

Listing Program Sistem Pengontrolan Miniatur Mesin (MTU).

unit ufrmVeyor;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, OleCtrls, ShockwaveFlashObjects_TLB, VrControls, VrDigit, CPort, StdCtrls, VrButtons, ExtCtrls, CPortCtl;

type

```
TForm1 = class(TForm)
  ShockwaveFlash1: TShockwaveFlash;
  ComPort1: TComPort;
  Label1: TLabel;
  VrPowerButton1: TVrPowerButton;
  VrPowerButton2: TVrPowerButton;
  ComDataPacket1: TComDataPacket;
  Timer1: TTimer;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Timer2: TTimer;
  Memo1: TMemo;
  ComTerminal1: TComTerminal;
  Memo2: TMemo;
  procedure FormCreate(Sender: TObject);
  procedure SetPort(address, Value: Word);
  function GetPort(address: Word): Word;
  procedure Sound(aFreq, aDelay: Integer);
  procedure NoSound;
  procedure Timer1Timer(Sender: TObject);
  procedure VrPowerButton1Click(Sender: TObject);
  procedure VrPowerButton2Click(Sender: TObject);
  procedure ComDataPacket1Packet(Sender: TObject; const Str: String);
  procedure FormShow(Sender: TObject);
  procedure Timer2Timer(Sender: TObject);
  procedure FormDblClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
```

```
{ Private declarations }  
public  
{ Public declarations }  
end;
```

```
var  
Form1: TForm1;  
SpSound: Boolean;  
KondisiOn: Boolean;  
Muncul: Boolean;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure Delay(MSecs: Integer);
```

```
var
```

```
FirstTickCount: DWord;
```

```
begin
```

```
FirstTickCount := GetTickCount;
```

```
repeat
```

```
Application.ProcessMessages;
```

```
Sleep(1);
```

```
until ((GetTickCount - FirstTickCount) >= DWord(MSecs));
```

```
end;
```

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

```
begin
```

```
ShockwaveFlash1.Movie := ExtractFilePath(ParamStr(0))+'konveyor.swf';
```

```
ShockwaveFlash1.Stop;
```

```
NoSound;
```

```
SpSound := True;
```

```
Label2.Color := clBlack;
```

```
ComPort1.LoadSettings(stIniFile, ExtractFilePath(ParamStr(0))+'konveyor.ini');
```

```
end;
```

```
procedure TForm1.SetPort(address, Value: Word);
```

```
var
```

```
bValue: Byte;
```

```
begin
```

```
bValue := trunc(Value and 255);
```

```
asm
```

```
mov dx, address
```

```
mov al, bValue
```

```
out dx, al
```

```
end;
```



```

end;
function TForm1.GetPort(address: Word): Word;
var
  bValue: Byte;
begin
  asm
    mov dx, address
    in al, dx
    mov bValue, al
  end;
  GetPort := bValue;
end;

procedure TForm1.Sound(aFreq, aDelay: Integer);

```

```

  procedure DoSound(Freq: Word);
  var
    B: Byte;
  begin
    if Freq > 18 then
      begin
        Freq := Word(1193181 div Longint(Freq));
        B := Byte(GetPort($61));

        if (B and 3) = 0 then
          begin
            SetPort($61, Word(B or 3));
            SetPort($43, $B6);
          end;

        SetPort($42, Freq);
        SetPort($42, Freq shr 8);
      end;
    end;
  end;

```

```

  procedure Delay(MSecs: Integer);
  var
    FirstTickCount: LongInt;
  begin
    FirstTickCount := GetTickCount;
    repeat
      //Sleep(1);
      //or use
      Application.ProcessMessages; // instead; // of Sleep
    until ((GetTickCount - FirstTickCount) >= Longint(MSecs));
  end;

```

```

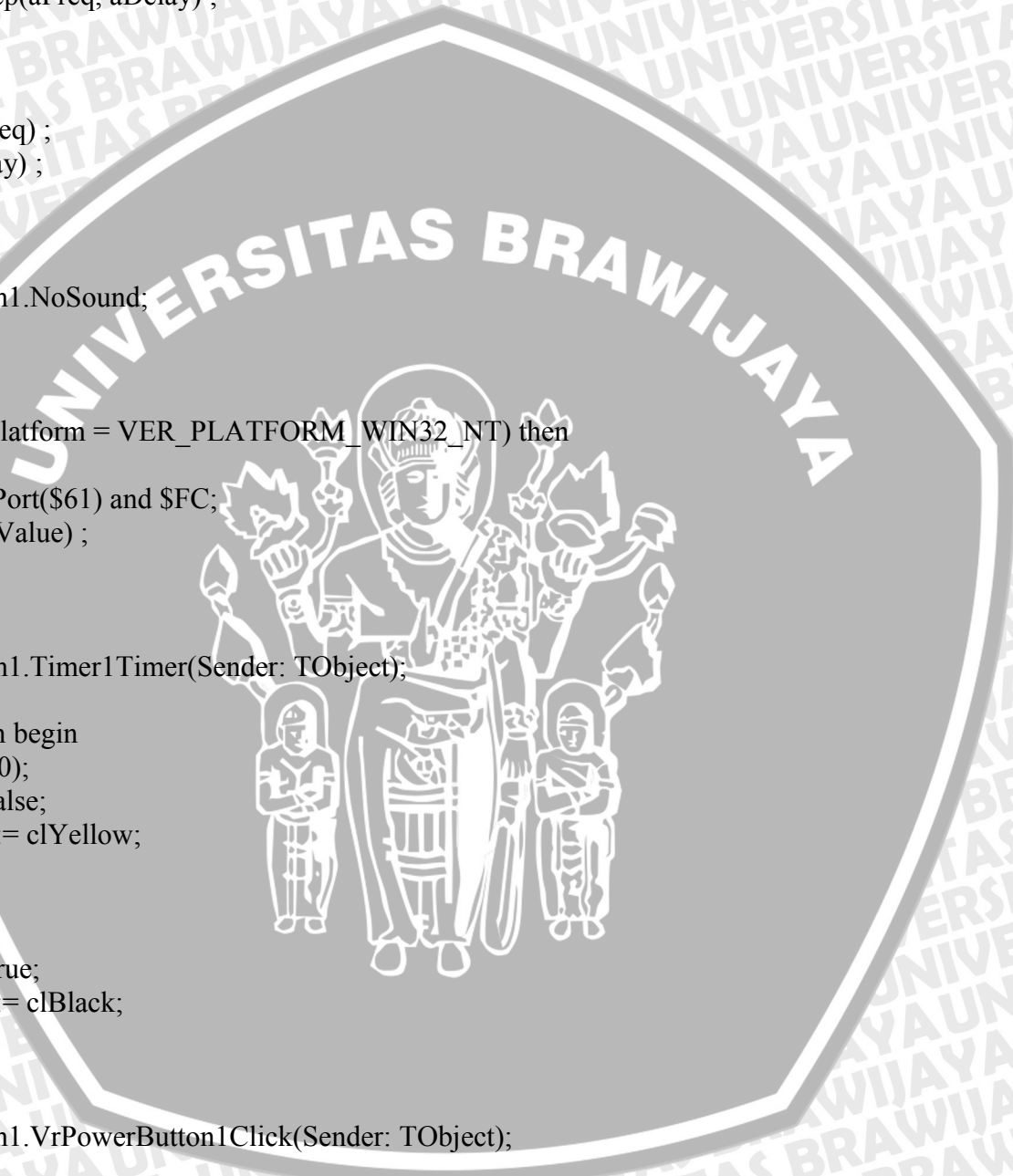
end;
begin
if Win32Platform = VER_PLATFORM_WIN32_NT then
begin
  Windows.Beep(aFreq, aDelay) ;
end
else
begin
  DoSound(aFreq) ;
  //Delay(aDelay) ;
end;
end;

procedure TForm1.NoSound;
var
  Value: Word;
begin
if not (Win32Platform = VER_PLATFORM_WIN32_NT) then
begin
  Value := GetPort($61) and $FC;
  SetPort($61, Value) ;
end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
if SpSound then begin
  Sound(500,500);
  SpSound := False;
  Label2.Color := clYellow;
end
else begin
  NoSound;
  SpSound := True;
  Label2.Color := clBlack;
end;
end;

procedure TForm1.VrPowerButton1Click(Sender: TObject);
begin
//if ImageButton1.Tag = 0 then begin
//ImageButton1.NoneBitmap := Image4.Picture.Bitmap;
//ImageButton2.NoneBitmap := Image5.Picture.Bitmap;
//ImageButton1.Tag := 1;
//ImageButton2.Tag := 0;

```



Paritas penjumlahan	
---------------------	--




```

Memo1.Lines.Add('ON');
KondisiOn := True;
//end
ShockwaveFlash1.Play;
end;

procedure TForm1.VrPowerButton2Click(Sender: TObject);
begin
//if ImageButton2.Tag = 0 then begin
//ImageButton2.NoneBitmap := Image3.Picture.Bitmap;
//ImageButton1.NoneBitmap := Image2.Picture.Bitmap;
//ImageButton2.Tag := 1;
//ImageButton1.Tag := 0;
Memo1.Lines.Add('OFF');
KondisiOn := False;
//end
ShockwaveFlash1.Stop;
end;

procedure TForm1.ComDataPacket1Packet(Sender: TObject; const Str: String);
var
    parity: byte;
begin
//Label1.Caption := Str;
Memo2.Lines.Add(IntToStr(Ord(Str[1])));
Memo2.Lines.Add(IntToStr(Ord(Str[2])));
Memo2.Lines.Add(IntToStr(Ord(Str[3])));
Memo2.Lines.Add(IntToStr(Ord(Str[4])));
parity := Ord(Str[3]) + Ord(Str[2]) + Ord(Str[1]);
if Ord(Str[4]) = parity then begin
    Label3.Caption := IntToStr(Ord(Str[2]));
    Label4.Caption := IntToStr(Ord(Str[1]));
    if (Ord(Str[3]) = $00) and (KondisiOn) then
        Timer1.Enabled := True
    else begin
        Timer1.Enabled := False;
        NoSound;
        SpSound := True;
        Label2.Color := clBlack;
    end;
end;
end;
{VrDigitGroup1.Value := StrToInt(Copy(Str,1,2));
if Copy(Str,1,1) = '1' then
    Timer1.Enabled := True
else begin

```

```

Timer1.Enabled := False;
NoSound;
SpSound := True;
Label2.Color := clBlack;
end;

```

```

procedure TForm1.FormShow(Sender: TObject);
begin
Sleep(1000);
ComPort1.Connected := True;
Timer2.Enabled := True;
KondisiOn := False;
end;

```

```

procedure TForm1.Timer2Timer(Sender: TObject);
begin
Timer2.Enabled := FALSE;
if Comport1.Connected then begin
if Memo1.Lines.Count <> 0 then begin
ComPort1.SetRTS(True);
Delay(1200);
Comport1.WriteStr(Memo1.Lines.Strings[0]);
Memo1.Lines.Delete(0);
Delay(100);
ComPort1.SetRTS(False);
end
else begin
ComPort1.SetRTS(True);
Delay(1200);
ComPort1.WriteStr('IN');
Delay(100);
ComPort1.SetRTS(False);
end;
end;
Timer2.Enabled := True;
end;

```

```

procedure TForm1.FormDbClick(Sender: TObject);
begin
Muncul := not Muncul;
if Muncul then begin
ComTerminal1.Visible := True;
end
else begin
ComTerminal1.Visible := False;
end;
end;

```



end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin

ComPort1.StoreSettings(stIniFile,ExtractFilePath(ParamStr(0))+'konveyor.ini');

end;

end.

