

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

แผนปฏิบัติการสอนสัปดาห์ที่ 7

วิชา 030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม ระดับ: ปริญญาตรี

เรื่อง รับข้อมูลต่าง ๆ จากอุปกรณ์ Input

เวลา: บรรยาย 120 นาที

ปฏิบัติ 120 นาที

ก. วัตถุประสงค์การสอน	รายละเอียดตามที่ระบุไว้ใน
1. รับค่าจาก Keyboard และ Mouse 2. รับค่าจาก Joy-Stick 3. รับค่าจาก Barcode Reader 4. ถ่ายภาพ และ Video จากกล้อง Web camera 5. บันทึกภาพเป็น File ต่างๆ	} หน้า 91-104

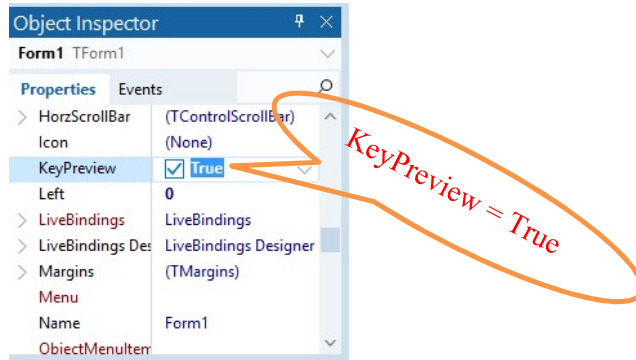
ข. การจัดการเรียนการสอน

เวลา - นาที	0	60	120	180	240
วัตถุประสงค์	1, 2, 3, 4, 5				
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
พัก					
ทำแบบฝึกหัด					
ให้เนื้อหา					
สรุปเนื้อหา					
ทดสอบและเก็บคะแนน					
ประเมินผล	พิจารณาจากผลการทดสอบในช่วงโม่งสุดท้าย				
วิธีการสอน:	บรรยาย				
	ถาม - ตอบ				
	ทำแบบฝึกหัด				
	บรรยาย และแสดงให้ดู				
	ทดสอบ				
สื่อการสอน:	คอมพิวเตอร์				
	Presentations				

Content

1. รับค่าจาก Keyboard และ Mouse

การรับค่าจาก Keyboard จะต้องตั้งค่า Properties ของ Form KeyPreview:=True; ก่อนเสมอถึงจะสามารถดักจับ Key ที่กด และสร้าง Event ได้



Event ที่เกี่ยวข้องกับการกด Key

```
procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);  
...  
procedure TForm1.FormKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);  
...  
procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
```

```
if Key in ['a'..'z'] + ['A'..'Z'] then Key := #0
```

Key code in delphi

Symbolic

constant name Value

(hexadecimal) Keyboard (or mouse) equivalent

VK_LBUTTON	01 Left mouse button
VK_RBUTTON	02 Right mouse button
VK_CANCEL	03 Control-break processing
VK_MBUTTON	04 Middle mouse button (three-button mouse)
VK_BACK	08 BACKSPACE key
VK_TAB	09 TAB key
VK_CLEAR	0C CLEAR key
VK_RETURN	0D ENTER key
VK_SHIFT	10 SHIFT key

VK_CONTROL	11 CTRL key
VK_MENU	12 ALT key
VK_PAUSE	13 PAUSE key
VK_CAPITAL	14 CAPS LOCK key
VK_ESCAPE	1B ESC key
VK_SPACE	20 SPACEBAR
VK_PRIOR	21 PAGE UP key
VK_NEXT	22 PAGE DOWN key
VK_END	23 END key
VK_HOME	24 HOME key
VK_LEFT	25 LEFT ARROW key
VK_UP	26 UP ARROW key
VK_RIGHT	27 RIGHT ARROW key
VK_DOWN	28 DOWN ARROW key

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

VK_SELECT	29 SELECT key	55	U key
VK_PRINT	2A PRINT key	56	V key
VK_EXECUTE	2B EXECUTE key	57	W key
VK_SNAPSHOT	2C PRINT SCREEN key	58	X key
VK_INSERT	2D INS key	59	Y key
VK_DELETE	2E DEL key	5A	Z key
VK_HELP	2F HELP key	VK_NUMPAD0	60 Numeric keypad 0 key
30	0 key	VK_NUMPAD1	61 Numeric keypad 1 key
31	1 key	VK_NUMPAD2	62 Numeric keypad 2 key
32	2 key	VK_NUMPAD3	63 Numeric keypad 3 key
33	3 key	VK_NUMPAD4	64 Numeric keypad 4 key
34	4 key	VK_NUMPAD5	65 Numeric keypad 5 key
35	5 key	VK_NUMPAD6	66 Numeric keypad 6 key
36	6 key	VK_NUMPAD7	67 Numeric keypad 7 key
37	7 key	VK_NUMPAD8	68 Numeric keypad 8 key
38	8 key	VK_NUMPAD9	69 Numeric keypad 9 key
39	9 key	VK_SEPARATOR	6C Separator key
41	A key	VK_SUBTRACT	6D Subtract key
42	B key	VK_DECIMAL	6E Decimal key
43	C key	VK_DIVIDE	6F Divide key
44	D key	VK_F1	70 F1 key
45	E key	VK_F2	71 F2 key
46	F key	VK_F3	72 F3 key
47	G key	VK_F4	73 F4 key
48	H key	VK_F5	74 F5 key
49	I key	VK_F6	75 F6 key
4A	J key	VK_F7	76 F7 key
4B	K key	VK_F8	77 F8 key
4C	L key	VK_F9	78 F9 key
4D	M key	VK_F10	79 F10 key
4E	N key	VK_F11	7A F11 key
4F	O key	VK_F12	7B F12 key
50	P key	VK_F13	7C F13 key
51	Q key	VK_F14	7D F14 key
52	R key	VK_F15	7E F15 key
53	S key	VK_F16	7F F16 key
54	T key	VK_F17	80H F17 key

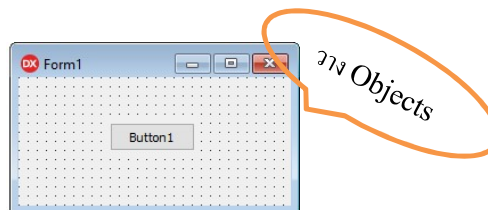
030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

VK_F18	81H F18 key
VK_F19	82H F19 key
VK_F20	83H F20 key
VK_F21	84H F21 key
VK_F22	85H F22 key
VK_F23	86H F23 key
VK_F24	87H F24 key
VK_NUMLOCK	90 NUM LOCK key
VK_SCROLL	91 SCROLL LOCK key

VK_LSHIFT	A0 Left SHIFT key
VK_RSHIFT	A1 Right SHIFT key
VK_LCONTROL	A2 Left CONTROL key
VK_RCONTROL	A3 Right CONTROL key
VK_LMENU	A4 Left MENU key
VK_RMENU	A5 Right MENU key
VK_PLAY	FA Play key
VK_ZOOM	FB Zoom key

<https://www.thoughtco.com/understanding-keyboard-events-in-delphi-1058213>

Example



Code

```
unit Unit1;

interface

uses

  Winapi.Windows, Winapi.Messages, System.SysUtils,
  System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

  TForm1 = class(TForm)
    Button1: TButton;
    procedure FormKeyDown(Sender: TObject; var Key:
    Word; Shift: TShiftState);
    procedure FormKeyUp(Sender: TObject; var Key:
    Word; Shift: TShiftState);
    procedure FormKeyPress(Sender: TObject; var Key:
    Char);
    procedure FormMouseDown(Sender: TObject; Button:
    TMouseButton;
    Shift: TShiftState; X, Y: Integer);
```

```
    procedure FormMouseMove(Sender: TObject; Shift:
    TShiftState; X, Y: Integer);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.FormKeyDown(Sender: TObject; var
  Key: Word;
  Shift: TShiftState);
begin
```

```
if Key = VK_F1 then
begin
  if ssCtrl in Shift then
    Button1.Top:=Button1.Top+10
  else
    Button1.Top:=Button1.Top-10
end;
end;

procedure TForm1.FormKeyPress(Sender: TObject; var
Key: Char);
begin
  if Key = 'i' then
    Key := '5';

  Form1.Caption:=Key;
end;

procedure TForm1.FormKeyUp(Sender: TObject; var Key:
Word; Shift: TShiftState);
begin
  Form1.Caption:=IntToStr(Key);
  if Key = VK_UP then
    Button1.Top:=Button1.Top-10
  else if Key = VK_DOWN then
    Button1.Top:=Button1.Top+10
  else if Key = VK_LEFT then
    Button1.Left:=Button1.Left-10
  else if Key = VK_RIGHT then
    Button1.Left:=Button1.Left+10;
end;
```

<http://www.festra.com/wwwboard/messages/12909.html>

```
procedure TForm1.FormMouseDown(Sender: TObject;
Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  Form1.Caption:="";
  if Button = mbLeft then
    begin
      Form1.Caption:='Left';
    end
  else if Button = mbMiddle then
    begin
      Form1.Caption:='Middle';
    end
  else if Button = mbRight then
    begin
      Form1.Caption:='Right';
    end;

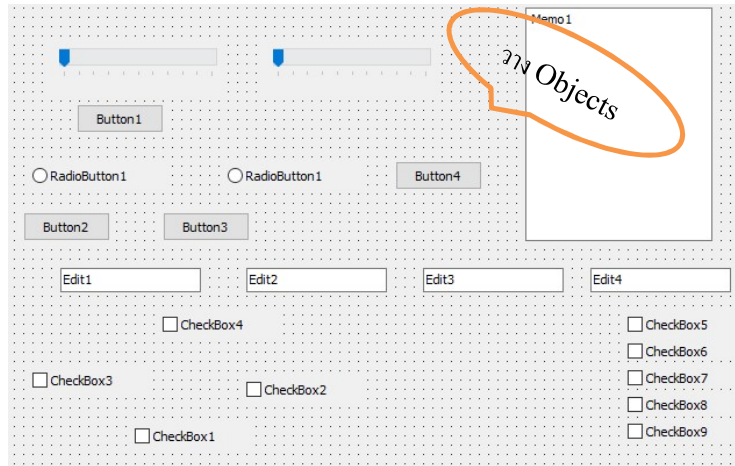
  Form1.Caption:=Form1.Caption+IntToStr(x)+';'+IntToStr(y)
;
  if ssCtrl in Shift then
    ShowMessage('Ctrl!');
end;

procedure TForm1.FormMouseMove(Sender: TObject;
Shift: TShiftState; X,
  Y: Integer);
begin
  Form1.Caption:=IntToStr(x)+';'+IntToStr(y);
end;

end.
```

2. รับค่าจาก Joy-Stick

Example



Code

```
unit UJoy1;

interface

uses

  Winapi.Windows, Winapi.Messages, System.SysUtils,
  System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, mmSystem,
  Vcl.StdCtrls, Vcl.ComCtrls;

type

  TMMJoyStick = packed record

    Msg: Cardinal; // The message ID

    Buttons: Longint; // The wParam

    XPos: word; // The lParam

    YPos: word;

    Result: Longint;

  end;

type

  TForm1 = class(TForm)

    Button1: TButton;

    TrackBar1: TTrackBar;

    RadioButton1: TRadioButton;

    TrackBar2: TTrackBar;
```

```
    RadioButton2: TRadioButton;

    Edit1: TEdit;

    Edit2: TEdit;

    CheckBox1: TCheckBox;

    CheckBox2: TCheckBox;

    CheckBox3: TCheckBox;

    CheckBox4: TCheckBox;

    Edit3: TEdit;

    CheckBox5: TCheckBox;

    CheckBox6: TCheckBox;

    Button2: TButton;

    Button3: TButton;

    Edit4: TEdit;

    CheckBox7: TCheckBox;

    CheckBox8: TCheckBox;

    CheckBox9: TCheckBox;

    Memo1: TMemo;

    Button4: TButton;

  procedure Button1Click(Sender: TObject);

  procedure FormCreate(Sender: TObject);

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

  procedure Button2Click(Sender: TObject);
```

```

procedure Button3Click(Sender: TObject);
private
    procedure MMJOY1BUTTONDOWN(var LocMessage:
TMMJoyStick); message MM_JOY1BUTTONDOWN;
    procedure MMJOY1BUTTONUP(var LocMessage:
TMMJoyStick); message MM_JOY1BUTTONUP;
    procedure MMJOY1MOVE(var LocMessage:
TMMJoyStick); message MM_JOY1MOVE; // ButtonDown
and Move are not shown, they just call ButtonUp
    procedure ProcessJoy(var LocMessage: TMMJoyStick);
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;
implementation

{$R *.dfm}
//http://sheepdogguides.com/dt4i.htm

procedure TForm1.Button1Click(Sender: TObject);
var
    myjoy: tjoyinfo;
begin
    joygetpos(joystickid1,@myjoy);
    trackbar1.position := myjoy.wypos;
    trackbar2.position := myjoy.wxpos;
    radiobutton1.checked := (myjoy.wbuttons and
joy_button1)>0;
    radiobutton2.checked := (myjoy.wbuttons and
joy_button2)>0;
end;

procedure TForm1.Button2Click(Sender: TObject);
var

```

```

myJoyCaps: TJoyCaps;
begin
    joyGetDevCaps(joystickid1,@myJoyCaps,
sizeof(myJoyCaps)); // for test
    joySetCapture(self.Handle, joystickid1, 100, true);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    joyReleaseCapture(joystickid1);
end;

procedure TForm1.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
    //joyReleaseCapture(joystickid1);
end;

procedure TForm1.FormCreate(Sender: TObject);
//var
    //myJoyCaps: TJoyCaps;
begin
    //joyGetDevCaps(joystickid1,@myJoyCaps,
sizeof(myJoyCaps)); // for test
    //joySetCapture(self.Handle, joystickid1, 100, true);
end;

procedure TForm1.MMJOY1BUTTONDOWN(var
LocMessage: TMMJoyStick);
begin
    ProcessJoy(LocMessage);
end;

procedure TForm1.MMJOY1BUTTONUP(var LocMessage:
TMMJoyStick);
begin
    ProcessJoy(LocMessage);
end;

```

```
end;

procedure TForm1.MMJOY1MOVE(var LocMessage:
TMMJoyStick);
begin
    ProcessJoy(LocMessage);
end;

procedure TForm1.ProcessJoy(var LocMessage:
TMMJoyStick);
begin
    Edit1.Text := IntToStr(LocMessage.ypos); // so you can
see
    Edit2.Text := IntToStr(LocMessage.xpos); // what is
happening
    Edit3.Text := IntToStr(LocMessage.Buttons);
    Edit4.Text := IntToStr(LocMessage.Result);
    Memo1.Lines.Add(IntToStr(LocMessage.Buttons));
    CheckBox1.checked := (LocMessage.Buttons and
joy_button1)>0;
    CheckBox2.checked := (LocMessage.Buttons and
joy_button2)>0;
    CheckBox3.checked := (LocMessage.Buttons and
joy_button3)>0;
```

```
    CheckBox4.checked := (LocMessage.Buttons and
joy_button4)>0;
    CheckBox5.checked := (LocMessage.Buttons and
joy_button5)>0;
    CheckBox6.checked := (LocMessage.Buttons and
joy_button6)>0;
    CheckBox7.checked := (LocMessage.Buttons and
joy_button7)>0;
    CheckBox8.checked := (LocMessage.Buttons and
joy_button8)>0;
    CheckBox9.checked := (LocMessage.Buttons and
joy_button9)>0;

    if (LocMessage.Buttons and joy_button1)>0 then
        Button4.Left:=Button4.Left-10;
    if CheckBox2.Checked then
        Button4.Left:=Button4.Left+10;
    if CheckBox4.Checked then
        Button4.Top:=Button4.Top-10;
    if CheckBox1.Checked then
        Button4.Top:=Button4.Top+10;
end;

end.
```

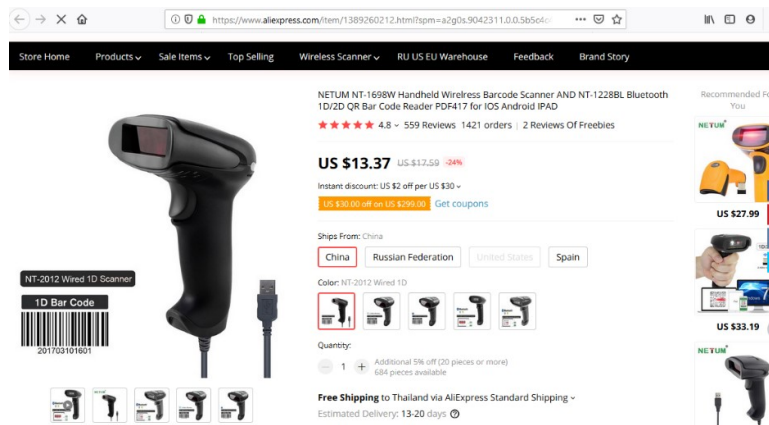
<http://mc-computing.com/Languages/Joysticks.htm#Delphi>

<http://sheepdogguides.com/dt4i.htm>

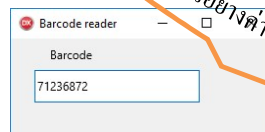
3. รับค่าจาก Barcode Reader

Barcode reader ส่วนใหญ่แล้วจะทำตัวเหมือน Keyboard คือเมื่อ Focus ไปที่ Object ที่สามารถรับค่า Key input ได้ และเมื่อ Reader อ่านค่า Barcode ได้ ค่านั้นจะถูกส่งให้ช่องรับข้อมูลเสมือนกับการพิมพ์ลงไปตรงๆ

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

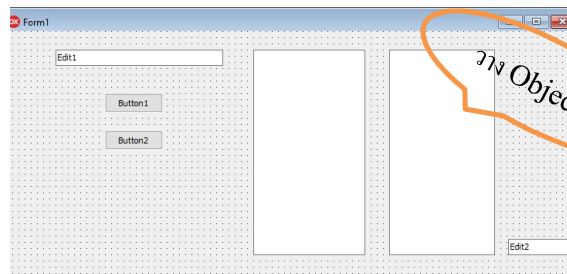


<https://www.aliexpress.com/item/1389260212.html?spm=a2g0s.9042241.0.0.5b5c4c4dUyqDiU>



ตัวอย่างคำที่อ่านได้จาก Barcode Reader

Example



Code

```
unit Unit1;  
  
interface  
  
uses  
  Winapi.Windows, Winapi.Messages, System.SysUtils,  
  System.Variants, System.Classes, Vcl.Graphics,  
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;  
  
type  
  TForm1 = class(TForm)  
    Edit1: TEdit;  
    Button1: TButton;
```

```
    Memo1: TMemo;  
    edi: TMemo;  
    Edit2: TEdit;  
    Button2: TButton;  
    procedure FormKeyDown(Sender: TObject; var Key:  
      Word; Shift: TShiftState);  
    procedure Button1Click(Sender: TObject);  
    procedure Button2Click(Sender: TObject);  
    procedure Edit1Change(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }
```

```
end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var
  Barcode:String;
  Price:Single;
begin
  BarCode:=Edit1.Text;
  if Barcode='D6901443047475' then
    Price:=11.23
  else if Barcode='A0000025' then
    Price:=25.23
  else
    Price:=0;
  Memo1.Lines.Add(format('%0.2f',[Price]));
  Edit1.SetFocus;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
```

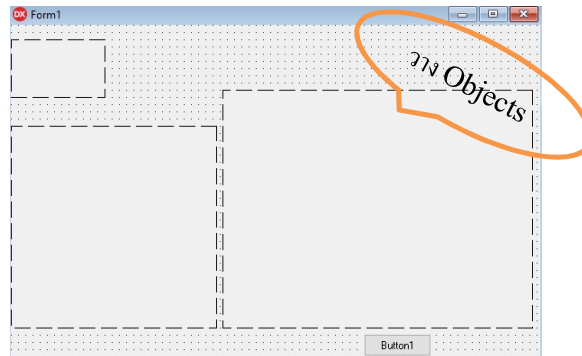
```
Total:Single;
i:Integer;
begin
  Total:=0;
  for i := 0 to Memo1.Lines.Count-1 do
    begin
      if Trim(Memo1.Lines.Strings[i])<>"" then
        Total:=Total+StrToFloat(Memo1.Lines.Strings[i]);
      end;
    Edit2.Text:=format('%0.2f',[Total]);
  end;

  procedure TForm1.Edit1Change(Sender: TObject);
  begin
    Memo1.Lines.Add(Edit1.Text);
  end;

  procedure TForm1.FormKeyDown(Sender: TObject; var
    Key: Word;
    Shift: TShiftState);
  begin
    if Key = VK_F2 then
      Edit1.SetFocus;
    end;
  end.
```

4. ถ่ายภาพ และ Video จากกล้อง Web camera

Example



Code

```
unit Demo01_MainForm;

interface

uses

  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms,
  Dialogs, ExtCtrls,
  VFrames, Vcl.StdCtrls;

type

  TForm1 = class(TForm)
    PaintBox1: TPaintBox;
    Image1: TImage;
    Button1: TButton;
    Image2: TImage;
    procedure FormActivate(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
  TCloseAction);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
    fActivated : boolean;
    fVideoImage: TVideoImage;
  public
```

```
    { Public declarations }
  end;
var
  Form1: TForm1;

implementation
{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
begin
  //PaintBox1.Align := alClient;
  fActivated := false;
  // Create instance of our video image class.
  fVideoImage := TVideoImage.Create;
  // Tell fVideoImage where to paint the images it
  receives from the camera
  // (Only in case we do not want to modify the images
  by ourselves)
  //fVideoImage.SetDisplayCanvas(PaintBox1.Canvas);
  fVideoImage.SetDisplayCanvas(Image1.Canvas);

  Caption := 'Demo01 [Initializing video...]';
end;

procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
Image1.Picture.SaveToFile('E:\abc.bmp');
Image2.Picture:=Image1.Picture;
end;

procedure TForm1.FormActivate(Sender: TObject);
var
    DeviceList: TStringList;
begin
    IF fActivated then
        exit;
    fActivated := true;
    // Get list of available cameras
    DeviceList := TStringList.Create;
    fVideoImage.GetListOfDevices(DeviceList);
    IF DeviceList.Count < 1 then
        begin
            // If no camera has been found, terminate program
            Caption := 'Demo01 [No video device found]';
```

```
        MessageDlg('No video device found.'#10'Application
will terminate.', mtError, [mbOK], 0);

        Application.Terminate;
        exit;
    end
    else begin
        // If at least one camera has been found, initialize
        first camera within the list
        fVideoImage.VideoStart(DeviceList[0]);
        Caption := 'Demo01 [' + DeviceList[0] + ']';
    end;
end;

procedure TForm1.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
    fVideoImage.VideoStop;
end;
end.
```

5. บันทึกภาพเป็น File ต่างๆ

Code

```
uses
JPEG;

procedure BMP_To_JPEG(BMPFName, JPEGFName :
string);
var
    MyJPEG : TJPEGImage;
    MyBitmap : TBitmap;
begin
    MyJPEG := TJPEGImage.Create;
try
    MyBitmap := TBitmap.Create;
try
    MyBitmap.LoadFromFile(BMPFName);
    MyJPEG.Assign(MyBitmap);
```

```
    MyJPEG.SaveToFile(JPEGFName);
finally
    MyBitmap.Free;
end;
finally
    MyJPEG.Free;
end;
end;

procedure BitmapToJpeg(FileName: string; Img:TGraphic);
var
    m_jpeg: TJpegImage;
    m_bitmap: TBitmap;
begin
```

030143361 การโปรแกรมคอมพิวเตอร์สำหรับงานควบคุม

```
try
  m_jpeg:= TJpegImage.Create;
  m_bitmap := TBitmap.Create;
  m_bitmap.Width := Img.Width;
  m_bitmap.Height := Img.Height;
  m_bitmap.Canvas.Draw( 0, 0, Img);
  // m_bitmap.Canvas.StretchDraw( 0, 0, , ...,Img);
```

<https://delphiaball.co.uk/2015/10/20/bitmap-to-pngjpegjpgs-in-delphi-cbuilder/>

```
m_jpeg.Assign( m_bitmap);
m_jpeg.SaveToFile( FileName);

finally
  m_jpeg.Free;

end;

end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var
  bmp: TBitmap;
  jpg: TJpegImage;
  scale: Double;
begin
  if opendialog1.execute then
    begin
      jpg := TJpegImage.Create;
      try
        jpg.LoadFromFile(opendialog1.filename);
        if jpg.Height > jpg.Width then
          scale := 50 / jpg.Height
        else
          scale := 50 / jpg.Width;
        bmp := TBitmap.Create;
        try
          {Create thumbnail bitmap, keep pictures aspect
ratio}
          bmp.Width := Round(jpg.Width * scale);
```

```
        bmp.Height:= Round(jpg.Height * scale);
        bmp.Canvas.StretchDraw(bmp.Canvas.Cliprect,
jpg);
        {Draw thumbnail as control}
        Self.Canvas.Draw(100, 10, bmp);
        {Convert back to JPEG and save to file}
        jpg.Assign(bmp);
        jpg.SaveToFile(
          ChangeFileExt(opendialog1.filename,
'_thumb.JPG')
        );
      finally
        bmp.free;
      end;
    finally
      jpg.free;
    end;
  end;
end;
```

<http://delphidabbler.com/tips/99>

Example

[Code\week8\Ex_KeyMouse_1](#)

[Code\week8\Ex_Joy_1](#)

[Code\week8\Ex_Barcode_1](#)

[Code\week8\EX_Cam_Demo_1](#)

[Code\week8\EX_Cam_Demo_2](#)

[Code\week8\EX_Cam_Demo_3](#)

Exercise

- อธิบาย Event OnKeyDown, OnKeyPress และ OnKeyUp
- เขียนโปรแกรมถ่ายภาพจากกล้อง Webcam แล้วบันทึกเป็น Jpeg Files

Assignment

- เขียนเกมส์เคลื่อนที่ Object ด้วย Joy-Stick ไปชนกับรูป 4 เหลี่ยมที่มีการ Random พิกัดให้อยู่บน Form 2 กล้อง เมื่อชนแล้วกล้องนั้นจะเปลี่ยนสี หากชนครบทั้ง 2 กล้องแล้วจะมีข้อความบอกว่าชนะ แล้วพร้อมบอกเวลาในการเล่น

Answer Sheet

[Code\week8\Ans_CamJpg_1](#)

[Code\week8\Ans_KeyGame_1](#)