA01.Brush.Color := OutputOFF;         // LED PA0 of 8255#1 = OFF
End
Else
Begin
    InOutData := InOutData OR ON_BIT0;         // PA0 = "1" (0000 0001)
    LEDPA01.Brush.Color := OutputON;          // LED PA0 of 8255#1 = ON
End;
Out32(BaseAddress+PA1,InOutData);             // Update Port-A
End;

Procedure TForm1.PA11Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+PA1);       // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT1;       // Check Last PA1 Status
    If CheckData = $02 Then
    Begin
        InOutData := InOutData AND OFF_BIT1;  // PA1 = "0" (1111 1101)
        LEDPA11.Brush.Color := OutputOFF;    // LED PA1 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT1;    // PA1 = "1" (0000 0010)
        LEDPA11.Brush.Color := OutputON;     // LED PA1 of 8255#1 = ON
    End;
End;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Out32(BaseAddress+PA1,InOutData);           // Update Port-A
End;

Procedure TForm1.PA21Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);       // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT2;       // Check Last PA2 Status
  If CheckData = $04 Then
  Begin
    InOutData := InOutData AND OFF_BIT2;    // PA2 = "0" (1111 1011)
    LEDPA21.Brush.Color := OutputOFF;      // LED PA2 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT2;      // PA2 = "1" (0000 0100)
    LEDPA21.Brush.Color := OutputON;       // LED PA2 of 8255#1 = ON
  End;
  Out32(BaseAddress+PA1,InOutData);         // Update Port-A
End;

Procedure TForm1.PA31Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);       // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT3;       // Check Last PA3 Status
  If CheckData = $08 Then
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Begin
  InOutData := InOutData AND OFF_BIT3;           // PA3 = "0" (1111 0111)
  LEDPA31.Brush.Color := OutputOFF;             // LED PA3 of 8255#1 = OFF
End
Else
Begin
  InOutData := InOutData OR ON_BIT3;           // PA3 = "1" (0000 1000)
  LEDPA31.Brush.Color := OutputON;             // LED PA3 of 8255#1 = ON
End;
Out32(BaseAddress+PA1,InOutData);               // Update Port-A
End;

Procedure TForm1.PA41Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);          // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT4;          // Check Last PA4 Status
  If CheckData = $10 Then
  Begin
    InOutData := InOutData AND OFF_BIT4;       // PA4 = "0" (1110 1111)
    LEDPA41.Brush.Color := OutputOFF;         // LED PA4 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT4;         // PA4 = "1" (0001 0000)
    LEDPA41.Brush.Color := OutputON;          // LED PA4 of 8255#1 = ON
  End;
  Out32(BaseAddress+PA1,InOutData);            // Update Port-A
End;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure TForm1.PA51Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT5;          // Check Last PA5 Status
  If CheckData = $20 Then
  Begin
    InOutData := InOutData AND OFF_BIT5;       // PA5 = "0" (1101 1111)
    LEDPA51.Brush.Color := OutputOFF;         // LED PA5 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT5;         // PA5 = "1" (0010 0000)
    LEDPA51.Brush.Color := OutputON;          // LED PA5 of 8255#1 = ON
  End;
  Out32(BaseAddress+PA1,InOutData);           // Update Port-A
End;

Procedure TForm1.PA61Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := Inp32(BaseAddress+PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT6;          // Check Last PA6 Status
  If CheckData = $40 Then
  Begin
    InOutData := InOutData AND OFF_BIT6;       // PA6 = "0" (1011 1111)
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
LEDPA61.Brush.Color := OutputOFF;           // LED PA6 of 8255#1 = OFF
End
Else
Begin
    InOutData := InOutData OR ON_BIT6;       // PA6 = "1" (0100 0000)
    LEDPA61.Brush.Color := OutputON;         // LED PA6 of 8255#1 = ON
End;
Out32(BaseAddress+PA1,InOutData);           // Update Port-A
End;

Procedure TForm1.PA71Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
    InOutData := Inp32(BaseAddress+PA1);     // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT7;     // Check Last PA7 Status
    If CheckData = $80 Then
    Begin
        InOutData := InOutData AND OFF_BIT7; // PA7 = "0" (0111 1111)
        LEDPA71.Brush.Color := OutputOFF;    // LED PA7 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT7;   // PA7 = "1" (1000 0000)
        LEDPA71.Brush.Color := OutputON;    // LED PA7 of 8255#1 = ON
    End;
    Out32(BaseAddress+PA1,InOutData);        // Update Port-A
End;

Procedure TForm1.Control_PA8255_1Click(Sender: TObject);
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Var ControlWord : Byte;
    InOutData   : Byte;
Begin
    // Control Word Bit7 = 1 = Active 8255 Mode0 //
    ControlWord := $80;                               //Default Control Word 8255 = Mode0 //

    // Port PA(PA7..0) = Bit-D4 in Control Word
    // If Control Word Bit4 = "0" = PA is Output
    // If Control Word Bit4 = "1" = PA is Input
    If Control_PA8255_1.ItemIndex = 0 Then           // If Output Select
    Begin
        ControlWord := ControlWord AND OFF_BIT4;    // 1110 1111 => D4 = "0" = PA is Output
        PA01.Enabled := True;                       // Enable PA Output Control
        PA11.Enabled := True;
        PA21.Enabled := True;
        PA31.Enabled := True;
        PA41.Enabled := True;
        PA51.Enabled := True;
        PA61.Enabled := True;
        PA71.Enabled := True;
    End
    Else                                             // If Input Select
    Begin
        ControlWord := ControlWord OR ON_BIT4;      // 0001 0000 => D4 = "1" = PA is Input
        PA01.Enabled := False;                      // Disable PA Output Control
        PA11.Enabled := False;
        PA21.Enabled := False;
        PA31.Enabled := False;
        PA41.Enabled := False;
        PA51.Enabled := False;
        PA61.Enabled := False;
    End
End
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
PA71.Enabled := False;
End;

// Start Initial 8255#1 = Control Word //
Out32(BaseAddress+PCC1,ControlWord);           // Write Control Port 8255#1
End;

Procedure TForm1.Timer1Timer(Sender: TObject);
Var  InputData : Byte;
     CheckData : Byte;
Begin
  InputData := Inp32(BaseAddress+PA1);           // Read Input/Output Latch of PA-8255#1
  If Control_PA8255_1.ItemIndex = 1 Then       // If PA = Input Mode
  Begin
    CheckData := InputData AND ON_BIT0;        // Check PA0 Status
    If CheckData = ON_BIT0 Then
      LEDPA01.Brush.Color := InputOFF          // PA0 = ("1")
    Else LEDPA01.Brush.Color := InputON;       // PA0 = ("0")

    CheckData := InputData AND ON_BIT1;        // Check PA1 Status
    If CheckData = ON_BIT1 Then
      LEDPA11.Brush.Color := InputOFF          // PA1 = ("1")
    Else LEDPA11.Brush.Color := InputON;       // PA1 = ("0")

    CheckData := InputData AND ON_BIT2;        // Check PA2 Status
    If CheckData = ON_BIT2 Then
      LEDPA21.Brush.Color := InputOFF          // PA2 = ("1")
    Else LEDPA21.Brush.Color := InputON;       // PA2 = ("0")

    CheckData := InputData AND ON_BIT3;        // Check PA3 Status
    If CheckData = ON_BIT3 Then
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
LEDPA31.Brush.Color := InputOFF           // PA3 = ("1")
Else LEDPA31.Brush.Color := InputON;      // PA3 = ("0")

CheckData := InputData AND ON_BIT4;      // Check PA4 Status
If CheckData = ON_BIT4 Then
    LEDPA41.Brush.Color := InputOFF       // PA4 = ("1")
Else LEDPA41.Brush.Color := InputON;     // PA4 = ("0")

CheckData := InputData AND ON_BIT5;      // Check PA5 Status
If CheckData = ON_BIT5 Then
    LEDPA51.Brush.Color := InputOFF       // PA5 = ("1")
Else LEDPA51.Brush.Color := InputON;     // PA5 = ("0")

CheckData := InputData AND ON_BIT6;      // Check PA6 Status
If CheckData = ON_BIT6 Then
    LEDPA61.Brush.Color := InputOFF       // PA6 = ("1")
Else LEDPA61.Brush.Color := InputON;     // PA6 = ("0")

CheckData := InputData AND ON_BIT7;      // Check PA7 Status
If CheckData = ON_BIT7 Then
    LEDPA71.Brush.Color := InputOFF       // PA7 = ("1")
Else LEDPA71.Brush.Color := InputON;     // PA7 = ("0")
End
Else                                     // If PA = Output Mode
Begin
    CheckData := InputData AND ON_BIT0;   // Check PA0 Status
    If CheckData = ON_BIT0 Then
        LEDPA01.Brush.Color := OutputON   // PA0 = ("1")
    Else LEDPA01.Brush.Color := OutputOFF; // PA0 = ("0")

    CheckData := InputData AND ON_BIT1;   // Check PA1 Status
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
If CheckData = ON_BIT1 Then
    LEDPA11.Brush.Color := OutputON           // PA1 = ("1")
Else LEDPA11.Brush.Color := OutputOFF;       // PA1 = ("0")

CheckData := InputData AND ON_BIT2;         // Check PA2 Status
If CheckData = ON_BIT2 Then
    LEDPA21.Brush.Color := OutputON           // PA2 = ("1")
Else LEDPA21.Brush.Color := OutputOFF;       // PA2 = ("0")

CheckData := InputData AND ON_BIT3;         // Check PA3 Status
If CheckData = ON_BIT3 Then
    LEDPA31.Brush.Color := OutputON           // PA3 = ("1")
Else LEDPA31.Brush.Color := OutputOFF;       // PA3 = ("0")

CheckData := InputData AND ON_BIT4;         // Check PA4 Status
If CheckData = ON_BIT4 Then
    LEDPA41.Brush.Color := OutputON           // PA4 = ("1")
Else LEDPA41.Brush.Color := OutputOFF;       // PA4 = ("0")

CheckData := InputData AND ON_BIT5;         // Check PA5 Status
If CheckData = ON_BIT5 Then
    LEDPA51.Brush.Color := OutputON           // PA5 = ("1")
Else LEDPA51.Brush.Color := OutputOFF;       // PA5 = ("0")

CheckData := InputData AND ON_BIT6;         // Check PA6 Status
If CheckData = ON_BIT6 Then
    LEDPA61.Brush.Color := OutputON           // PA6 = ("1")
Else LEDPA61.Brush.Color := OutputOFF;       // PA6 = ("0")

CheckData := InputData AND ON_BIT7;         // Check PA7 Status
If CheckData = ON_BIT7 Then
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
LEDPA71.Brush.Color := OutputON           // PA7 = ("1")
Else LEDPA71.Brush.Color := OutputOFF;     // PA7 = ("0")
End;
End;
End.
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

อธิบายการทำงานของคำสั่งในโปรแกรม ตัวอย่างที่ 1.3

สำหรับการทำงานของตัวอย่างโปรแกรมนี้ ส่วนมากแล้วการทำงานของโปรแกรมโดยรวมจะมีลักษณะการทำงานเหมือนกับ 2 ตัวอย่างที่ผ่านมา ซึ่งอาจกล่าวได้ว่าเป็นการนำเอา ตัวอย่างที่ 1.1 และ ตัวอย่างที่ 1.2 มารวมเข้าด้วยกัน โดยมีการเพิ่มเติม Radio Group เข้ามาเพื่อใช้เป็นตัวเลือกหน้าที่การทำงานของ Port-PA ของ 8255 เท่านั้นเอง โดยจะอาศัยหลักการทำงานที่ว่า ถ้า Radio Group ถูกเลือกไว้ที่ Input ก็จะสั่ง Initial การทำงานของ พอร์ต PA ของ 8255 ให้ทำหน้าที่เป็น Input โดยในกรณีนี้จะทำการสั่งปิดการทำงานของ Button สำหรับใช้สั่งงาน Output ของ PA0 ถึง PA7 ให้ไม่สามารถใช้งานได้ แต่ถ้า Radio Group ถูกเลือกไว้ที่ Output ก็จะสั่ง Initial การทำงานของ 8255 เพื่อสั่งให้พอร์ต PA ทำหน้าที่เป็น Output โดยในกรณีนี้ก็จะสั่งเปิดการทำงานของ Button สำหรับใช้ควบคุมการทำงานของ PA0..PA7 ไว้ด้วย เพื่อให้ผู้ใช้งานสามารถสั่งงาน Output ของ PA0..PA7 ได้ตามปรกติ โดยการทำงานของตัวอย่างโปรแกรมนี้ โปรแกรมย่อยส่วน Setup นั้นจะทำการ Initial เฉพาะการทำงานของชิพ TIGER320 เท่านั้น ส่วนการทำงานของ 8255 จะไม่ได้ทำการสั่ง Initial จากการ Setup ด้วย การ Setup จะทำหน้าที่เพียงการ Initial เพื่อเปิดสัญญาณการรีเซ็ตและเปิดการทำงานของสัญญาณ Chips Select (CS#) เท่านั้น โดยการกำหนดหน้าที่ของ 8255 นั้นจะกระทำในโปรแกรมย่อยของ Procedure TForm1.Control_PA8255_1Click(); เท่านั้น โดยถ้าเลือก Radio Group ไว้ทางด้าน Input โปรแกรมจะสั่ง Initial ให้ 8255 ทำหน้าที่เป็น Input โดยจะสั่งปิดการทำงานของ Button สำหรับควบคุมการทำงานของ Output PA0..PA7 ด้วย แต่ถ้าเลือก Radio Group ไว้ทางด้าน Output โปรแกรมจะทำการสั่ง Initial การทำงานของ 8255 ให้ทำหน้าที่เป็น Output โดยจะสั่งเปิดการทำงานของ Button สำหรับใช้ควบคุมการทำงานของ Output PA0..PA7 ไปด้วย โดยเทคนิควิธีการของการสั่ง Initial หน้าที่ของ 8255 นั้นจะแตกต่างไปจากทั้ง 2 ตัวอย่างที่ผ่านมา เนื่องจากใน 2 ตัวอย่างแรกนั้น จะใช้วิธีการส่งค่า Control Word ด้วยรหัสคงที่ซึ่งกำหนดไว้ล่วงหน้าแล้ว คือ 80H และ 9BH โดยค่า 80H จะเป็นการสั่งให้ทุกพอร์ตของ 8255 เป็น Output ทั้งหมด ส่วนค่า 9BH จะเป็นการสั่งให้ 8255 ทุกพอร์ตเป็น Input ทั้งหมด แต่ในตัวอย่างนี้ จะใช้วิธีการแยก Initial การทำงานของ 8255 ของแต่ละพอร์ตโดยอิสระ โดยจะเริ่มต้นจากการกำหนดค่าเริ่มต้นให้กับ Control Word ด้วยรหัส 80H ซึ่งเป็นการกำหนดให้ 8255 ทำงานแบบพื้นฐานใน โหมด0 จากนั้นจึงตรวจสอบตัวเลือกที่ใช้กำหนดหน้าที่การทำงานของแต่ละพอร์ตว่าพอร์ตใดถูกกำหนดให้ทำหน้าที่เป็น Input หรือ Output บ้าง แล้ว จึงส่งกำหนดค่าข้อมูลของบิตข้อมูลสำหรับใช้ควบคุมการทำงานของแต่ละพอร์ตใน 8255 โดยตรง โดยถ้าค่าบิตควบคุมเป็น “0” จะเป็น Output ถ้าเป็น “1” จะเป็น Input

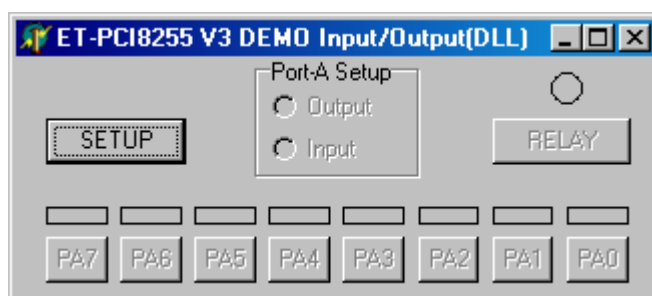
- Port-PA จะใช้ บิต4 สำหรับเลือกหน้าที่การทำงาน
- Port-PB จะใช้ บิต1สำหรับเลือกหน้าที่การทำงาน
- Port-PC ล่าง (PC0..PC3) จะใช้ บิต0 สำหรับเลือกหน้าที่การทำงาน
- Port-PC บน (PC4..PC7) จะใช้ บิต3 สำหรับเลือกหน้าที่การทำงาน

โดยจากตัวอย่างนี้จะแสดงให้เห็นเฉพาะการสั่ง Initial หน้าที่ของ Port-PA เพียงพอร์ตเดียวเท่านั้น

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

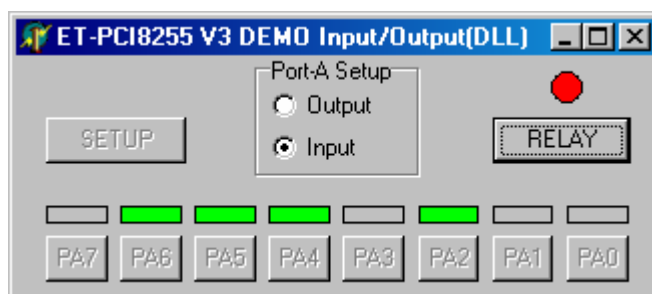
การทดสอบการทำงานของโปรแกรม ตัวอย่างที่ 1.3

1. ก่อนทำการทดสอบให้ต่อบอร์ด ET-TEST I/O เข้ากับพอร์ต PA ของ 8255#1 เพื่อใช้แสดงการทำงานของ Output จาก 8255 แต่ถ้าไม่มีก็อาจไม่ต้องต่อก็ได้
2. สั่ง Run โปรแกรมที่เขียนเสร็จแล้วโดยกด F5 ซึ่งในครั้งแรกจะเห็นว่าปุ่มควบคุมการทำงานของ RELAY จะถูกปิดการทำงานไว้ จะมีเฉพาะปุ่ม SETUP เท่านั้นที่ถูกเปิดการทำงานอยู่ ทั้งนี้ก็เพื่อเป็นการบังคับให้ผู้ใช้ต้องสั่ง Setup การทำงานของการ์ดก่อนเป็นอันดับแรกดังรูป



แสดงรูป ผลการ Run โปรแกรมตัวอย่างในครั้งแรก

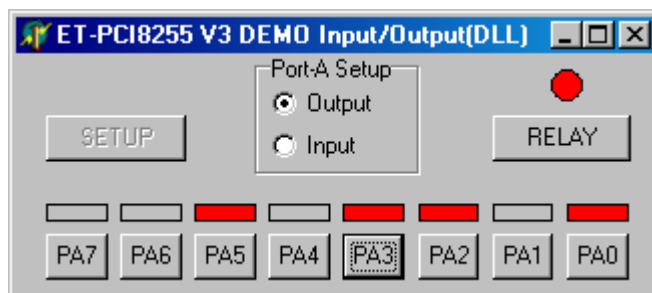
3. เริ่มต้นทดสอบการทำงานของโปรแกรมโดยการ คลิกเมาส์ที่ปุ่ม SETUP เพื่อสั่ง Setup การทำงานของชิพ TIGER320 และ 8255 ตัวที่ 1 ซึ่งจะทำให้ปุ่ม SETUP ถูกปิดการทำงาน ขณะเดียวกันปุ่ม ควบคุมการทำงานของ RELAY จะถูกเปิดการทำงานขึ้นมาทันที ให้ทดสอบคลิกเมาส์ที่ปุ่ม RELAY สลับไปมา ซึ่งจะต้องเห็นการทำงานของ Relay สลับการทำงานระหว่าง ON/OFF ตามการคลิกเมาส์แต่ละครั้ง โดยจะสังเกตเห็น LED แสดงสถานะการทำงานของ Relay สลับสีไปมาระหว่างสีแดง กับสีพื้นของเมนู ตามจังหวะการ ON/OFF ด้วยทุกครั้ง และถ้าต้องการทดสอบการทำงานของ Input จาก Port-PA ก็ให้เลือก Port-A Select ไว้ที่ตำแหน่ง Input และทดลองปรับตำแหน่งของ Dip-Switch ที่ต่อไว้กับ Port-PA ดู ซึ่งจะสังเกตเห็นหลอดแสดงผล LED ของ Input แต่ละบิตเปลี่ยนแปลงไปตามการทำงานของ Input ด้วยทันที ดังรูป



แสดงรูป ผลการทดสอบโปรแกรมของ ตัวอย่างที่ 1.3 เมื่อเลือกเป็น Input

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

และในทำนองเดียวกัน ถ้าต้องการทดสอบการทำงานของ Port-PA แบบ Output Port ก็ให้ทำการคลิกเมาส์เลือกการทำงานของ Port-PA ที่ตัวเลือก Port-A Select ให้เป็น Output ก็จะสังเกตเห็นปุ่ม Button สำหรับใช้ควบคุมการทำงานของ Output PA0..PA7 เปลี่ยนไปอยู่ในสถานะพร้อมทำงาน ซึ่งเราสามารถสั่ง ON/OFF การทำงานของ Output บิต ต่างๆได้ตามต้องการทันที ดังรูป



แสดงรูป ผลการทดสอบ โปรแกรมของ ตัวอย่างที่1.3 เมื่อเลือกเป็น Output

ซึ่งจากตัวอย่างโปรแกรมทั้ง 3 ตัวอย่างที่ผ่านมานั้น ก็คงพอจะทำให้ผู้อ่านเกิดความเข้าใจ ถึงวิธีการเขียนโปรแกรมเพื่อควบคุมการทำงานของ การ์ด ET-PCI8255 V3 กันพอสมควรแล้ว และคงเกิดแนวคิดในการนำเอาการ์ด ET-PCI8255 V3 ไปดัดแปลงหรือประยุกต์ใช้งานในรูปแบบต่างๆได้มากขึ้น

ซึ่งตัวอย่างทั้งหมดที่แสดงให้เห็นมาแล้วนั้น จะแสดงการทำงานของ 8255 เพียงแค่พอร์ตเดียวเท่านั้น คือ Port-PA ของ 8255 ตัวที่ 1 ซึ่งเมื่อผู้อ่านสามารถทำความเข้าใจกับการทำงานของตัวอย่างโปรแกรมได้เป็นอย่างดีแล้ว ก็ย่อมสามารถที่จะนำโปรแกรมตัวอย่างต่างๆเหล่านี้ไปดัดแปลงเพื่อใช้ควบคุมการทำงานของพอร์ต I/O อื่นๆที่เหลือได้โดยไม่ยากนัก เนื่องจากแนวคิดและหลักการเขียนโปรแกรมจะเหมือนกันทั้งหมด จะมีความแตกต่างกันเฉพาะการอ้างตำแหน่งแอดเดรสของพอร์ตเท่านั้นเอง

การเขียนโปรแกรมควบคุมการ์ด ET-PCI8255 V3 ด้วย Library Driver

การเขียนโปรแกรมควบคุมการทำงานของการ์ดด้วยวิธีการนี้จะมีข้อดี ต่อการใช้งานเป็นอย่างมาก คือ การทำงานของโปรแกรมจะมีความอ่อนตัวเป็นอย่างมาก เนื่องจากเราไม่จำเป็นต้องคอยไปตรวจสอบตำแหน่งการทำงานของการ์ดว่ามีค่าเป็นเท่าใด แต่เราสามารถสั่งโหลดโปรแกรม Driver ของการ์ดขึ้นมาในโปรแกรมเพื่อสั่งตรวจสอบหาตำแหน่งการทำงานของการ์ดได้เองโดยตรง ซึ่งไม่ว่าเราจะนำการ์ดไปติดตั้งใช้งานกับคอมพิวเตอร์เครื่องใดก็ตาม โปรแกรมที่เราเขียนขึ้นก็สามารถที่จะตรวจสอบและปรับการทำงานให้สามารถ ใช้งาน ร่วมกับการ์ดได้อย่างถูกต้อง ซึ่งถ้าเป็นวิธีการเรียกใช้ Library DLL เหมือนกับตัวอย่างโปรแกรมทั้ง 3 ตัวอย่าง ที่ผ่านมาแล้วนั้นจะต้องตรวจสอบตำแหน่งการทำงานของการ์ดเพื่อปรับตำแหน่ง Base Address ของโปรแกรมให้ตรงกับการ์ดเสียก่อนจึงจะสามารถใช้งานโปรแกรมร่วมกับการ์ดได้อย่างถูกต้อง

โดยทางทีมงานอีทีที ได้ทำการพัฒนา Library Driver ของการ์ด ET-PCI8255 V3 ให้สามารถใช้งานร่วมกับโปรแกรม Delphi 5 ของ Borland จัดเตรียมไว้สำหรับนักพัฒนาโปรแกรมที่ต้องการเขียนโปรแกรมด้วยภาษา Delphi ได้นำไปใช้งานกันอยู่แล้ว โดย Library Driver ดังกล่าวจะจัดอยู่ในรูปแบบของ File Library ของโปรแกรม Delphi คือ ไฟล์ "ET_PCI8255V3_DP_LIB.PAS" โดยภายใน Library ตัวนี้จะประกอบไปด้วยฟังก์ชันต่างๆที่จะใช้ในการติดต่อกับการ์ด ET-PCI8255 V3 อย่างครบถ้วน

สำหรับแนวทางในการเขียนโปรแกรมภาษา Delphi 5 ร่วมกับ Library Driver ของการ์ด ET-PCI8255 V3 นั้นหลักการเขียนโปรแกรมต่างๆส่วนมากจะยังคงเดิมทั้งหมด จะมีความแตกต่างกันบ้างเล็กน้อยตรงส่วนของคำสั่งที่ใช้ในการประกาศเรียกใช้ Library และการ Input Port และ Output Port เท่านั้นเอง

การติดตั้ง Library Driver

สำหรับการติดตั้ง Library Driver ให้กับระบบปฏิบัติการของ Windows นั้น ใน Folder ของโปรแกรม Driver ที่ทาง อีทีที จัดเตรียมไว้ให้ นั้น จะทำการจัดเตรียม Bat File สำหรับสั่งติดตั้ง Library Driver สำหรับการ์ดไว้ให้ด้วยเรียบร้อยแล้ว ในการติดตั้งนั้นผู้ใช้เพียงแต่สั่ง Run ไฟล์ดังกล่าว โดยให้ทำการ Double Click ที่ไฟล์ดังกล่าว เพื่อทำการสั่งติดตั้ง Library Driver ให้เรียบร้อยเสียก่อน ซึ่งการติดตั้ง Driver นี้จะกระทำเพียงครั้งเดียวเท่านั้น โดยถ้าเป็น Windows 95/98/Me ให้เรียก Run File ชื่อ Install-98.bat แต่ถ้าเป็น Windows 2000/NT/XP ให้เรียก Run File ชื่อ Install-xp.bat แทน หลังจากสั่งติดตั้ง Library Driver ให้กับระบบปฏิบัติการของ Windows เป็นที่เรียบร้อยแล้วก็สามารถเขียนโปรแกรมเพื่อติดต่อกับการ์ด โดยเรียกใช้งานฟังก์ชันต่างๆใน Library ได้ทันที

การใช้งาน Library Driver ไฟล์ ET_PCI8255V3_DP_LIB.PAS

ไฟล์ ET_PCI8255V3_DP_LIB.PAS เป็นไฟล์ Library สำหรับใช้งานกับโปรแกรม Delphi 5 โดยจะประกอบด้วยคำสั่งและฟังก์ชันต่างๆที่ใช้สำหรับสั่งงานการ์ด ET-PCI8255 V3 โดยในการเขียนโปรแกรม Delphi 5 เพื่อเรียกใช้งานฟังก์ชันต่างๆที่บรรจุได้ภายใน Library นี้ ในอันดับแรกจะต้องสั่งติดตั้ง Library Driver ให้กับระบบปฏิบัติการของ Windows ให้เรียบร้อยเสียก่อน ซึ่งในขั้นตอนนี้เป็นการสั่ง Copy File ของ Driver ไปไว้ใน Folder ต่างๆของ Windows ซึ่งเมื่อสั่งติดตั้ง Library เป็นที่เรียบร้อยแล้วก็สามารถเขียนโปรแกรมเพื่อเรียกใช้งานฟังก์ชันต่างๆผ่านทาง Library File ของ Visual Basic ได้ทันที ซึ่งในการเรียกใช้ฟังก์ชันนั้นจะมีขั้นตอนและข้อกำหนดดังนี้คือ

1. ให้ทำการสั่ง Add Library Driver ชื่อ ET_PCI8255V3_DP_LIB.PAS เข้ากับ Project File ของ Delphi 5 ให้เรียบร้อย
2. ให้ทำการติดตั้งการ์ด ET-PCI8255 V3 เข้ากับคอมพิวเตอร์ให้เรียบร้อย
3. ให้ทำการประกาศตัวแปรเพื่อใช้งานในการผ่านค่าข้อมูลระหว่างโปรแกรมหลักกับฟังก์ชันต่างๆดังนี้

```
hET_PCI8255_V3 : ET_PCI8255_V3_HANDLE; // Handle ET-PCI8255 V3
hWD : HANDLE;
PciSlot : WD_PCI_SLOT;
```

ซึ่งหลังจากสั่งประกาศตัวแปรให้กับโปรแกรม เพื่อใช้รองรับการทำงานร่วมกับฟังก์ชันต่างๆ ที่บรรจุไว้ภายใน Library Driver เรียบร้อยแล้ว เราก็สามารถที่จะเริ่มต้นเขียนคำสั่งเพื่อเรียกใช้คำสั่งหรือฟังก์ชันต่างๆจาก Library ได้แล้ว ซึ่งภายใน Library นั้นจะบรรจุคำสั่งและฟังก์ชันต่างๆไว้หลายฟังก์ชัน ซึ่งผู้อ่านสามารถเข้าไปศึกษารายละเอียดการใช้งานฟังก์ชัน และคำสั่งต่างๆได้เอง ซึ่งในที่นี้จะขออธิบายถึง ฟังก์ชันหลักๆที่มีความสำคัญและจำเป็นต้องใช้ในการติดต่อกับการ์ดดังต่อไปนี้

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

การตรวจสอบการโหลด Driver และเปิดการ Handle การ์ด

ในการสั่งตรวจสอบผลการโหลด Driver ของ Windows นั้นจะใช้ฟังก์ชัน PCI_Get_WD_handle() ในการตรวจสอบโดยจะได้ผลลัพธ์เป็นจริงถ้าการทำงานถูกต้อง ซึ่งถ้าได้ผลลัพธ์เป็นเท็จแสดงว่าการติดตั้ง Driver ของระบบปฏิบัติการ Windows ยังไม่สำเร็จ ซึ่งเราจะไม่สามารถเรียกใช้ฟังก์ชันอื่นๆต่อไปได้อีกต้องสั่งปิดการทำงานของโปรแกรมไป แต่ถ้าผลการทำงานของฟังก์ชันเป็นจริงก็แสดงว่า Windows ได้ทำการโหลด Driver เป็นที่เรียบร้อยแล้ว ซึ่งเราสามารถสั่งตรวจสอบการ์ดว่ามีอยู่ในคอมพิวเตอร์หรือไม่ และมีค่าพารามิเตอร์ต่างๆเป็นอย่างไรบ้าง โดยมีรูปแบบดังนี้

```
hET_PCI8255_V3 := nil; // Open Driver on Create Form //
If Not PCI_Get_WD_handle(@hWD) Then Exit; // Make sure Driver is loaded //
WD_Close(hWD); // Close Driver
hET_PCI8255_V3 := ET_PCI8255_V3_LocateAndOpenBoard(ET_PCI8255_V3_DEFAULT_VENDOR_ID,
ET_PCI8255_V3_DEFAULT_DEVICE_ID);
```

โดยส่วนนี้จะเป็นการสั่งตรวจสอบการโหลด Library Driver ของ Windows ซึ่งจะใช้ฟังก์ชันสำหรับตรวจสอบคือ PCI_Get_WD_handle(hWD) โดยการทำงานของฟังก์ชันนี้จะให้ผลเป็น Boolean คือ จริง กับ เท็จ โดยถ้าการสั่งโหลด Driver ทำได้สำเร็จถูกต้องจะให้ผลการทำงานของฟังก์ชันเป็นจริง แต่ถ้าเกิดความผิดพลาด เช่น ยังไม่มีการติดตั้ง Driver ให้กับ Windows ก่อนจะให้ผลเป็นเท็จ ซึ่งถ้าได้ผลเป็นจริง ซึ่งหมายถึง การโหลด Driver ของ Windows สำเร็จ เราก็จะใช้ฟังก์ชัน ET_PCI8255_V3_LocateAndOpenBoard สำหรับทำหน้าที่ตรวจสอบค่าพารามิเตอร์ต่างๆของการ์ด ET-PCI8255 V3 ซึ่งถ้ามีการติดตั้งการ์ด ET-PCI8255 V3 ไว้ในเครื่องเรียบร้อยแล้ว ก็จะได้ผลการทำงานของฟังก์ชันนี้เป็นจริง โดยคำสั่งนี้จะทำการจองค่าพารามิเตอร์ต่างๆของการ์ด จาก Kernel ของ Windows หรือเรียกว่าการเปิด Handle ของการ์ดไว้ เพื่อห้ามไม่ให้โปรแกรม Application อื่นๆมาเรียกใช้งานการ์ดนี้ได้อีกจนกว่าจะมีการคืนค่า Handle ให้ โดยค่าพารามิเตอร์ต่างๆของการ์ดจะถูกเก็บไว้ใน Handle คือ hET_PCI8255_V3 ซึ่งหลังจากนี้เป็นต้นไปเราจะใช้ค่า Handle นี้เป็นค่าอ้างอิงในการติดต่อกับการ์ดได้ตลอดไปจนกว่าจะสั่งคืนค่า Handle

การสั่งคืนค่า Handle ของการ์ด

หลังจากที่เราสั่งจองตำแหน่งการทำงานของการ์ดกับ Kernel ของ Windows ไปแล้ว จะมีผลทำให้ มีเพียงโปรแกรม Application ของเราเท่านั้นที่สามารถเข้าถึงการ์ด ET-PCI8255 V3 ได้ ดังนั้นก่อนที่จะจบการทำงานจาก Application ของเราไป จะต้องมีการสั่งคืนค่าตำแหน่งต่างๆให้กับ Windows เสียก่อนด้วย ไม่งั้นแล้วก็จะทำให้โปรแกรมอื่นๆ หรือแม้แต่โปรแกรม Application ที่เราเขียนขึ้นเองนี้ ไม่สามารถที่จะเข้าถึงการ์ดได้อีกในครั้งต่อไป จนกว่าจะมีการ Shutdown หรือ Restart Windows ใหม่

ซึ่งถ้าเราสั่งเปิด Handle การทำงานของการ์ด ET-PCI8255 V3 กับ Kernel ของ Windows ไว้ แล้ว และก่อนจบการทำงานของโปรแกรมไป ยังไม่มีการสั่งคืนค่า Handle ดังกล่าวให้กับ Kernel ของ Windows อีก ก็จะทำให้ Kernel เข้าใจว่าการ์ด ET-PCI8255 V3 ยังถูกรอครอบครองด้วยโปรแกรม Application ของเราอยู่ ก็จะไม่ยอมให้โปรแกรมอื่นๆเข้ามาเรียกใช้การ์ดได้อีก และในครั้งถัดไปเมื่อเราสั่งของ Handle การทำงานของการ์ดใหม่ก็จะไม่สามารถทำได้อีก จนกว่าจะมีการ Shutdown หรือ Restart Windows ใหม่ ซึ่งการสั่งคืนค่า Handle ของการ์ดนั้น จะต้องกระทำก่อนที่จะจบจากการทำงานของโปรแกรม ซึ่งอาจเขียนไว้ในส่วนของ Event ของการสั่งปิดฟอร์มก็ได้ โดยวิธีการสั่งคืนค่า Handle ของการ์ดจะมีรูปแบบคำสั่งดังนี้

```
If hET_PCI8255_V3 <> nil Then ET_PCI8255_V3_Close(hET_PCI8255_V3);
```

ในส่วนนี้จะเป็นการสั่งตรวจสอบว่า Handle เป็นจริงอยู่หรือไม่ ซึ่งถ้าเป็นจริงก็หมายความว่า เราได้ทำการสั่งเปิด Handle การทำงานของการ์ด ET-PCI8255 V3 ไว้ก่อนแล้วจากคำสั่ง ET_PCI8255_V3_LocateAndOpenBoard ก็ให้สั่งทำการปิดการ Handle ให้เรียบร้อย

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

การสั่งอ่านข้อมูลจากการ์ด

ในการสั่งอ่านข้อมูลจากการ์ด จะใช้ฟังก์ชัน ET_PCI8255_V3_ReadByte โดยฟังก์ชันนี้จะใช้สำหรับสั่งอ่านค่าข้อมูลขนาด 1 ไบต์จากการ์ด โดยจะต้องสร้างตัวแปรแบบ Byte ขึ้นมาไว้รับค่าผลลัพธ์จากฟังก์ชันด้วย โดยจะต้องมีการผ่านค่าตำแหน่ง Offset Address ของตำแหน่งรีจิสเตอร์ หรือพอร์ต I/O ภายในการ์ดที่เราต้องการจะอ่านข้อมูลให้กับฟังก์ชันด้วย ซึ่งมีรูปแบบการใช้งานดังนี้

```
Var := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),Offset);
```

- Var เป็นตัวแปรแบบ Byte สำหรับใช้เก็บผลลัพธ์ของข้อมูลที่อ่านได้จากการ์ด
- Offset เป็นค่า Offset Address ของรีจิสเตอร์ หรือ พอร์ต I/O ที่ต้องการจะอ่านค่า ซึ่งต้องเป็นค่าข้อมูลขนาด 8บิต โดยมีค่าอยู่ระหว่าง \$00 ถึง \$FF ตัวอย่างเช่น ถ้าต้องการอ่านค่าข้อมูลจากพอร์ต PA ของ 8255 ตัวที่ 1 ก็จะต้องกำหนดค่า Offset ให้กับคำสั่งด้วยค่า \$C0 (C0H เป็นตำแหน่งแอดเรสของ PA ในการ์ด) เป็นต้น

ตัวอย่างคำสั่ง การสั่งอ่านข้อมูลจากพอร์ต PA ของ 8255 ตัวที่1 ของการ์ด ET-PCI8255 V3 โดยใช้ฟังก์ชัน ET_PCI8255_V3_ReadByte

```
SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR  
(ET_PCI8255_V3_AD_BAR0), PA1);
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

การส่งเขียนข้อมูลให้กับการ์ด

ในการส่งเขียนข้อมูลให้กับการ์ดจะใช้คำสั่ง ET_PCI8255_V3_WriteByte ซึ่งมีรูปแบบเป็นโปรแกรมย่อย โดยจะต้องมีการส่งผ่านค่าข้อมูลและ Offset Address ของตำแหน่งรีจิสเตอร์หรือพอร์ต I/O ที่ต้องการจะเขียนค่าให้กับคำสั่งด้วย โดยการใช้งานคำสั่งมีรูปแบบดังนี้

```
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR  
(ET_PCI8255_V3_AD_BAR0), Offset, Data);
```

Data เป็นค่าข้อมูลหรือตัวแปรที่เก็บค่าข้อมูลแบบ Byte ที่ต้องการเขียนไปยังการ์ด

Offset เป็นค่าตำแหน่ง Offset Address ขนาด 8 บิต ของรีจิสเตอร์ หรือตำแหน่งพอร์ต I/O ของการ์ดที่ต้องการจะเขียนข้อมูลไปให้

ตัวอย่างคำสั่ง การส่งเขียนข้อมูลไปยังพอร์ต PA ของ 8255 ตัวที่ 1 ของการ์ด ET-PCI8255 V3 โดยใช้ฟังก์ชัน ET_PCI8255_V3_WriteByte

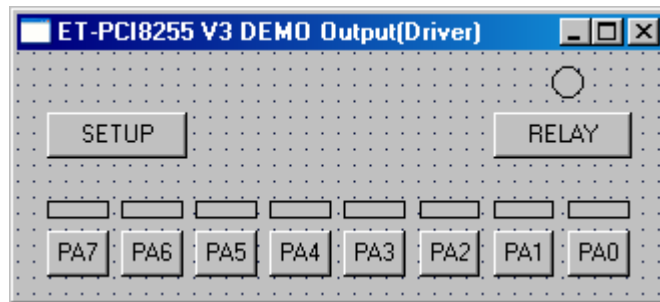
```
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR  
(ET_PCI8255_V3_AD_BAR0), PA1, $55);
```

สำหรับตัวอย่างโปรแกรมของ Delphi 5 สำหรับติดต่อสั่งงานการ์ด ET-PCI8255 V3 ด้วยวิธีการเรียกใช้งาน Library Driver ของการ์ดโดยตรงนั้น จะขอยกตัวอย่างโปรแกรมให้มีคุณสมบัติการทำงานที่เหมือนกับตัวอย่างโปรแกรม ที่ใช้วิธีการเรียกผ่าน DLL ของ InpOut32.DLL ทุกประการ โดยจะปรับเปลี่ยนเฉพาะส่วนของคำสั่งที่ใช้ในการเรียกใช้ Library และ คำสั่งที่ใช้ในการติดต่อ Input / Output Port กับการ์ดเท่านั้น โดยตัวอย่างโปรแกรมในส่วนนี้ผู้เขียนจะขออธิบายถึงเฉพาะส่วนที่เกี่ยวข้องกับการใช้งาน Library Driver ของการ์ดเท่านั้น ส่วนรายละเอียดการทำงานส่วนอื่นๆที่ได้กล่าวถึงไปแล้วใน 3 ตัวอย่างแรก จะไม่ขอกล่าวถึงในที่นี้ อีก ถ้าผู้อ่านเกิดความไม่เข้าใจส่วนใดก็ขอให้ผู้อ่านย้อนกลับไปศึกษารายละเอียดจากตัวอย่างโปรแกรมทั้ง 3 ตัวอย่างที่ผ่านมาแล้วเพิ่มเติมเอง

ตัวอย่างที่ 2.1 Demo Output (เรียกใช้ Library Driver ไฟล์ ET_PCI8255V3_DP_LIB.PAS)

สำหรับตัวอย่างโปรแกรมนี้อาจมีคุณสมบัติ และผลการทำงานเหมือนกันกับตัวอย่างโปรแกรมที่ 1.1 ของวิธีการเรียกใช้ ไฟล์ InpOut32.DLL ทุกอย่าง เพียงแต่ตัวอย่างโปรแกรมนี้อาจจะเปลี่ยนมาใช้ Library Driver แทน โดยในการเขียนโปรแกรมมีลำดับขั้นตอนดังนี้

1. เปิด Project ใหม่โดยไปที่ File > New Application จะได้ฟอร์มเปล่าขึ้นมา 1 ฟอร์ม จากนั้นให้นำคอนโทรลชนิดต่างๆ มาจัดวางใน Form ดังรูป



รูปแสดง การจัดวาง Component บน Form ของโปรแกรม ตัวอย่างที่2.1

โดยในตัวอย่างนี้จะใช้ Component ทั้งหมด 2 ชนิดด้วยกัน คือ Command Button และ Shape ซึ่ง Command Button จะมีทั้งหมด 10 ชุด ใช้สำหรับ Setup การทำงานของการ์ด ET-PCI8255 V3 จำนวน 1 ชุด โดยเมื่อกดปุ่ม SETUP จะเป็นการสั่ง Initial การทำงานของการ์ด ET-PCI8255 V3 โดยจะเริ่มต้นด้วยการเปิดการทำงานของสัญญาณ Chip Select และปิดการทำงานของ Relay ไว้ พร้อมกับ Initial การทำงานของ 8255 ตัวที่ 1 ให้ทำหน้าที่เป็น Output Port ทั้ง 3 พอร์ต (PA,PB และ PC ทำหน้าที่เป็น Output) ส่วน Command Button ที่เหลือจะใช้

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

สำหรับสั่ง ON และ OFF การทำงานของ Relay และ Output ของ PA0..PA7 ตามลำดับ สำหรับ Shape นั้นจะใช้สำหรับทำหน้าที่เป็นเสมือน หลอด LED หรือไฟแสดงสถานะ การทำงานของ Relay และ Output ของ PA0..PA7 โดยถ้า Output มีค่าเป็น Logic “0” จะให้สถานะเป็นสีแดง แต่ถ้า Output มีค่าเป็น Logic “1” จะให้สถานะเป็นสีฟ้าของ Form แทน

2. ทำการกำหนดค่า Properties ให้กับคอนโทรลต่างๆ ดังนี้

ชื่อ	คอนโทรล	พรีอพเพอร์ตี้	ค่าที่กำหนด
Form1	Form	Name	Form1
		Border Style	bsSizeable
		Caption	ET-PCI8255 V3 DEMO Output(Driver)
SETUP	Command Button	Name	SETUP
		Enable	True
		Caption	SETUP
PA0 . . PA7	Command Button	Name	PA01,PA11,PA21,...PA71
		Enable	False
		Caption	PA0,PA1,PA2,...PA7
	Shape	Name	LEDPA01,LEDPA11,LEDPA21,...LEDPA71
		Fill Style	Solid
		Shape	Rectangle
RELAY	Command Button	Name	RELAY
		Enable	False
		Caption	RELAY
	Shape	Name	LEDRelay
		Fill Style	Solid
		Shape	Circle

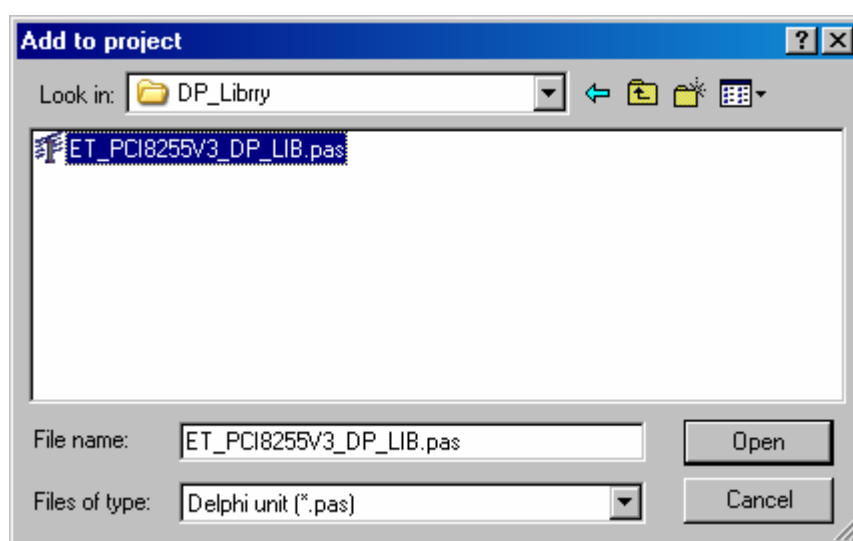
สำหรับการกำหนดค่า พรีอพเพอร์ตี้ ของคอนโทรลต่างๆนั้น จะแสดงเฉพาะค่าที่เปลี่ยนแปลงไปจากค่า Default เดิมเท่านั้น โดยในส่วนของ Command Button นั้นจะเปิดการทำงาน (Enabled) ไว้เฉพาะที่เป็นของ Setup เพียงตัวเดียวเท่านั้น ส่วน Command Button ตัวอื่นๆจะต้องปิด(Disable) ไว้ก่อน เพื่อบังคับให้ผู้ใช้งาน

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ต้องทำการ Setup การทำงานให้กับการ์ดเสียก่อน จากนั้นจึงจะสามารถสั่งงาน Output ต่างๆได้ โดยการทำงานของ Command Button ของ Relay และ PA0..PA7 นั้นจะถูกเปิดการทำงานเมื่อทำการสั่ง Setup แล้ว

โดย Command Button ของพอร์ต PA นั้นจะมีทั้งหมด 8 ชุด คือ PA0 ถึง PA7 โดยกำหนดให้ตำแหน่งของ Command Button สำหรับควบคุมการทำงานของ Output ทั้ง 8 มีชื่อเป็น PA01 ถึง PA71 ตามลำดับ ซึ่ง PA01 หมายถึงพอร์ต PA0 ของ 8255 ตัวที่ 1 ส่วน PA11 ก็จะหมายถึงพอร์ต PA1 ของ 8255 ตัวที่ 1 และ PA71 ก็จะหมายถึงพอร์ต PA7 ของ 8255 ตัวที่ 1 เป็นต้น

3. ทำการเรียกใช้ไฟล์ Library Driver ของการ์ดโดยการสั่งจาก Project → Add to project → ระบุตำแหน่ง Folder และชื่อไฟล์ ET_PCI8255V3_DP_LIB.PAS แล้วเลือก Open ดังรูป



หมายเหตุ สำหรับ Source Code ตัวอย่างนั้น ได้ทำการสั่ง Add Module ดังกล่าวไว้ให้เป็นที่ยอมรับแล้ว โดยเรียกจาก ..\DP_Library\ET_PCI8255V3_DP_LIB.PAS

ซึ่งถ้าต้องการสั่งแปลโปรแกรมตัวอย่างใหม่โดยไม่ให้เกิดความผิดพลาดจะต้องทำการ Copy ไฟล์ตัวอย่าง และไฟล์ Library Driver ให้อยู่ใน Folder ระดับชั้นเดียวกัน ตัวอย่างเช่น

```
C:\>DemoDP_Driver\Demo1\Project1.dpr
```

```
C:\> DemoDP_Driver\DP_Library\ET_PCI8255V3_DP_LIB.PAS
```

ซึ่งถ้ามีการ Copy ไฟล์ไปไว้ใน Folder ที่ต่างไปจากนี้จะต้องทำการสั่ง Add to project ใหม่ให้ถูกต้องด้วย ไมเช่นนั้นแล้วจะไม่สามารถสั่งแปลโปรแกรมได้

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

4. เขียนโค้ดคำสั่งเพื่อควบคุมการทำงานของโปรแกรกดังต่อไปนี้

```
{
Demo1 Output Demo Access Driver "ET_PCI8255V3_DP_LIB.PAS"
Auto Detect ET-PCI8255 V3 Hardware for Handle Function
}

Unit Unit1;

Interface

Uses   Windows,Messages,SysUtils,Classes,Graphics,Controls,Forms,Dialogs,StdCtrls,ExtCtrls,
       ET_PCI8255V3_DP_LIB;           /* Used ET-PCI8255 V3 Delphi Library */

Type   TForm1 = class(TForm)
       SETUP_8255: TButton;
       RELAY: TButton;
       PA01: TButton;
       PA11: TButton;
       PA21: TButton;
       PA31: TButton;
       PA41: TButton;
       PA51: TButton;
       PA61: TButton;
       PA71: TButton;
       LEDRELAY: TShape;
       LEDPA01: TShape;
       LEDPA11: TShape;
       LEDPA21: TShape;
       LEDPA31: TShape;
       LEDPA41: TShape;
       LEDPA51: TShape;
       LEDPA61: TShape;
       LEDPA71: TShape;

       Procedure SETUP_8255Click(Sender: TObject);
       Procedure RELAYClick(Sender: TObject);
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure PA01Click(Sender: TObject);
Procedure PA11Click(Sender: TObject);
Procedure PA21Click(Sender: TObject);
Procedure PA31Click(Sender: TObject);
Procedure PA41Click(Sender: TObject);
Procedure PA51Click(Sender: TObject);
Procedure PA61Click(Sender: TObject);
Procedure PA71Click(Sender: TObject);
Procedure FormClose(Sender: TObject; var Action: TCloseAction);

Private
  { Private declarations }
Public
  { Public declarations }
End;

Var
  Form1          : TForm1;
  hET_PCI8255_V3 : ET_PCI8255_V3_HANDLE;    // Handle ET-PCI8255 V3
  hWD            : HANDLE;
  PciSlot       : WD_PCI_SLOT;

Implementation

// Tiger-320 Register Offset
Const  PIB = $00;          // Reset & PIB Cycle
Const  AUXC = $02;        // Aux Direction Port
Const  AUXD = $03;        // Aux Data Port

Const  PA1 = $C0;         // Port-A 8255#1
Const  PB1 = $C4;         // Port-B 8255#1
Const  PC1 = $C8;         // Port-C 8255#1
Const  PCC1 = $CC;        // Port Control 8255#1
Const  PA2 = $D0;         // Port-A 8255#2
Const  PB2 = $D4;         // Port-B 8255#2
Const  PC2 = $D8;         // Port-C 8255#2
Const  PCC2 = $DC;        // Port Control 8255#2
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Const PA3 = $E0;           // Port-A 8255#3
Const PB3 = $E4;           // Port-B 8255#3
Const PC3 = $E8;           // Port-C 8255#3
Const PCC3 = $EC;         // Port Control 8255#3

Const ON_Bit0 = $01;       // XXXX XXXX OR 0000 0001 = XXXX XXX1
Const OFF_Bit0 = $FE;      // XXXX XXXX AND 1111 1110 = XXXX XXX0
Const ON_Bit1 = $02;       // XXXX XXXX OR 0000 0010 = XXXX XX1X
Const OFF_Bit1 = $FD;      // XXXX XXXX AND 1111 1101 = XXXX XX0X
Const ON_Bit2 = $04;       // XXXX XXXX OR 0000 0100 = XXXX X1XX
Const OFF_Bit2 = $FB;      // XXXX XXXX AND 1111 1011 = XXXX X0XX
Const ON_Bit3 = $08;       // XXXX XXXX OR 0000 1000 = XXXX 1XXX
Const OFF_Bit3 = $F7;      // XXXX XXXX AND 1111 0111 = XXXX 0XXX
Const ON_Bit4 = $10;       // XXXX XXXX OR 0001 0000 = XXX1 XXXX
Const OFF_Bit4 = $EF;      // XXXX XXXX AND 1110 1111 = XXX0 XXXX
Const ON_Bit5 = $20;       // XXXX XXXX OR 0010 0000 = XX1X XXXX
Const OFF_Bit5 = $DF;      // XXXX XXXX AND 1101 1111 = XX0X XXXX
Const ON_Bit6 = $40;       // XXXX XXXX OR 0100 0000 = X1XX XXXX
Const OFF_Bit6 = $BF;      // XXXX XXXX AND 1011 1111 = X0XX XXXX
Const ON_Bit7 = $80;       // XXXX XXXX OR 1000 0000 = 1XXX XXXX
Const OFF_Bit7 = $7F;      // XXXX XXXX AND 0111 1111 = 0XXX XXXX

Const OutputOFF = CIMenu;  // Color of Output OFF ("0") Status
Const OutputON = CRed;     // Color of Output ON ("1") Status
Const InputOFF = Clmenu;   // Color of Input OFF ("1") Status
Const InputON = Clime;     // Color of Input ON ("0") Status

{$R *.DFM}

Procedure TForm1.SETUP_8255Click(Sender: TObject);
Var SetupData : Byte;

Begin
  hET_PCI8255_V3 := nil;           // Open Driver on Create Form //
  If Not PCI_Get_WD_handle(@hWD) Then Exit; // Make sure Driver is loaded //
  WD_Close(hWD);                 // Close Driver
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
hET_PCI8255_V3 := ET_PCI8255_V3_LocateAndOpenBoard(ET_PCI8255_V3_DEFAULT_VENDOR_ID,
                                                    ET_PCI8255_V3_DEFAULT_DEVICE_ID);

If hET_PCI8255_V3 <> nil Then
Begin
    // Initial Reset and 8255 Bus Cycle //
    SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        PIB);           // Read PIB Reset Port
    SetupData := SetupData AND OFF_BIT0;           // Bit0 = EXTRST# = "0" (Reset:RES#)
    SetupData := SetupData OR  ON_BIT5;           // Bit5:4 = 11 = PIB Cycle Slowest
    SetupData := SetupData OR  ON_BIT4;
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PIB,
                            SetupData);           // Active RES# & Relay

    // Initial Data (AUX) For CS# and Relay //
    SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        AUXD);       // Read Aux Data Port
    SetupData := SetupData AND OFF_BIT0;           // Bit0 = Aux0 = "0" (Enable CS)
    SetupData := SetupData OR  ON_BIT4;           // Bit4 = Aux4 = "1" (Relay OFF)
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXD,
                            SetupData);           // Active Chips Select & Relay
    LEDRelay.Brush.Color := OutputOFF;           // LED Relay = OFF

    // Initial Direction (AUX) For CS# and Relay //
    SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        AUXC);       // Read Aux Port Direction
    SetupData := SetupData OR ON_BIT4;           // Aux4 = "1" = Output
    SetupData := SetupData OR ON_BIT0;           // Aux0 = "1" = Output
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXC,
                            SetupData);           // Setup Aux Direction

    // Initial 8255#1 = All Output Port //
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PCC1, $80);
    Setup_8255.Enabled := False;                 // Disable Setup After Setup
    RELAY.Enabled := True;                       // Enable Relay Control
    PA01.Enabled := True;                       // Enable Output Control
    PA11.Enabled := True;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
PA21.Enabled := True;
PA31.Enabled := True;
PA41.Enabled := True;
PA51.Enabled := True;
PA61.Enabled := True;
PA71.Enabled := True;
End;
End;

Procedure TForm1.RelayClick(Sender: TObject);
Var  InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     AUXD);           // Read Output Latch AUX

  CheckData := InOutData AND ON_BIT4;                // Check Last AUX4 Status

  If CheckData = $10 Then
  Begin
    InOutData := InOutData AND OFF_BIT4;             // AUX4 = "0" (1110 1111)
    LEDRelay.Brush.Color := OutputON;                // LED Relay = ON
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT4;               // AUX4 = "1" (0001 0000)
    LEDRelay.Brush.Color := OutputOFF;               // LED Relay = OFF
  End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXD,
                          InOutData);               // Update AuxD
End;

Procedure TForm1.PA01Click(Sender: TObject);
Var  InOutData : Byte;
      CheckData : Byte;

Begin
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A
CheckData := InOutData AND ON_BIT0;           // Check Last PA0 Status
If CheckData = $01 Then
Begin
    InOutData := InOutData AND OFF_BIT0;       // PA0 = "0" (1111 1110)
    LEDPA01.Brush.Color := OutputOFF;         // LED PA0 of 8255#1 = OFF
End
Else
Begin
    InOutData := InOutData OR ON_BIT0;        // PA0 = "1" (0000 0001)
    LEDPA01.Brush.Color := OutputON;         // LED PA0 of 8255#1 = ON
End;
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA11Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
    InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT1;           // Check Last PA1 Status
    If CheckData = $02 Then
    Begin
        InOutData := InOutData AND OFF_BIT1;       // PA1 = "0" (1111 1101)
        LEDPA11.Brush.Color := OutputOFF;         // LED PA1 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT1;        // PA1 = "1" (0000 0010)
        LEDPA11.Brush.Color := OutputON;         // LED PA1 of 8255#1 = ON
    End;
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure TForm1.PA21Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A

  CheckData := InOutData AND ON_BIT2;             // Check Last PA2 Status

  If CheckData = $04 Then
  Begin
    InOutData := InOutData AND OFF_BIT2;         // PA2 = "0" (1111 1011)
    LEDPA21.Brush.Color := OutputOFF;           // LED PA2 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT2;          // PA2 = "1" (0000 0100)
    LEDPA21.Brush.Color := OutputON;            // LED PA2 of 8255#1 = ON
  End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA31Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A

  CheckData := InOutData AND ON_BIT3;             // Check Last PA3 Status

  If CheckData = $08 Then
  Begin
    InOutData := InOutData AND OFF_BIT3;         // PA3 = "0" (1111 0111)
    LEDPA31.Brush.Color := OutputOFF;           // LED PA3 of 8255#1 = OFF
  End
  Else
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Begin
  InOutData := InOutData OR ON_BIT3;           // PA3 = "1" (0000 1000)
  LEDPA31.Brush.Color := OutputON;           // LED PA3 of 8255#1 = ON
End;
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA41Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT4;           // Check Last PA4 Status
  If CheckData = $10 Then
  Begin
    InOutData := InOutData AND OFF_BIT4;       // PA4 = "0" (1110 1111)
    LEDPA41.Brush.Color := OutputOFF;         // LED PA4 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT4;         // PA4 = "1" (0001 0000)
    LEDPA41.Brush.Color := OutputON;          // LED PA4 of 8255#1 = ON
  End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA51Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT5;         // Check Last PA5 Status
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
If CheckData = $20 Then
Begin
  InOutData := InOutData AND OFF_BIT5;           // PA5 = "0" (1101 1111)
  LEDPA51.Brush.Color := OutputOFF;             // LED PA5 of 8255#1 = OFF
End
Else
Begin
  InOutData := InOutData OR ON_BIT5;           // PA5 = "1" (0010 0000)
  LEDPA51.Brush.Color := OutputON;             // LED PA5 of 8255#1 = ON
End;
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA61Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT6;           // Check Last PA6 Status
  If CheckData = $40 Then
  Begin
    InOutData := InOutData AND OFF_BIT6;       // PA6 = "0" (1011 1111)
    LEDPA61.Brush.Color := OutputOFF;         // LED PA6 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT6;       // PA6 = "1" (0100 0000)
    LEDPA61.Brush.Color := OutputON;         // LED PA6 of 8255#1 = ON
  End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA71Click(Sender: TObject);
Var  InOutData : Byte;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1);           // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT7;           // Check Last PA7 Status
  If CheckData = $80 Then
    Begin
      InOutData := InOutData AND OFF_BIT7;           // PA7 = "0" (0111 1111)
      LEDPA71.Brush.Color := OutputOFF;           // LED PA7 of 8255#1 = OFF
    End
  Else
    Begin
      InOutData := InOutData OR ON_BIT7;           // PA7 = "1" (1000 0000)
      LEDPA71.Brush.Color := OutputON;           // LED PA7 of 8255#1 = ON
    End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
Begin
  //Close ET-PCI8255 V3 Driver Before Exit Program //
  If hET_PCI8255_V3 <> nil Then ET_PCI8255_V3_Close(hET_PCI8255_V3);
End;

End.
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

อธิบายการทำงานของคำสั่งในโปรแกรม ตัวอย่างที่ 2.1

```
Private
  { Private declarations }

Public
  { Public declarations }

End;

Var
  Form1          : TForm1;
  hET_PCI8255_V3 : ET_PCI8255_V3_HANDLE; // Handle ET-PCI8255 V3
  hWD            : HANDLE;
  PciSlot       : WD_PCI_SLOT;

Implementation
```

ส่วนนี้เป็นการประกาศตัวแปรสำหรับรองรับการเรียกใช้งาน Library Driver ซึ่งจะต้องประกาศไว้ในส่วนของโปรแกรมหลักของ Form ด้วย สำหรับส่วนของโปรแกรมที่ทำหน้าที่เรียกใช้งาน Library Driver นั้น จะอยู่ในส่วนของโปรแกรมน้อย Procedure TForm1.SETUP_8255Click(Sender: TObject); ซึ่งมีคำสั่งดังนี้

```
hET_PCI8255_V3 := nil; // Open Driver on Create Form //
If Not PCI_Get_WD_handle(@hWD) Then Exit; // Make sure Driver is loaded //
WD_Close(hWD); // Close Driver
hET_PCI8255_V3 := ET_PCI8255_V3_LocateAndOpenBoard(ET_PCI8255_V3_DEFAULT_VENDOR_ID,
                                                    ET_PCI8255_V3_DEFAULT_DEVICE_ID);

If hET_PCI8255_V3 <> nil Then
Begin
  // Initial Reset and 8255 Bus Cycle //
  SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                      PIB); // Read PIB Reset Port
  SetupData := SetupData AND OFF_BIT0; // Bit0 = EXTRST# = "0" (Reset:RES#)
  SetupData := SetupData OR ON_BIT5; // Bit5:4 = 11 = PIB Cycle Slowest
  SetupData := SetupData OR ON_BIT4;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PIB,
                          SetupData);
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ซึ่งจะเห็นว่าในครั้งแรกนั้นจะมีการกำหนดค่าตัวแปร Boolean ให้เป็นเท็จไว้ก่อน จากนั้นจึงทำการสั่งตรวจสอบการโหลด Library Driver ว่าเป็นจริงหรือไม่โดยใช้คำสั่ง PCI_Get_WD_handle(@HWD) ซึ่งถ้าผลออกมาเป็นเท็จก็แสดงว่าการโหลด Library Driver ไม่สำเร็จ ซึ่งอาจเกิดจากยังไม่ได้ทำการติดตั้ง Driver ให้กับ Windows ซึ่งโปรแกรมก็จะจบการทำงานไป แต่ถ้าการโหลด Library Driver ถูกต้องเรียบร้อยแล้วก็จะตรวจสอบหาการ์ดว่าได้รับการติดตั้งไว้ภายในเครื่องคอมพิวเตอร์หรือไม่โดยใช้ตัวแปร hET_PCI8255_V3 ในการตรวจสอบ ซึ่งถ้าหากว่า Driver สามารถตรวจสอบการ์ด ET-PCI8255 V3 ได้ถูกต้องก็จะให้ผลเป็นจริง แต่ถ้าไม่สามารถตรวจสอบการ์ดได้ก็จะยังให้ผลเป็นเท็จอยู่เช่นเดิม ซึ่งถ้าหากว่าได้ผลเป็นจริงในขั้นตอนนี้แล้ว หลังจากนั้นไปเราก็สามารถเรียกใช้คำสั่งสำหรับอ่านค่าและเขียนค่าข้อมูลให้กับการ์ดได้ทันที โดยรูปแบบการส่งอ่านข้อมูลให้กับการ์ดผ่าน Library Driver จากตัวอย่างจะมีรูปแบบดังนี้คือ

```
SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR(ET_PCI8255_V3_AD_BAR0),  
PIB); // Read PIB Reset Port
```

จากรูปแบบคำสั่งข้างต้น SetupData เป็นตัวแปรแบบ Byte ใช้สำหรับส่งผ่านข้อมูลระหว่างโปรแกรมและฟังก์ชันคำสั่งโดยบรรทัดแรกนั้นเป็นการสั่งอ่านข้อมูลจากการ์ดเข้ามาเก็บไว้ยังตัวแปร โดยตำแหน่งแอดเดรสที่จะอ่านนั้นจะกำหนดเฉพาะค่า Offset ของตำแหน่ง เท่านั้นส่วนค่า Base Address จะถูกกำหนดโดยอัตโนมัติจาก Driver อยู่แล้วโดยจากตัวอย่างจะเป็นการสั่งอ่านค่าจากรีจิสเตอร์ PIB ซึ่งถ้าต้องการเปลี่ยนไปอ่านค่าจากรีจิสเตอร์อื่นๆก็ให้เปลี่ยนค่า Offset ของคำสั่งให้ตรงกับตำแหน่งรีจิสเตอร์หรือ I/O Address ที่ต้องการอ่านแทน เช่น ถ้าต้องการอ่านค่าจากพอร์ต PA ของ 8255 ตัวที่ 1 ก็ให้เปลี่ยนค่า Offset จาก PIB เป็น PA1 แทนเป็นต้น

ส่วนการสั่งเขียนข้อมูลไปยังการ์ด โดยผ่าน Library Driver นั้นจะมีรูปแบบการใช้คำสั่ง ตรงกันข้ามกับการสั่งอ่านข้อมูล ดังตัวอย่าง

```
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR(ET_PCI8255_V3_AD_BAR0), PIB, SetupData);
```

จากตัวอย่างเป็นการสั่งเขียนข้อมูลซึ่งกำหนดไว้ในตัวแปร SetupData ออกไปยังรีจิสเตอร์ PIB ของการ์ด ซึ่งถ้าต้องการเปลี่ยนตำแหน่งการเขียนไปยังรีจิสเตอร์อื่นๆ เช่น พอร์ต PA ของ 8255 ตัวที่ 1 ก็ให้เปลี่ยนค่า Offset จาก PIB เป็น PA1 แทน

```
Procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
Begin
  //Close ET-PCI8255 V3 Driver Before Exit Program //
  If hET_PCI8255_V3 <> nil Then ET_PCI8255_V3_Close(hET_PCI8255_V3);
End;
```

ในส่วนนี้เป็นส่วนของโปรแกรมย่อยสำหรับสั่งปิดการทำงานของ Library Driver โดยจะต้องกระทำก่อนจบการทำงานของโปรแกรม ซึ่งจากตัวอย่างจะเขียน Code คำสั่งนี้ไว้ใน Event ของการปิดฟอร์ม ซึ่งเหตุการณ์นี้จะเกิดขึ้นตอนสั่ง Close โปรแกรมทุกครั้ง

สำหรับการทำงานส่วนอื่นๆของโปรแกรมนั้นจะเหมือนกันกับตัวอย่างโปรแกรมทั้ง 3 ตัวอย่าง ที่ผ่านมาแล้วทุกอย่าง ซึ่งจะมีการเปลี่ยนแปลงเฉพาะส่วนของคำสั่งสำหรับอ่านค่าข้อมูลจากการ์ด และการส่งเขียนข้อมูลไปยังการ์ดเท่านั้นเอง

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ตัวอย่างที่ 2.2 DEMO Input (เรียกใช้ Library Driver ไฟล์ ET_PCI8255V3_DP_LIB.PAS)

สำหรับตัวอย่างนี้ จะดัดแปลงมาจากตัวอย่างโปรแกรมที่ 1.2 ซึ่งวิธีการและผลการทำงานของโปรแกรมจะเหมือนกันทุกประการ เพียงแต่เปลี่ยนรูปแบบคำสั่งในการติดต่อกับการ์ด จากการเรียกใช้ InpOut32.DLL มาเป็นการเรียกใช้ Library Driver ของการ์ดแทน ซึ่งมีรูปแบบการเขียนโปรแกรมดังต่อไปนี้

```
{
Demo2 Input Demo Access Driver "ET_PCI8255V3_DP_LIB.PAS"
Auto Detect ET-PCI8255 V3 Hardware for Handle Function
}

Unit Unit1;
Interface
Uses   Windows,Messages, SysUtils,Classes,Graphics,Controls,Forms,Dialogs,StdCtrls,ExtCtrls,
      ET_PCI8255V3_DP_LIB;           /**Used ET-PCI8255 V3 Delphi Library**/
Type   TForm1           = class(TForm)
      SETUP_8255       : TButton;
      RELAY             : TButton;
      LEDRELAY         : TShape;
      LEDPA01          : TShape;
      LEDPA11          : TShape;
      LEDPA21          : TShape;
      LEDPA31          : TShape;
      LEDPA41          : TShape;
      LEDPA51          : TShape;
      LEDPA61          : TShape;
      LEDPA71          : TShape;
      PA0Name          : TLabel;
      PA1Name          : TLabel;
      PA2Name          : TLabel;
      PA3Name          : TLabel;
      PA4Name          : TLabel;
      PA5Name          : TLabel;
      PA6Name          : TLabel;
      PA7Name          : TLabel;
      Timer1           : TTimer;
end;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure      SETUP_8255Click(Sender: TObject);
Procedure      RELAYClick(Sender: TObject);
Procedure      Timer1Timer(Sender: TObject);
Procedure      FormClose(Sender: TObject; var Action: TCloseAction);

Private
  { Private declarations }
Public
  { Public declarations }
End;

Var
  Form1          : TForm1;
  BaseAddress    : Word;
  hET_PCI8255_V3 : ET_PCI8255_V3_HANDLE;
  hWD            : HANDLE;
  PciSlot       : WD_PCI_SLOT;

Implementation

// Tiger-320 Register Offset
Const  PIB      = $00;      // Reset & PIB Cycle
Const  AUXC     = $02;      // Aux Direction Port
Const  AUXD     = $03;      // Aux Data Port

Const  PA1      = $C0;      // Port-A 8255#1
Const  PB1      = $C4;      // Port-B 8255#1
Const  PC1      = $C8;      // Port-C 8255#1
Const  PCC1     = $CC;      // Port Control 8255#1
Const  PA2      = $D0;      // Port-A 8255#2
Const  PB2      = $D4;      // Port-B 8255#2
Const  PC2      = $D8;      // Port-C 8255#2
Const  PCC2     = $DC;      // Port Control 8255#2
Const  PA3      = $E0;      // Port-A 8255#3
Const  PB3      = $E4;      // Port-B 8255#3
Const  PC3      = $E8;      // Port-C 8255#3
Const  PCC3     = $EC;      // Port Control 8255#3
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```

Const  ON_Bit0      = $01;           // XXXX XXXX OR 0000 0001 = XXXX XXX1
Const  OFF_Bit0     = $FE;           // XXXX XXXX AND 1111 1110 = XXXX XXX0
Const  ON_Bit1      = $02;           // XXXX XXXX OR 0000 0010 = XXXX XX1X
Const  OFF_Bit1     = $FD;           // XXXX XXXX AND 1111 1101 = XXXX XX0X
Const  ON_Bit2      = $04;           // XXXX XXXX OR 0000 0100 = XXXX X1XX
Const  OFF_Bit2     = $FB;           // XXXX XXXX AND 1111 1011 = XXXX X0XX
Const  ON_Bit3      = $08;           // XXXX XXXX OR 0000 1000 = XXXX 1XXX
Const  OFF_Bit3     = $F7;           // XXXX XXXX AND 1111 0111 = XXXX 0XXX
Const  ON_Bit4      = $10;           // XXXX XXXX OR 0001 0000 = XXX1 XXXX
Const  OFF_Bit4     = $EF;           // XXXX XXXX AND 1110 1111 = XXX0 XXXX
Const  ON_Bit5      = $20;           // XXXX XXXX OR 0010 0000 = XX1X XXXX
Const  OFF_Bit5     = $DF;           // XXXX XXXX AND 1101 1111 = XX0X XXXX
Const  ON_Bit6      = $40;           // XXXX XXXX OR 0100 0000 = X1XX XXXX
Const  OFF_Bit6     = $BF;           // XXXX XXXX AND 1011 1111 = X0XX XXXX
Const  ON_Bit7      = $80;           // XXXX XXXX OR 1000 0000 = 1XXX XXXX
Const  OFF_Bit7     = $7F;           // XXXX XXXX AND 0111 1111 = 0XXX XXXX

Const  OutputOFF    = CIMenu;        // Color of Output OFF ("0") Status
Const  OutputON     = CIRed;         // Color of Output ON ("1") Status
Const  InputOFF     = Clmenu;        // Color of Input OFF ("1") Status
Const  InputON      = CILime;        // Color of Input ON ("0") Status

{$R *.DFM}

Procedure TForm1.SETUP_8255Click(Sender: TObject);
Var  SetupData : Byte;

Begin
  hET_PCI8255_V3 := nil;
  If Not PCI_Get_WD_handle(@hWD) Then Exit;    // Make sure Driver is loaded //
  WD_Close(hWD);                               // Close Driver
  hET_PCI8255_V3 := ET_PCI8255_V3_LocateAndOpenBoard(ET_PCI8255_V3_DEFAULT_VENDOR_ID,
                                                    ET_PCI8255_V3_DEFAULT_DEVICE_ID);

  If hET_PCI8255_V3 <> nil Then
  Begin
    // Initial Reset and 8255 Bus Cycle //
  
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PIB);           // Read PIB Reset Port

SetupData := SetupData AND OFF_BIT0;           // Bit0 = EXTRST# = "0" (Reset:RES#)

SetupData := SetupData OR  ON_BIT5;           // Bit5:4 = 11 = PIB Cycle Slowest

SetupData := SetupData OR  ON_BIT4;

ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PIB,
                        SetupData);           // Active RES# & Relay

// Initial Data (AUX) For CS# and Relay //

SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     AUXD);       // Read Aux Data Port

SetupData := SetupData AND OFF_BIT0;           // Bit0 = Aux0 = "0" (Enable CS)

SetupData := SetupData OR  ON_BIT4;           // Bit4 = Aux4 = "1" (Relay OFF)

ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXD,
                        SetupData);           // Active Chips Select & Relay

LEDRelay.Brush.Color := OutputOFF;           // LED Relay = OFF

// Initial Direction (AUX) For CS# and Relay //

SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     AUXC);       // Read Aux Port Direction

SetupData := SetupData OR ON_BIT4;           // Aux4 = "1" = Output

SetupData := SetupData OR ON_BIT0;           // Aux0 = "1" = Output

ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXC,
                        SetupData);           // Setup Aux Direction

// Initial 8255#1 = All Output Port //

Setup_8255.Enabled := False;                 // Disable Setup After Setup

ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PCC1, $9B);

RELAY.Enabled := True;                       // Enable Relay Control

Timer1.Enabled := True;                      // Start Timer For Auto Read

End;

End;

Procedure TForm1.RELAYClick(Sender: TObject);

Var  InOutData : Byte;

     CheckData : Byte;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     AUXD); // Read Output Latch AUX
  CheckData := InOutData AND ON_BIT4;      // Check Last AUX4 Status
  If CheckData = $10 Then
  Begin
    InOutData := InOutData AND OFF_BIT4;    // AUX4 = "0" (1110 1111)
    LEDRelay.Brush.Color := OutputON;      // LED RElay = ON
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT4;      // AUX4 = "1" (0001 0000)
    LEDRelay.Brush.Color := OutputOFF;     // LED Relay = OFF
  End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXD,
                          InOutData);      // Update AuxD
End;

Procedure TForm1.Timer1Timer(Sender: TObject);
Var  InputData : Byte;
     CheckData : Byte;
Begin
  InputData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1); // Read Input/Output Latch of PA-8255#1
  CheckData := InputData AND ON_BIT0;     // Check PA0 Status
  If CheckData = ON_BIT0 Then
    LEDPA01.Brush.Color := InputOFF      // PA0 = ("1")
  Else LEDPA01.Brush.Color := InputON;    // PA0 = ("0")

  CheckData := InputData AND ON_BIT1;     // Check PA1 Status
  If CheckData = ON_BIT1 Then
    LEDPA11.Brush.Color := InputOFF      // PA1 = ("1")
  Else LEDPA11.Brush.Color := InputON;    // PA1 = ("0")

  CheckData := InputData AND ON_BIT2;     // Check PA2 Status
  If CheckData = ON_BIT2 Then
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
LEDPA21.Brush.Color := InputOFF           // PA2 = ("1")
Else LEDPA21.Brush.Color := InputON;      // PA2 = ("0")

CheckData := InputData AND ON_BIT3;      // Check PA3 Status
If CheckData = ON_BIT3 Then
    LEDPA31.Brush.Color := InputOFF       // PA3 = ("1")
Else LEDPA31.Brush.Color := InputON;     // PA3 = ("0")

CheckData := InputData AND ON_BIT4;      // Check PA4 Status
If CheckData = ON_BIT4 Then
    LEDPA41.Brush.Color := InputOFF       // PA4 = ("1")
Else LEDPA41.Brush.Color := InputON;     // PA4 = ("0")

CheckData := InputData AND ON_BIT5;      // Check PA5 Status
If CheckData = ON_BIT5 Then
    LEDPA51.Brush.Color := InputOFF       // PA5 = ("1")
Else LEDPA51.Brush.Color := InputON;     // PA5 = ("0")

CheckData := InputData AND ON_BIT6;      // Check PA6 Status
If CheckData = ON_BIT6 Then
    LEDPA61.Brush.Color := InputOFF       // PA6 = ("1")
Else LEDPA61.Brush.Color := InputON;     // PA6 = ("0")

CheckData := InputData AND ON_BIT7;      // Check PA7 Status
If CheckData = ON_BIT7 Then
    LEDPA71.Brush.Color := InputOFF       // PA7 = ("1")
Else LEDPA71.Brush.Color := InputON;     // PA7 = ("0")
End;

Procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
Begin
    If hET_PCI8255_V3 <> nil Then ET_PCI8255_V3_Close(hET_PCI8255_V3);
End;

End.
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

ตัวอย่างที่ 2.3 DEMO Input/Output (เรียกใช้ Library Driver ไฟล์ ET_PCI8255V3_DP_LIB.PAS)

สำหรับตัวอย่างนี้ จะดัดแปลงมาจากตัวอย่างโปรแกรมที่ 1.3 ซึ่งวิธีการและผลการทำงานของโปรแกรมจะเหมือนกันทุกประการ เพียงแต่เปลี่ยนรูปแบบคำสั่งในการติดต่อกับการ์ด จากการเรียกใช้ InpOut32.DLL มาเป็นการเรียกใช้ Library Driver ของการ์ดแทน ซึ่งมีรูปแบบการเขียนโปรแกรมหาดังต่อไปนี้

```
{
Demo3 Input/Output Demo Access Driver "ET-PCI8255V3_DP_LIB.PAS"
Auto Detect ET-PCI8255 V3 Hardware for Handle Function
}

Unit Unit1;
Interface
Uses  Windows,Messages,SysUtils,Classes,Graphics,Controls,Forms,Dialogs,StdCtrls,ExtCtrls,
      ET_PCI8255V3_DP_LIB;          /* Used ET-PCI8255 V3 Delphi Library */

Type  TForm1 = class(TForm)
      SETUP_8255: TButton;
      RELAY: TButton;
      PA01: TButton;
      PA11: TButton;
      PA21: TButton;
      PA31: TButton;
      PA41: TButton;
      PA51: TButton;
      PA61: TButton;
      PA71: TButton;
      LEDRELAY: TShape;
      LEDPA01: TShape;
      LEDPA11: TShape;
      LEDPA21: TShape;
      LEDPA31: TShape;
      LEDPA41: TShape;
      LEDPA51: TShape;
      LEDPA61: TShape;
      LEDPA71: TShape;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Control_PA8255_1: TRadioGroup;

Timer1: TTimer;

Procedure SETUP_8255Click(Sender: TObject);

Procedure RELAYClick(Sender: TObject);

Procedure PA01Click(Sender: TObject);

Procedure PA11Click(Sender: TObject);

Procedure PA21Click(Sender: TObject);

Procedure PA31Click(Sender: TObject);

Procedure PA41Click(Sender: TObject);

Procedure PA51Click(Sender: TObject);

Procedure PA61Click(Sender: TObject);

Procedure PA71Click(Sender: TObject);

Procedure Control_PA8255_1Click(Sender: TObject);

Procedure Timer1Timer(Sender: TObject);

Procedure FormClose(Sender: TObject; var Action: TCloseAction);

Private
  { Private declarations }

Public
  { Public declarations }

End;

Var
  Form1          : TForm1;
  BaseAddress    : Word;
  hET_PCI8255_V3 : ET_PCI8255_V3_HANDLE;
  hWD            : HANDLE;
  PciSlot        : WD_PCI_SLOT;

Implementation

// Tiger-320 Register Offset
Const  PIB      = $00;           // Reset & PIB Cycle
Const  AUXC     = $02;           // Aux Direction Port
Const  AUXD     = $03;           // Aux Data Port
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```

Const PA1   = $C0;           // Port-A 8255#1
Const PB1   = $C4;           // Port-B 8255#1
Const PC1   = $C8;           // Port-C 8255#1
Const PCC1  = $CC;           // Port Control 8255#1
Const PA2   = $D0;           // Port-A 8255#2
Const PB2   = $D4;           // Port-B 8255#2
Const PC2   = $D8;           // Port-C 8255#2
Const PCC2  = $DC;           // Port Control 8255#2
Const PA3   = $E0;           // Port-A 8255#3
Const PB3   = $E4;           // Port-B 8255#3
Const PC3   = $E8;           // Pprt-C 8255#3
Const PCC3  = $EC;           // Port Control 8255#3

Const ON_Bit0   = $01;           // XXXX XXXX OR 0000 0001 = XXXX XXX1
Const OFF_Bit0  = $FE;           // XXXX XXXX AND 1111 1110 = XXXX XXX0
Const ON_Bit1   = $02;           // XXXX XXXX OR 0000 0010 = XXXX XX1X
Const OFF_Bit1  = $FD;           // XXXX XXXX AND 1111 1101 = XXXX XX0X
Const ON_Bit2   = $04;           // XXXX XXXX OR 0000 0100 = XXXX X1XX
Const OFF_Bit2  = $FB;           // XXXX XXXX AND 1111 1011 = XXXX X0XX
Const ON_Bit3   = $08;           // XXXX XXXX OR 0000 1000 = XXXX 1XXX
Const OFF_Bit3  = $F7;           // XXXX XXXX AND 1111 0111 = XXXX 0XXX
Const ON_Bit4   = $10;           // XXXX XXXX OR 0001 0000 = XXX1 XXXX
Const OFF_Bit4  = $EF;           // XXXX XXXX AND 1110 1111 = XXX0 XXXX
Const ON_Bit5   = $20;           // XXXX XXXX OR 0010 0000 = XX1X XXXX
Const OFF_Bit5  = $DF;           // XXXX XXXX AND 1101 1111 = XX0X XXXX
Const ON_Bit6   = $40;           // XXXX XXXX OR 0100 0000 = X1XX XXXX
Const OFF_Bit6  = $BF;           // XXXX XXXX AND 1011 1111 = X0XX XXXX
Const ON_Bit7   = $80;           // XXXX XXXX OR 1000 0000 = 1XXX XXXX
Const OFF_Bit7  = $7F;           // XXXX XXXX AND 0111 1111 = 0XXX XXXX

Const OutputOFF = CIMenu;       // Color of Output OFF ("0") Status
Const OutputON  = CIRed;        // Color of Output ON ("1") Status
Const InputOFF  = Clmenu;       // Color of Input OFF ("1") Status
Const InputON   = CILime;       // Color of Input ON ("0") Status

```

{\$R *.DFM}

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure TForm1.SETUP_8255Click(Sender: TObject);
Var  SetupData : Byte;

Begin
  hET_PCI8255_V3 := nil;
  If Not PCI_Get_WD_handle(@hWD) Then Exit;           // Make sure Driver is loaded //
  WD_Close(hWD);                                     // Close Driver
  hET_PCI8255_V3 := ET_PCI8255_V3_LocateAndOpenBoard(ET_PCI8255_V3_DEFAULT_VENDOR_ID,
                                                    ET_PCI8255_V3_DEFAULT_DEVICE_ID);

  If hET_PCI8255_V3 <> nil Then
  Begin
    // Initial Reset and 8255 Bus Cycle //
    SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        PIB);           // Read PIB Reset Port
    SetupData := SetupData AND OFF_BIT0;             // Bit0 = EXTRST# = "0" (Reset:RES#)
    SetupData := SetupData OR  ON_BIT5;             // Bit5:4 = 11 = PIB Cycle Slowest
    SetupData := SetupData OR  ON_BIT4;
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PIB,
                            SetupData);             // Active RES# & Relay

    // Initial Data (AUX) For CS# and Relay //
    SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        AUXD);         // Read Aux Data Port
    SetupData := SetupData AND OFF_BIT0;             // Bit0 = Aux0 = "0" (Enable CS)
    SetupData := SetupData OR  ON_BIT4;             // Bit4 = Aux4 = "1" (Relay OFF)
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXD,
                            SetupData);             // Active Chips Select & Relay
    LEDRelay.Brush.Color := OutputOFF;             // LED Relay = OFF

    // Initial Direction (AUX) For CS# and Relay //
    SetupData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        AUXC);         // Read Aux Port Direction
    SetupData := SetupData OR ON_BIT4;             // Aux4 = "1" = Output
    SetupData := SetupData OR ON_BIT0;             // Aux0 = "1" = Output
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXC,
                            SetupData);             // Setup Aux Direction
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
// Initial 8255#1 = All Output Port //
Setup_8255.Enabled := False;           // Disable Setup After Setup
RELAY.Enabled := True;                 // Enable Relay Control
Control_PA8255_1.Enabled := True;     // Enable Select Control Port-PA 8255#1
Control_PA8255_1.ItemIndex := 0;     // Default Port-PA = Output

Timer1.Interval := 100;                // Setup Input Scan Time = 100mS
Timer1.Enabled := True;                // Start Auto Scan Input
End;
End;

Procedure TForm1.RelayClick(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     AUXD); // Read Output Latch AUX
  CheckData := InOutData AND ON_BIT4;     // Check Last AUX4 Status
  If CheckData = $10 Then
  Begin
    InOutData := InOutData AND OFF_BIT4;  // AUX4 = "0" (1110 1111)
    LEDRelay.Brush.Color := OutputON;    // LED RELay = ON
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT4;    // AUX4 = "1" (0001 0000)
    LEDRelay.Brush.Color := OutputOFF;   // LED Relay = OFF
  End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), AUXD,
                          InOutData);    // Update AuxD
End;

Procedure TForm1.PA01Click(Sender: TObject);
Var  InOutData : Byte;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1); // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT0;    // Check Last PA0 Status
  If CheckData = $01 Then
    Begin
      InOutData := InOutData AND OFF_BIT0; // PA0 = "0" (1111 1110)
      LEDPA01.Brush.Color := OutputOFF;    // LED PA0 of 8255#1 = OFF
    End
  Else
    Begin
      InOutData := InOutData OR ON_BIT0;   // PA0 = "1" (0000 0001)
      LEDPA01.Brush.Color := OutputON;     // LED PA0 of 8255#1 = ON
    End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA11Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1); // Read Output Latch Port-A
  CheckData := InOutData AND ON_BIT1;    // Check Last PA1 Status
  If CheckData = $02 Then
    Begin
      InOutData := InOutData AND OFF_BIT1; // PA1 = "0" (1111 1101)
      LEDPA11.Brush.Color := OutputOFF;    // LED PA1 of 8255#1 = OFF
    End
  Else
    Begin
      InOutData := InOutData OR ON_BIT1;   // PA1 = "1" (0000 0010)
      LEDPA11.Brush.Color := OutputON;     // LED PA1 of 8255#1 = ON
    End;
End;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
End;
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA21Click(Sender: TObject);
Var InOutData : Byte;
    CheckData : Byte;

Begin
    InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        PA1); // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT2; // Check Last PA2 Status
    If CheckData = $04 Then
    Begin
        InOutData := InOutData AND OFF_BIT2; // PA2 = "0" (1111 1011)
        LEDPA21.Brush.Color := OutputOFF; // LED PA2 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT2; // PA2 = "1" (0000 0100)
        LEDPA21.Brush.Color := OutputON; // LED PA2 of 8255#1 = ON
    End;
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA31Click(Sender: TObject);
Var InOutData : Byte;
    CheckData : Byte;

Begin
    InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        PA1); // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT3; // Check Last PA3 Status
    If CheckData = $08 Then
    Begin
        InOutData := InOutData AND OFF_BIT3; // PA3 = "0" (1111 0111)
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
    LEDPA31.Brush.Color := OutputOFF;           // LED PA3 of 8255#1 = OFF
End
Else
Begin
    InOutData := InOutData OR ON_BIT3;         // PA3 = "1" (0000 1000)
    LEDPA31.Brush.Color := OutputON;          // LED PA3 of 8255#1 = ON
End;
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA41Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
    InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                         PA1); // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT4;      // Check Last PA4 Status
    If CheckData = $10 Then
    Begin
        InOutData := InOutData AND OFF_BIT4; // PA4 = "0" (1110 1111)
        LEDPA41.Brush.Color := OutputOFF;    // LED PA4 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT4;   // PA4 = "1" (0001 0000)
        LEDPA41.Brush.Color := OutputON;     // LED PA4 of 8255#1 = ON
    End;
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA51Click(Sender: TObject);
Var  InOutData : Byte;
     CheckData : Byte;

Begin
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1); // Read Output Latch Port-A
CheckData := InOutData AND ON_BIT5;      // Check Last PA5 Status
If CheckData = $20 Then
Begin
    InOutData := InOutData AND OFF_BIT5; // PA5 = "0" (1101 1111)
    LEDPA51.Brush.Color := OutputOFF;    // LED PA5 of 8255#1 = OFF
End
Else
Begin
    InOutData := InOutData OR ON_BIT5;   // PA5 = "1" (0010 0000)
    LEDPA51.Brush.Color := OutputON;     // LED PA5 of 8255#1 = ON
End;
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.PA61Click(Sender: TObject);
Var InOutData : Byte;
    CheckData : Byte;

Begin
    InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1); // Read Output Latch Port-A
    CheckData := InOutData AND ON_BIT6;   // Check Last PA6 Status
    If CheckData = $40 Then
    Begin
        InOutData := InOutData AND OFF_BIT6; // PA6 = "0" (1011 1111)
        LEDPA61.Brush.Color := OutputOFF;    // LED PA6 of 8255#1 = OFF
    End
    Else
    Begin
        InOutData := InOutData OR ON_BIT6;   // PA6 = "1" (0100 0000)
        LEDPA61.Brush.Color := OutputON;     // LED PA6 of 8255#1 = ON
    End;
    ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Procedure TForm1.PA71Click(Sender: TObject);
Var   InOutData : Byte;
      CheckData : Byte;

Begin
  InOutData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                     PA1); // Read Output Latch Port-A

  CheckData := InOutData AND ON_BIT7;      // Check Last PA7 Status

  If CheckData = $80 Then
  Begin
    InOutData := InOutData AND OFF_BIT7;    // PA7 = "0" (0111 1111)
    LEDPA71.Brush.Color := OutputOFF;      // LED PA7 of 8255#1 = OFF
  End
  Else
  Begin
    InOutData := InOutData OR ON_BIT7;      // PA7 = "1" (1000 0000)
    LEDPA71.Brush.Color := OutputON;       // LED PA7 of 8255#1 = ON
  End;
  ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PA1, InOutData);
End;

Procedure TForm1.Control_PA8255_1Click(Sender: TObject);
Var   ControlWord : Byte;
      InOutData   : Byte;

Begin
  // Control Word Bit7 = 1 = Active 8255 Mode0 //
  ControlWord := $80;                          //Default Control Word 8255 = Mode0 //

  // Port PA(PA7..0) = Bit-D4 in Control Word
  // If Control Word Bit4 = "0" = PA is Output
  // If Control Word Bit4 = "1" = PA is Input
  If Control_PA8255_1.ItemIndex = 0 Then      // If Output Select
  Begin
    ControlWord := ControlWord AND OFF_BIT4; // 1110 1111 => D4 = "0" = PA is Output
    PA01.Enabled := True;                    // Enable PA Output Control
  End;
End;
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
PA11.Enabled := True;
PA21.Enabled := True;
PA31.Enabled := True;
PA41.Enabled := True;
PA51.Enabled := True;
PA61.Enabled := True;
PA71.Enabled := True;

End

Else                                     // If Input Select
Begin
    ControlWord := ControlWord OR ON_BIT4;    // 0001 0000 => D4 = "1" = PA is Input
    PA01.Enabled := False;                    // Disable PA Output Control
    PA11.Enabled := False;
    PA21.Enabled := False;
    PA31.Enabled := False;
    PA41.Enabled := False;
    PA51.Enabled := False;
    PA61.Enabled := False;
    PA71.Enabled := False;
End;

// Start Initial 8255#1 = Control Word //
ET_PCI8255_V3_WriteByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0), PCC1,
                        ControlWord);        // Write Control Port 8255#1

End;

Procedure TForm1.Timer1Timer(Sender: TObject);
Var  InputData : Byte;
     CheckData : Byte;
Begin
    InputData := ET_PCI8255_V3_ReadByte(hET_PCI8255_V3, ET_PCI8255_V3_ADDR (ET_PCI8255_V3_AD_BAR0),
                                        PA1);    // Read Input/Output Latch of PA-8255#1

    If Control_PA8255_1.ItemIndex = 1 Then    // If PA = Input Mode
    Begin
        CheckData := InputData AND ON_BIT0;    // Check PA0 Status

        If CheckData = ON_BIT0 Then
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
LEDPA01.Brush.Color := InputOFF           // PA0 = ("1")
Else LEDPA01.Brush.Color := InputON;      // PA0 = ("0")

CheckData := InputData AND ON_BIT1;      // Check PA1 Status
If CheckData = ON_BIT1 Then
    LEDPA11.Brush.Color := InputOFF       // PA1 = ("1")
Else LEDPA11.Brush.Color := InputON;      // PA1 = ("0")

CheckData := InputData AND ON_BIT2;      // Check PA2 Status
If CheckData = ON_BIT2 Then
    LEDPA21.Brush.Color := InputOFF       // PA2 = ("1")
Else LEDPA21.Brush.Color := InputON;      // PA2 = ("0")

CheckData := InputData AND ON_BIT3;      // Check PA3 Status
If CheckData = ON_BIT3 Then
    LEDPA31.Brush.Color := InputOFF       // PA3 = ("1")
Else LEDPA31.Brush.Color := InputON;      // PA3 = ("0")

CheckData := InputData AND ON_BIT4;      // Check PA4 Status
If CheckData = ON_BIT4 Then
    LEDPA41.Brush.Color := InputOFF       // PA4 = ("1")
Else LEDPA41.Brush.Color := InputON;      // PA4 = ("0")

CheckData := InputData AND ON_BIT5;      // Check PA5 Status
If CheckData = ON_BIT5 Then
    LEDPA51.Brush.Color := InputOFF       // PA5 = ("1")
Else LEDPA51.Brush.Color := InputON;      // PA5 = ("0")

CheckData := InputData AND ON_BIT6;      // Check PA6 Status
If CheckData = ON_BIT6 Then
    LEDPA61.Brush.Color := InputOFF       // PA6 = ("1")
Else LEDPA61.Brush.Color := InputON;      // PA6 = ("0")

CheckData := InputData AND ON_BIT7;      // Check PA7 Status
If CheckData = ON_BIT7 Then
    LEDPA71.Brush.Color := InputOFF       // PA7 = ("1")
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
Else LEDPA71.Brush.Color := InputON;           // PA7 = ("0")
End
Else                                           // If PA = Output Mode
Begin
    CheckData := InputData AND ON_BIT0;       // Check PA0 Status
    If CheckData = ON_BIT0 Then
        LEDPA01.Brush.Color := OutputON       // PA0 = ("1")
    Else LEDPA01.Brush.Color := OutputOFF;     // PA0 = ("0")

    CheckData := InputData AND ON_BIT1;       // Check PA1 Status
    If CheckData = ON_BIT1 Then
        LEDPA11.Brush.Color := OutputON       // PA1 = ("1")
    Else LEDPA11.Brush.Color := OutputOFF;     // PA1 = ("0")

    CheckData := InputData AND ON_BIT2;       // Check PA2 Status
    If CheckData = ON_BIT2 Then
        LEDPA21.Brush.Color := OutputON       // PA2 = ("1")
    Else LEDPA21.Brush.Color := OutputOFF;     // PA2 = ("0")

    CheckData := InputData AND ON_BIT3;       // Check PA3 Status
    If CheckData = ON_BIT3 Then
        LEDPA31.Brush.Color := OutputON       // PA3 = ("1")
    Else LEDPA31.Brush.Color := OutputOFF;     // PA3 = ("0")

    CheckData := InputData AND ON_BIT4;       // Check PA4 Status
    If CheckData = ON_BIT4 Then
        LEDPA41.Brush.Color := OutputON       // PA4 = ("1")
    Else LEDPA41.Brush.Color := OutputOFF;     // PA4 = ("0")

    CheckData := InputData AND ON_BIT5;       // Check PA5 Status
    If CheckData = ON_BIT5 Then
        LEDPA51.Brush.Color := OutputON       // PA5 = ("1")
    Else LEDPA51.Brush.Color := OutputOFF;     // PA5 = ("0")

    CheckData := InputData AND ON_BIT6;       // Check PA6 Status
    If CheckData = ON_BIT6 Then
```

ตัวอย่าง การพัฒนาโปรแกรมของการ์ด ET-PCI8255 V3 ด้วย Delphi 5

```
LEDPA61.Brush.Color := OutputON           // PA6 = ("1")
Else LEDPA61.Brush.Color := OutputOFF;    // PA6 = ("0")

CheckData := InputData AND ON_BIT7;      // Check PA7 Status
If CheckData = ON_BIT7 Then
    LEDPA71.Brush.Color := OutputON       // PA7 = ("1")
Else LEDPA71.Brush.Color := OutputOFF;    // PA7 = ("0")
End;
End;

Procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
Begin
    If hET_PCI8255_V3 <> nil Then ET_PCI8255_V3_Close(hET_PCI8255_V3); //Close Driver Before Exit Program //
End;

End.
```