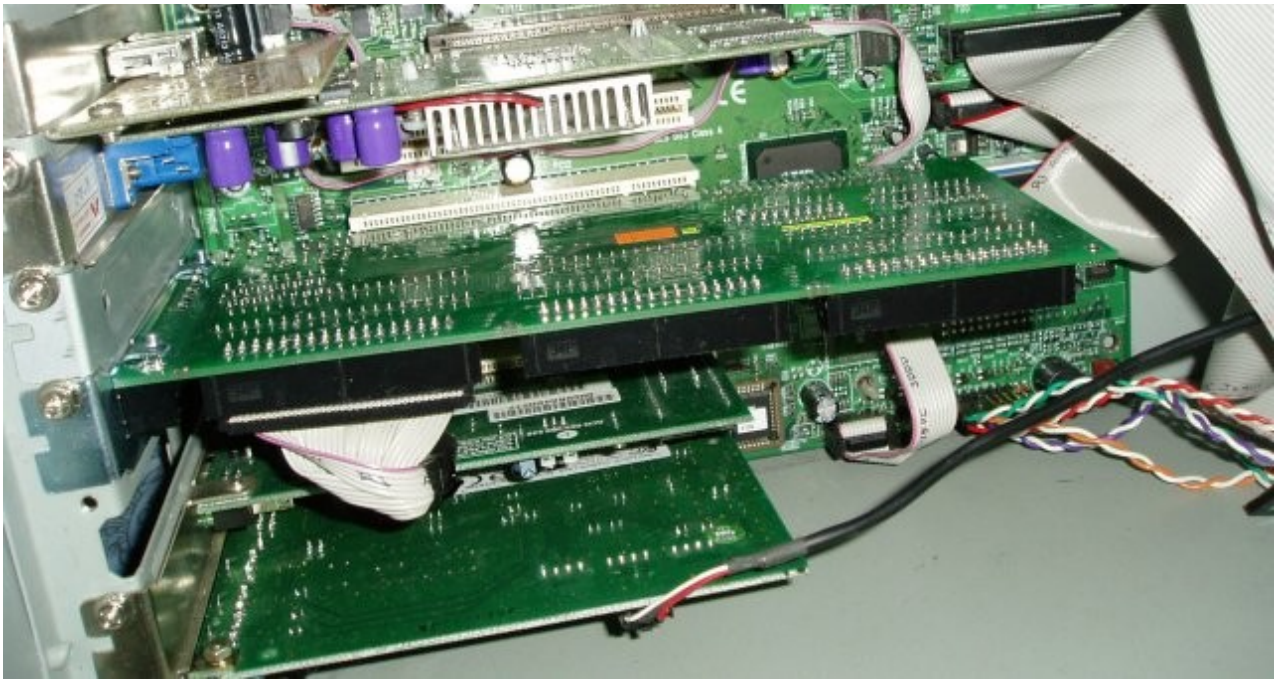
ตัวอย่างส่งข้อมูลขับแอลอีดีจาก ET-PCI8255 V3 ด้วย Visual C++ 6

โดย ศุภชัย บุศราทิจ
คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเพชรบุรี

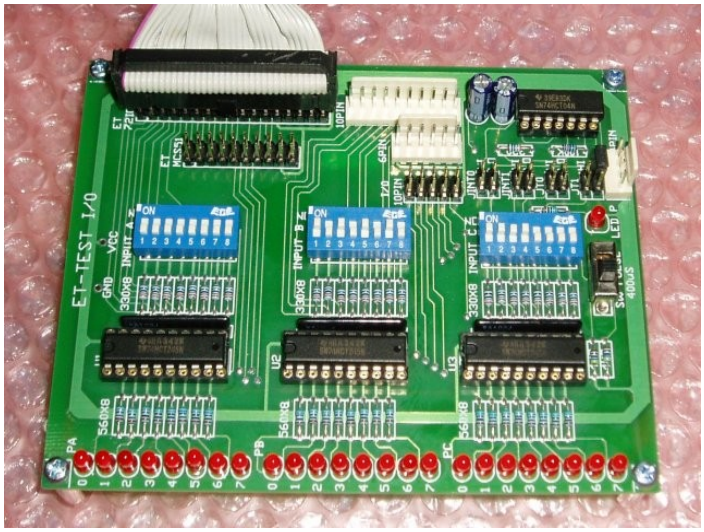
สวัสดีครับ หลังจากหายหน้าหายตาไปหลายเดือน พอดีช่วงนี้พอมีเวลาสำหรับเขียนบทความเลยนำตัวอย่างที่ได้ทำเอาไว้มาเรียบเรียงคำอธิบายคร่าวๆ เพื่อเป็นแนวทางสำหรับการนำไปพัฒนาโปรแกรมของผู้ที่ชอบเขียนด้วย Visual C++ 6 โดยใช้คลาสพื้นฐานของไมโครซอฟต์ที่เรียกกับว่า**เอ็มเอฟซี**(MFC/Microsoft Foundation Class)ในการพัฒนา ทั้งนี้เนื่องจากการ์ด ET-PCI8255 V3 ของอีทีทีนั้นมีตัวอย่างที่เป็นของ Visual BASIC และ Delphi เท่านั้น

เพื่อไม่ใช้ซ้ำซ้อนกับคำอธิบายที่มีอยู่แล้วในเอกสารของการ์ดผมจะขอกล่าวถึงเฉพาะรายละเอียดที่เกี่ยวข้องกับ Visual C++ เท่านั้น ส่วนรายละเอียดเรื่องวงจร/วิธีการติดตั้งและวิธีดูว่าการ์ดของเรานั้นติดตั้งในตำแหน่งใดของหน่วยความจำนั้นขอให้ดูจากเอกสารของการ์ด



รูปที่ 1 การ์ด ET-PCI8255v3 ที่ติดตั้งแล้ว

รูปที่2 การ์ด ET-PCI8255v3 ที่เชื่อมต่อกับ ET-Test I/IO



รูปที่ 3 การ์ด ET-Test I/O

อุปกรณ์ประกอบการทดลอง

1. การ์ด ET-PCI8255v3

2. บอร์ด ET-TEST I/O

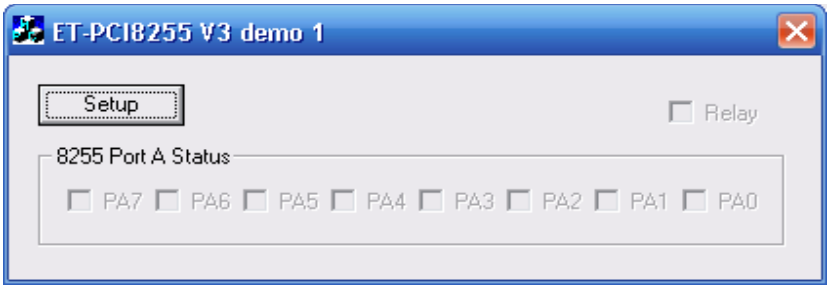
3. โปรแกรม Visual C++ 6 Professional (ถูกลิขสิทธิ์ครับ ^_^)

4. ระบบปฏิบัติการ Microsoft Windows 2000 Professional (ถูกลิขสิทธิ์ครับ ^_^) แต่ภาพที่จับนั้นเป็นการรันโปรแกรมบนระบบปฏิบัติการ Microsoft Windows XP Home Edition บนเครื่อง IBM Think Pad

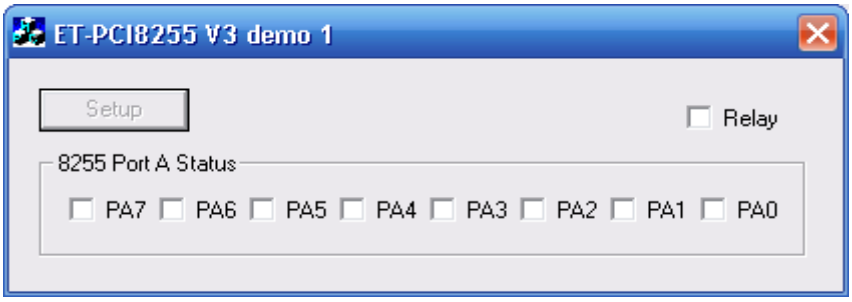
5. ฮาร์ดแวร์ Intel Pentium 3 (800MHz) หน่วยความจำ 1GB ฮาร์ดดิสก์ 40GB (Maxtor) การ์ดแสดงผล GeForce 4 MX 440/64 MB

ตัวอย่างโปรแกรมที่ 1

การขับหลอดแอลอีดีผ่านทาง 8255 ตัวที่ 1 ของการ์ด ET-PCI8255 V3 จะมีหน้าตาการทำงานของโปรแกรมดังรูปที่ 4 เมื่อเราคลิกที่ปุ่ม Setup จะเป็นการเริ่มโปรแกรม(รูปที่ 5) ซึ่งในขั้นตอนนี้ผู้ใช้สามารถกำหนดให้แอลอีดีติดหรือดับได้ด้วยการคลิกที่ช่อง PA0 ถึง PA7 หรือถ้าต้องการควบคุมรีเลย์(Relay) ก็สามารถทำได้เช่นเดียวกันด้วยคลิกที่ช่องของ Relay เพื่อสั่งให้เปิดและปิดการทำงาน และเมื่อคลิกที่ปุ่มปิดโปรแกรมจะเป็นการตัดการเชื่อมต่อกับการ์ด ET-PCI8255v3



รูปที่ 4 หน้าจอเมื่อเริ่มโปรแกรม



รูปที่ 5 หน้าจอเมื่อเริ่มโปรแกรม

จากการทำงานดังกล่าวสามารถเขียนเป็นโค้ดภาษาซีพลัสพลัส(C++) แยกเป็นแฟ้มได้ดังนี้

1. แฟ้มส่วนหัว(inout32.h)สำหรับเรียกใช้ inout32.dll

```
short _stdcall Inp32(short PortAddress);
void _stdcall Out32(short PortAddress, short data);
```

2. แฟ้มส่วนหัว(et8255v3dem1Dlg.h)สำหรับไดอะลอก(Dialog)ที่แสดงในรูปที่ 4 และ 5

```
// et8255v3dem1Dlg.h : header file
//

#if !
defined(AFX_ET8255V3DEM1DLG_H__549CCF61_3B01_4C28_9254_C04FCB6E4787__INCLUDED_)
#define AFX_ET8255V3DEM1DLG_H__549CCF61_3B01_4C28_9254_C04FCB6E4787__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

/////////////////////////////////////////////////////////////////////////////
// CEt8255v3dem1Dlg dialog

class CEt8255v3dem1Dlg : public CDialog
{
// Construction
public:
	CEt8255v3dem1Dlg(CWnd* pParent = NULL);   // standard constructor

// Dialog Data
	//{{AFX_DATA(CEt8255v3dem1Dlg)
	enum { IDD = IDD_ET8255V3DEM1_DIALOG };
	CButton	m_setup;
	CButton	m_relay;
	CButton	m_pa7;
	CButton	m_pa6;
	CButton	m_pa5;
	CButton	m_pa4;
	CButton	m_pa3;
	CButton	m_pa2;
	CButton	m_pa1;
	CButton	m_pa0;
	//}}AFX_DATA

	// ClassWizard generated virtual function overrides
	//{{AFX_VIRTUAL(CEt8255v3dem1Dlg)
	protected:
	virtual void DoDataExchange(CDataExchange* pDX);	// DDX/DDV support
	//}}AFX_VIRTUAL

// Implementation
```

```
protected:
	HICON m_hIcon;

//////// For ET-PCI8255 V3
	SHORT BaseAddress;
	BYTE  InOutData, CheckData;
	SHORT IO_BaseAddress;
	// Tiger-320 Register offset
	BYTE PIB,  // Reset & PIB cycle
		AUXC, // AUX Direction port
		AUXD, // AUX Data port
		PA1, PB1, PC1, PCC1, // 8255 Port A/B/C/Control #1
		PA2, PB2, PC2, PCC2, // 8255 Port A/B/C/Control #2
		PA3, PB3, PC3, PCC3, // 8255 Port A/B/C/Control #3
		ON_Bit0,  // XXXX XXXX OR 0000 0001 = XXXX XXX1
		OFF_Bit0, // XXXX XXXX AND 1111 1110 = XXXX XXX0
		ON_Bit1,  // XXXX XXXX OR 0000 0010 = XXXX XX1X
		OFF_Bit1, // XXXX XXXX AND 1111 1101 = XXXX XX0X
		ON_Bit2,  // XXXX XXXX OR 0000 0100 = XXXX X1XX
		OFF_Bit2, // XXXX XXXX AND 1111 1011 = XXXX X0XX
		ON_Bit3,  // XXXX XXXX OR 0000 1000 = XXXX 1XXX
		OFF_Bit3, // XXXX XXXX AND 1111 0111 = XXXX 0XXX
		ON_Bit4,  // XXXX XXXX OR 0001 0000 = XXX1 XXXX
		OFF_Bit4, // XXXX XXXX AND 1110 1111 = XXX0 XXXX
		ON_Bit5,  // XXXX XXXX OR 0010 0000 = XX1X XXXX
		OFF_Bit5, // XXXX XXXX AND 1101 1111 = XX0X XXXX
		ON_Bit6,  // XXXX XXXX OR 0100 0000 = X1XX XXXX
		OFF_Bit6, // XXXX XXXX AND 1011 1111 = X0XX XXXX
		ON_Bit7,  // XXXX XXXX OR 1000 0000 = 1XXX XXXX
		OFF_Bit7; // XXXX XXXX AND 0111 1111 = 0XXX XXXX

	// Generated message map functions
	//{{AFX_MSG(CEt8255v3dem1Dlg)
	virtual BOOL OnInitDialog();
	afx_msg void OnPaint();
	afx_msg HCURSOR OnQueryDragIcon();
	afx_msg void OnBtnSetup();
	afx_msg void OnChkPa0();
	afx_msg void OnChkPa1();
	afx_msg void OnChkPa2();
	afx_msg void OnChkPa3();
	afx_msg void OnChkPa4();
	afx_msg void OnChkPa5();
	afx_msg void OnChkPa6();
	afx_msg void OnChkPa7();
	afx_msg void OnChkRelay();
```

```
	//}}AFX_MSG
	DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before
the previous line.

#endif // !
defined(AFX_ET8255V3DEM1DLG_H__549CCF61_3B01_4C28_9254_C04FCB6E4787__INCLUDED_)
```

3. แฟ้มหลัก()สำหรับควบคุมการทำงานของโปรแกรม

```
// et8255v3dem1Dlg.cpp : implementation file
//

#include "stdafx.h"
#include "et8255v3dem1.h"
#include "et8255v3dem1Dlg.h"

#include "inpout32.h"


#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CEt8255v3dem1Dlg dialog

CEt8255v3dem1Dlg::CEt8255v3dem1Dlg(CWnd* pParent /*=NULL*/)
	: CDialog(CEt8255v3dem1Dlg::IDD, pParent)
{
	//{{AFX_DATA_INIT(CEt8255v3dem1Dlg)
		// NOTE: the ClassWizard will add member initialization here
	//}}AFX_DATA_INIT
	// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CEt8255v3dem1Dlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
```

```
	//{{AFX_DATA_MAP(CEt8255v3dem1Dlg)
	DDX_Control(pDX, IDC_BTN_SETUP, m_setup);
	DDX_Control(pDX, IDC_CHK_RELAY, m_relay);
	DDX_Control(pDX, IDC_CHK_PA7, m_pa7);
	DDX_Control(pDX, IDC_CHK_PA6, m_pa6);
	DDX_Control(pDX, IDC_CHK_PA5, m_pa5);
	DDX_Control(pDX, IDC_CHK_PA4, m_pa4);
	DDX_Control(pDX, IDC_CHK_PA3, m_pa3);
	DDX_Control(pDX, IDC_CHK_PA2, m_pa2);
	DDX_Control(pDX, IDC_CHK_PA1, m_pa1);
	DDX_Control(pDX, IDC_CHK_PA0, m_pa0);
	//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CEt8255v3dem1Dlg, CDialog)
	//{{AFX_MSG_MAP(CEt8255v3dem1Dlg)
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_BN_CLICKED(IDC_BTN_SETUP, OnBtnSetup)
	ON_BN_CLICKED(IDC_CHK_PA0, OnChkPa0)
	ON_BN_CLICKED(IDC_CHK_PA1, OnChkPa1)
	ON_BN_CLICKED(IDC_CHK_PA2, OnChkPa2)
	ON_BN_CLICKED(IDC_CHK_PA3, OnChkPa3)
	ON_BN_CLICKED(IDC_CHK_PA4, OnChkPa4)
	ON_BN_CLICKED(IDC_CHK_PA5, OnChkPa5)
	ON_BN_CLICKED(IDC_CHK_PA6, OnChkPa6)
	ON_BN_CLICKED(IDC_CHK_PA7, OnChkPa7)
	ON_BN_CLICKED(IDC_CHK_RELAY, OnChkRelay)
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CEt8255v3dem1Dlg message handlers

BOOL CEt8255v3dem1Dlg::OnInitDialog()
{
	CDialog::OnInitDialog();

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

	// TODO: Add extra initialization here
	IO_BaseAddress = 0xE800; // I/O Base address for ET-PCI8255 V3
    PIB      = 0x00; // Reset & PIB cycle
```

```
    AUXC     = 0x02; // AUX Direction port
      AUXD     = 0x03; // AUX Data port
      PA1      = 0xC0;
      PB1      = 0xC4;
      PC1      = 0xC8;
      PCC1     = 0xCC; // 8255 Port A/B/C/Control #1
      PA2      = 0xD0;
      PB2      = 0xD4;
      PC2      = 0xD8;
      PCC2     = 0xDC; // 8255 Port A/B/C/Control #2
      PA3      = 0xE0;
      PB3      = 0xE4;
      PC3      = 0xE8;
      PCC3     = 0xEC; // 8255 Port A/B/C/Control #3
      ON_Bit0  = 0x01; // XXXX XXXX OR 0000 0001 = XXXX XXX1
      OFF_Bit0 = 0xFE; // XXXX XXXX AND 1111 1110 = XXXX XXX0
      ON_Bit1  = 0x02; // XXXX XXXX OR 0000 0010 = XXXX XX1X
      OFF_Bit1 = 0xFD; // XXXX XXXX AND 1111 1101 = XXXX XX0X
      ON_Bit2  = 0x04; // XXXX XXXX OR 0000 0100 = XXXX X1XX
      OFF_Bit2 = 0xFB; // XXXX XXXX AND 1111 1011 = XXXX X0XX
      ON_Bit3  = 0x08; // XXXX XXXX OR 0000 1000 = XXXX 1XXX
      OFF_Bit3 = 0xF7; // XXXX XXXX AND 1111 0111 = XXXX 0XXX
      ON_Bit4  = 0x10; // XXXX XXXX OR 0001 0000 = XXX1 XXXX
      OFF_Bit4 = 0xEF; // XXXX XXXX AND 1110 1111 = XXX0 XXXX
      ON_Bit5  = 0x20; // XXXX XXXX OR 0010 0000 = XX1X XXXX
      OFF_Bit5 = 0xDF; // XXXX XXXX AND 1101 1111 = XX0X XXXX
      ON_Bit6  = 0x40; // XXXX XXXX OR 0100 0000 = X1XX XXXX
      OFF_Bit6 = 0xBF; // XXXX XXXX AND 1011 1111 = X0XX XXXX
      ON_Bit7  = 0x80; // XXXX XXXX OR 1000 0000 = 1XXX XXXX
      OFF_Bit7 = 0x7F; // XXXX XXXX AND 0111 1111 = 0XXX XXXX

      m_setup.EnableWindow(true);
      m_relay.EnableWindow(false);
      m_pa0.EnableWindow(false);
      m_pa1.EnableWindow(false);
      m_pa2.EnableWindow(false);
      m_pa3.EnableWindow(false);
      m_pa4.EnableWindow(false);
      m_pa5.EnableWindow(false);
      m_pa6.EnableWindow(false);
      m_pa7.EnableWindow(false);


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

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

void CEt8255v3dem1Dlg::OnPaint()
{
	if (IsIconic())
	{
		CPaintDC dc(this); // device context for painting

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

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

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

// The system calls this to obtain the cursor to display while the user drags
//  the minimized window.
HCURSOR CEt8255v3dem1Dlg::OnQueryDragIcon()
{
	return (HCURSOR) m_hIcon;
}

void CEt8255v3dem1Dlg::OnBtnSetup()
{
	// TODO: Add your control notification handler code here
	SHORT SetupData;

//	AfxMessageBox("คุณคลิกที่ปุ่ม Setup");

	BaseAddress = IO_BaseAddress;
	// Initial Reset and 8255 Bus Cycle
	SetupData = Inp32(BaseAddress+PIB); // Read PIB Reset Port
```

```
	SetupData = SetupData & OFF_Bit0;	// Bit0 = EXTRST# = "0" (Reset:RES#)
	SetupData = SetupData | ON_Bit5;	// Bit5:4 = 11 = PIB Cycle Slowest
	SetupData = SetupData | 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 & OFF_Bit0; // Bit0 = Aux0 = "0" (Enable CS)
	SetupData = SetupData | ON_Bit4; // Bit4 = Aux4 = "1" (Relay OFF)
	Out32(BaseAddress+AUXD,SetupData); // Active Chips Select & Relay
	m_relay.SetCheck(0);

	// Initial Direction (AUX) For CS# and Relay //
	SetupData = Inp32(BaseAddress+AUXC); // Read Aux Port Direction
	SetupData = SetupData | ON_Bit4; // Aux4 = "1" = Output
	SetupData = SetupData | ON_Bit0; // Aux0 = "1" = Output
	Out32(BaseAddress+AUXC,SetupData); // Setup Aux Direction

	// Initial 8255#1 = All Output Port //
	m_setup.EnableWindow(false);
	Out32(BaseAddress+PCC1,0x80); // Write Control Port 8255#1
	m_relay.EnableWindow(true);
	m_pa0.EnableWindow(true);
	m_pa1.EnableWindow(true);
	m_pa2.EnableWindow(true);
	m_pa3.EnableWindow(true);
	m_pa4.EnableWindow(true);
	m_pa5.EnableWindow(true);
	m_pa6.EnableWindow(true);
	m_pa7.EnableWindow(true);
}

void CEt8255v3dem1Dlg::OnChkPa0()
{
	// TODO: Add your control notification handler code here
//	AfxMessageBox("คุณคลิกที่ปุ่ม PA0");
	InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
	CheckData = InOutData & ON_Bit0; // Check Last PA0 Status
	if (CheckData == 0x01) {
		InOutData = InOutData & OFF_Bit0; // PA0 = "0" (1111 1110)
		m_pa0.SetCheck(0);
	} else {
		InOutData = InOutData | ON_Bit0; // PA0 = "1" (0000 0001)
		m_pa0.SetCheck(1);
	}
	Out32(BaseAddress+PA1,InOutData); // Update Port-A
```

```
}

void CEt8255v3dem1Dlg::OnChkPa1()
{
	// TODO: Add your control notification handler code here
//	AfxMessageBox("คุณคลิกที่ปุ่ม PA1");
	InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
	CheckData = InOutData & ON_Bit1; // Check Last PA1 Status
	if (CheckData == 0x02) {
		InOutData = InOutData & OFF_Bit1; // PA1 = "0" (1111 1101)
		m_pa1.SetCheck(0);
	} else {
		InOutData = InOutData | ON_Bit1; // PA1 = "1" (0000 0010)
		m_pa1.SetCheck(1);
	}
	Out32(BaseAddress+PA1,InOutData); // Update Port-A
}

void CEt8255v3dem1Dlg::OnChkPa2()
{
	// TODO: Add your control notification handler code here
//	AfxMessageBox("คุณคลิกที่ปุ่ม PA2");
	InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
	CheckData = InOutData & ON_Bit2; // Check Last PA2 Status
	if (CheckData == 0x04) {
		InOutData = InOutData & OFF_Bit2; // PA2 = "0" (1111 1011)
		m_pa2.SetCheck(0);
	} else {
		InOutData = InOutData | ON_Bit2; // PA2 = "1" (0000 0100)
		m_pa2.SetCheck(1);
	}
	Out32(BaseAddress+PA1,InOutData); // Update Port-A
}

void CEt8255v3dem1Dlg::OnChkPa3()
{
	// TODO: Add your control notification handler code here
//	AfxMessageBox("คุณคลิกที่ปุ่ม PA3");
	InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
	CheckData = InOutData & ON_Bit3; // Check Last PA3 Status
	if (CheckData == 0x08) {
		InOutData = InOutData & OFF_Bit3; // PA3 = "0" (1111 0111)
		m_pa3.SetCheck(0);
	} else {
		InOutData = InOutData | ON_Bit3; // PA3 = "1" (0000 1000)
		m_pa3.SetCheck(1);
```

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

void CEt8255v3dem1Dlg::OnChkPa4()
{
	// TODO: Add your control notification handler code here
//	AfxMessageBox("คุณคลิกที่ปุ่ม PA4");
	InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
	CheckData = InOutData & ON_Bit4; // Check Last PA4 Status
	if (CheckData == 0x10) {
		InOutData = InOutData & OFF_Bit4; // PA4 = "0" (1110 1111)
		m_pa4.SetCheck(0);
	} else {
		InOutData = InOutData | ON_Bit4; // PA4 = "1" (0001 0000)
		m_pa4.SetCheck(1);
	}
	Out32(BaseAddress+PA1,InOutData); // Update Port-A
}

void CEt8255v3dem1Dlg::OnChkPa5()
{
	// TODO: Add your control notification handler code here
//  AfxMessageBox("คุณคลิกที่ปุ่ม PA5");
	InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
	CheckData = InOutData & ON_Bit5; // Check Last PA5 Status
	if (CheckData == 0x20) {
		InOutData = InOutData & OFF_Bit5; // PA5 = "0" (1101 1111)
		m_pa5.SetCheck(0);
	} else {
		InOutData = InOutData | ON_Bit5; // PA5 = "1" (0010 0000)
		m_pa5.SetCheck(1);
	}
	Out32(BaseAddress+PA1,InOutData); // Update Port-A
}

void CEt8255v3dem1Dlg::OnChkPa6()
{
	// TODO: Add your control notification handler code here
//	AfxMessageBox("คุณคลิกที่ปุ่ม PA6");
	InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
	CheckData = InOutData & ON_Bit6; // Check Last PA6 Status
	if (CheckData == 0x40) {
		InOutData = InOutData & OFF_Bit6; // PA6 = "0" (1011 1111)
		m_pa6.SetCheck(0);
	} else {
```

```
            InOutData = InOutData | ON_Bit6; // PA6 = "1" (0100 0000)
            m_pa6.SetCheck(1);
      }
      Out32(BaseAddress+PA1,InOutData); // Update Port-A
}

void CEt8255v3dem1Dlg::OnChkPa7()
{
      // TODO: Add your control notification handler code here
//    AfxMessageBox("คุณคลิกที่ปุ่ม PA7");
      InOutData = Inp32(BaseAddress+PA1); // Read Output Latch Port-A
      CheckData = InOutData & ON_Bit7; // Check Last PA7 Status
      if (CheckData == 0x80) {
            InOutData = InOutData & OFF_Bit7; // PA7 = "0" (0111 1111)
            m_pa7.SetCheck(0);
      } else {
            InOutData = InOutData | ON_Bit7; // PA7 = "1" (1000 0000)
            m_pa7.SetCheck(1);
      }
      Out32(BaseAddress+PA1,InOutData); // Update Port-A
}

void CEt8255v3dem1Dlg::OnChkRelay()
{
      // TODO: Add your control notification handler code here
//    AfxMessageBox("คุณคลิกที่ปุ่ม RELAY");
      InOutData = Inp32(BaseAddress+AUXD); // Read Output Latch AUX
      CheckData = InOutData & ON_Bit4; // Check Last AUX4 Status
      if (CheckData == 0x10) {
            InOutData = InOutData & OFF_Bit4; // AUX4 = "0" (1110 1111)
            m_relay.SetCheck(1);
      } else {
            InOutData = InOutData | ON_Bit4; // AUX4 = "1" (0001 0000)
            m_relay.SetCheck(0);
      }
      Out32(BaseAddress+AUXD,InOutData); // Update AuxD
}
```

หมายเหตุ

ถ้าโปรแกรมทำงานแต่ไม่แสดงอะไรที่บอร์ดแอลอีดี ขอให้ดูที่ฟังก์ชัน `::OnInitDialog()` แล้วเปลี่ยนแปลงค่าของ IO_BaseAddress ให้ถูกต้อง (ดูจากคู่มือการ์ดประกอบ)

สรุป

จากตัวอย่างโปรแกรมแรกนี้ถ้าได้นำไปเปรียบเทียบกับโค้ดที่เขียนด้วยเดลไฟ(Delphi) จะพบว่าเหมือนกันมาก ทั้งนี้เนื่องจากผมทำการแปลงโค้ดจากภาษาปาสกาลมาเป็นซีพลัสพลัส โดยให้การสั่งงานนั้นเหมือนเดิม เพื่อเป็นแนวทางสำหรับคนที่สนใจศึกษาการแปลงโค้ดระหว่างเดลไฟและวิชวลซีพลัสพลัส ครั้งต่อไปจะเป็นตัวอย่างโปรแกรมที่ 2 ซึ่งจะทำงานโดยอาศัยตัวตั้งเวลา(Timer) ในการควบคุมการส่งข้อมูลไปที่พอร์ต

สุดท้ายนี้ขอขอบคุณคุณกอบกิจ เติมผาติ และทีมงานอีทีทีเป็นอย่างสูงที่ได้ส่งอุปกรณ์ต่างๆ มาให้ทดลองจนออกมาเป็นบทความนี้ (น่าเสียดายที่ช่วงนี้เวลามันเร่งรีบไปซะทุกเรื่องไม่อย่างนั้นคงได้นำโค้ดโปรแกรมมาลงเป็นบทความได้อีกหลายตอน) พบกันครั้งหน้าครับ (ตอนนี้ลูก 5 เดือนแล้ว เย้..เหนื่อยจัง ฮ่ะๆๆ)