

```

{
This is a delphi 4 version of John Hyde's Buttons and lights example

Author: Ray Wright RayWright111@hotmail.com (15 November 2000)

Please feel free to make any modifications or improvements to this, but
please send me an email letting me know what changes/improvements you've made.
Thanks.

This unit uses HidControllerClass.pas by Robert Marquardt. robert_marquardt@gmx.de the
hidcontroller unit can be found at http://usb-by-example.com
}
unit main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  HidControllerClass, StdCtrls, Buttons, ExtCtrls;

type
  TForm1 = class(TForm)
    JvHidDeviceController1: TJvHidDeviceController;
    Panel1: TPanel;
    Shape1: TShape;
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;
    Shape5: TShape;
    Shape6: TShape;
    Shape7: TShape;
    Shape8: TShape;
    Panel2: TPanel;
    Shape9: TShape;
    Shape10: TShape;
    Shape11: TShape;
    Shape12: TShape;
    Shape13: TShape;
    Shape14: TShape;
    Shape15: TShape;
    Shape16: TShape;
    Panel3: TPanel;
    Shape17: TShape;
    Shape18: TShape;
    Shape19: TShape;
    Shape20: TShape;
    Shape21: TShape;
    Shape22: TShape;
    Shape23: TShape;
    Shape24: TShape;
    Label1: TLabel;
    Label2: TLabel;
    Label4: TLabel;
    Shape25: TShape;
    Shape26: TShape;
    Shape27: TShape;
    Shape28: TShape;
    Shape29: TShape;
    Shape30: TShape;
    Shape31: TShape;
    Shape32: TShape;
    Shape33: TShape;
    Shape34: TShape;
    Shape35: TShape;
    Shape36: TShape;
    Shape37: TShape;
    Shape38: TShape;
    Shape39: TShape;
    Shape40: TShape;
    StaticText1: TStaticText;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Timer1: TTimer;
    ComboBox1: TComboBox;
    Shape41: TShape;
    Shape42: TShape;
    Shape43: TShape;
  end;

```

```

Shape44: TShape;
Shape45: TShape;
procedure JvHidDeviceController1DeviceChange(Sender: TObject);
function JvHidDeviceController1Enumerate(HidDev: TJvHidDevice;
  Index: Integer): Boolean;
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
function ReverseBits(index, NumBits: word ): word;
procedure Timer1Timer(Sender: TObject);
procedure Shape9MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape25MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape26MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape27MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape28MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape10MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape11MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape12MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape13MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape14MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape15MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape16MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape29MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape30MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape31MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Shape32MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton3Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure ComboBox1DragOver(Sender, Source: TObject; X, Y: Integer;
  State: TDragState; var Accept: Boolean);
procedure ComboBox1Change(Sender: TObject);
private
  { Private declarations }
  DevList: TList;
public
  { Public declarations }
end;

var
  Form1: TForm1;
  TheDev: TJvHidDevice;

  soft_buttons : byte; //each bit is a soft buttons state
  real_buttons : byte; //each bit is a real buttons state
  lights       : byte; //each bit is an LED state

implementation

uses bal2;

{$R *.DFM}

procedure write_lights(what : byte);
var
  buf: array [0..1] of Byte;
  written: Cardinal;
  towrite: Cardinal;
begin
  towrite:=2;
  if (form1.statictext1.caption>'') and (towrite > 1) then
  begin
    TheDev := form1.DevList.Items[0];
    buf[1]:=what;

```

```

    HidCheck(TheDev.WriteFile(buf,towrite,written));
end;
end;

procedure TForm1.JvHidDeviceController1DeviceChange(Sender: TObject);
var
    I: Integer;
begin
    if DevList <> nil then
        begin
            for I := 0 to DevList.Count-1 do
                begin
                    TheDev := DevList.Items[I];
                    TheDev.Free;
                end;
            DevList.Clear;
        end
    else
        DevList := TList.Create;
        JvHidDeviceController1.Enumerate;
    end;

function TForm1.JvHidDeviceController1Enumerate(HidDev: TJvHidDevice;
    Index: Integer): Boolean;
var
    Dev: TJvHidDevice;
begin
    if HidDev.ProductName <> '' then
        statictext1.caption:=HidDev.ProductName
    else
        statictext1.caption:=Format('Device VID=%x PID=%x',[HidDev.Attributes.VendorID,HidDev.Attributes.ProductID]);

    JvHidDeviceController1.CheckOutByIndex(Dev, Index);
    DevList.Add(Dev);
    Result := True;
end;

function ToInt(str: string): Cardinal;
begin
    Result := 0;
    if str <> '' then
        Result := StrToInt('$'+str);
end;

procedure update_lights_display;
begin
    if lights and $80 = 0 then form1.shape1.brush.color:=clblack else form1.shape1.brush.color:=clred;
    if lights and $40 = 0 then form1.shape2.brush.color:=clblack else form1.shape2.brush.color:=clred;
    if lights and $20 = 0 then form1.shape3.brush.color:=clblack else form1.shape3.brush.color:=clred;
    if lights and $10 = 0 then form1.shape4.brush.color:=clblack else form1.shape4.brush.color:=clred;
    if lights and $08 = 0 then form1.shape5.brush.color:=clblack else form1.shape5.brush.color:=clred;
    if lights and $04 = 0 then form1.shape6.brush.color:=clblack else form1.shape6.brush.color:=clred;
    if lights and $02 = 0 then form1.shape7.brush.color:=clblack else form1.shape7.brush.color:=clred;
    if lights and $01 = 0 then form1.shape8.brush.color:=clblack else form1.shape8.brush.color:=clred;
end;

procedure update_soft_buttons_display;
begin
    if soft_buttons and $80 = 0 then form1.shape25.top:=35 else form1.shape25.top:=10;
    if soft_buttons and $40 = 0 then form1.shape26.top:=35 else form1.shape26.top:=10;
    if soft_buttons and $20 = 0 then form1.shape27.top:=35 else form1.shape27.top:=10;
    if soft_buttons and $10 = 0 then form1.shape28.top:=35 else form1.shape28.top:=10;
    if soft_buttons and $08 = 0 then form1.shape29.top:=35 else form1.shape29.top:=10;
    if soft_buttons and $04 = 0 then form1.shape30.top:=35 else form1.shape30.top:=10;
    if soft_buttons and $02 = 0 then form1.shape31.top:=35 else form1.shape31.top:=10;
    if soft_buttons and $01 = 0 then form1.shape32.top:=35 else form1.shape32.top:=10;
end;

procedure update_real_buttons_display;
begin
    if real_buttons and $80 = 0 then form1.shape40.top:=10 else form1.shape40.top:=35;
    if real_buttons and $40 = 0 then form1.shape39.top:=10 else form1.shape39.top:=35;
    if real_buttons and $20 = 0 then form1.shape38.top:=10 else form1.shape38.top:=35;
    if real_buttons and $10 = 0 then form1.shape37.top:=10 else form1.shape37.top:=35;
    if real_buttons and $08 = 0 then form1.shape36.top:=10 else form1.shape36.top:=35;
    if real_buttons and $04 = 0 then form1.shape35.top:=10 else form1.shape35.top:=35;
    if real_buttons and $02 = 0 then form1.shape34.top:=10 else form1.shape34.top:=35;
    if real_buttons and $01 = 0 then form1.shape33.top:=10 else form1.shape33.top:=35;

```

```

end;

procedure update_all;
begin
  update_soft_buttons_display;

  case form1.combobox1.itemindex of
    0: lights:=soft_buttons;
    1: lights:=real_buttons;
    2: lights:=soft_buttons or real_buttons;
    3: lights:=soft_buttons and real_buttons;
    4: lights:=soft_buttons xor real_buttons;
  end;

  write_lights(lights);
  update_lights_display;
end;

procedure update_screen;
begin
  form1.statictext1.Caption := TheDev.VendorName;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  JvHidDeviceController1.OnDeviceChange := JvHidDeviceController1DeviceChange;
  JvHidDeviceController1.OnEnumerate := JvHidDeviceController1Enumerate;
  JvHidDeviceController1.Enumerate;
end;

procedure TForm1.FormDestroy(Sender: TObject);
var
  I: Integer;
begin
  for I := 0 to DevList.Count-1 do
  begin
    TheDev := DevList.Items[I];
    JvHidDeviceController1.CheckIn(TheDev);
  end;
  DevList.Free;
end;

function tform1.ReverseBits ( index, NumBits: word ): word;
var
  i, rev: word;
begin
  rev := 0;
  for i := 0 to NumBits-1 do
  begin
    rev := (rev SHL 1) OR (index AND 1);
    index := index SHR 1;
  end;
  ReverseBits := rev;
end;

procedure TForm1.Timer1Timer(Sender: TObject); //i've hooked the reading up to the timer, its fast enough for this
purpose.
var //For interrupt i'd link the read to its own thread
  buf: array [0..1] of Byte;
  toread: Cardinal;
  read: Cardinal;
  temp : byte;
  rev,i : word;
begin
  // if (ListBox1.Items.Count > 0) and (ListBox1.ItemIndex >= 0) then
  begin
    TheDev := DevList.Items[0];
    toread := TheDev.Caps.InputReportByteLength;
    HidCheck(TheDev.ReadFile(buf,toread,read));
    real_buttons:=buf[1];

    real_buttons:=reversebits(real_buttons,8); //for plug orientation

    update_real_buttons_display;
    update_all;
  end;
end;

procedure TForm1.Shape9MouseDown(Sender: TObject; Button: TMouseButton;

```

```

    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $80; //toggle bit 7
    update_all;
end;

procedure TForm1.Shape25MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $80; //toggle bit 7
    update_all;
end;

procedure TForm1.Shape26MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $40; //toggle bit 6
    update_all;
end;

procedure TForm1.Shape27MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $20; //toggle bit 5
    update_all;
end;

procedure TForm1.Shape28MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $10; //toggle bit 4
    update_all;
end;

procedure TForm1.Shape10MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $40; //toggle bit 6
    update_all;
end;

procedure TForm1.Shape11MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $20; //toggle bit 5
    update_all;
end;

procedure TForm1.Shape12MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $10; //toggle bit 4
    update_all;
end;

procedure TForm1.Shape13MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $08; //toggle bit 3
    update_all;
end;

procedure TForm1.Shape14MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $04; //toggle bit 2
    update_all;
end;

procedure TForm1.Shape15MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $02; //toggle bit 1
    update_all;
end;

procedure TForm1.Shape16MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin

```

```

    soft_buttons:=soft_buttons xor $01; //toggle bit 0 Least significant bit
    update_all;
end;

procedure TForm1.Shape29MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $08; //toggle bit 3
    update_all;
end;

procedure TForm1.Shape30MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $04; //toggle bit 2
    update_all;
end;

procedure TForm1.Shape31MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $02; //toggle bit 1
    update_all;
end;

procedure TForm1.Shape32MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    soft_buttons:=soft_buttons xor $01; //toggle bit 0 Least significant bit
    update_all;
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
var
    Form: TInfoForm;
begin
    begin
        TheDev := DevList.Items[0];
        Form := TInfoForm.Create(Self);
        Form.ShowModal;
        Form.Free;
    end;
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
    lights:=0;
    soft_buttons:=0;
    real_buttons:=0;
    update_all;
    combobox1.itemindex:=0;
end;

procedure TForm1.ComboBox1DragOver(Sender, Source: TObject; X, Y: Integer;
    State: TDragState; var Accept: Boolean);
begin
    update_all;
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
    update_all;
end;

end.

```