KARYA TULIS ILMIAH

APLIKASI PENGONTROLAN PERALATAN LISTRIK RUMAH BERBASIS SMS

Di Susun Oleh : Agusman

Nim : 10103462

Pembimbing I : Budi Irawan,S,Si.MT

Pembimbing II : Santi Novani.S,Si.MT

PROGRAM STUDI S1
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
BANDUNG
2007

ABSTRAK

Saat ini teknologi Short Message Services (SMS) menjadi sebuah teknologi alternatif, sudah banyak aplikasi-aplikasi menggunakan SMS sebagai media komunikasi datanya. Dengan metode seperti ini diharapkan pelanggan menjadi terpenuhi semua kebutuhannya.

Banyaknya penggunaan alat kontrol secara otomatis saat ini menunjukan pesatnya perkembangan teknologi. Banyaknya kebutuhan akan alat pengontrolan tersebut dikarenakan banyaknya permintaan dari konsumen sehingga para produsen sangatlah membutuhkan akan alat pengontrolan otomatis tersebut. Berbicara tentang mesin otomatis, maka akan berbicara tentang kontrol.

LATAR BELAKANG MASALAH

Kebutuhan akan sistem untuk pengendalian jarak jauh semakin meningkat sejalan dengan era globalisasi dimana perpindahan dan pergerakan manusia semakin luas dan cepat. Selama ini masyarakat dapat mengontrol sesuatu dari jarak jauh dengan menggunakan *remote control*, akan tetapi pengontrolan tersebut terhambat oleh jarak. Apabila jarak antara alat yang dikontrol dengan pengontrol itu melewati jarak toleransinya, maka peralatan tersebut tidak dapat berfungsi sesuai dengan yang diinginkan. Salah satu solusi untuk mengatasi masalah tersebut yaitu menggunakan ponsel sebagai *remote control*. Dengan memanfaatkan ponsel sebagai pengendali jarak jauh batasan jarak dapat diabaikan selama ponsel yang digunakan sebagai pengirim perintah berada di dalam jangkauan operator dari *SIM Card* yang digunakan. Pemanfaatan ponsel sebagai pengendali peralatan jarak jauh dapat mewujudkan keinginan manusia untuk mengendalikan peralatan tanpa adanya batasan jarak. Saat ini perkembangan teknologi di bidang ponsel semakin canggih, membuat perusahaan yang menyediakan fasilitas komunikasi ponsel di Indonesia semakin gencar memperluas jangkauannya. Maka dibuatlah suatu pengontrolan jarak jauh melalui SMS.

IDENTIFIKASI MASALAH

Dari uraian latar belakang diatas, maka terdapat beberapa masalah utama yang perlu diidentifikasikan. Masalah-masalah tersebut antara lain :

1. Bagaimana aplikasi yang dibuat dapat mengenali perintah SMS yang dimasukkan oleh *user*.

2. Bagaimana aplikasi yang dibuat dapat mengenali dan mengolah perintah tersebut, sehingga sirkuit dapat menangkap maksud dari perintah tersebut.
3. Bagaimana aplikasi pengontrolan yang dibuat dapat memberikan jawaban ke perangkat listrik, sehingga dapat memberikan laporan status.

MAKSUD DAN TUJUAN

Tujuan dari penelitian ini adalah mengendalikan pengontrolan peralatan listrik rumah melalui ponsel sebagai *remote control dengan perintah SMS*, sehingga masyarakat dapat mengendalikan peralatan listrik rumah tangga saat akan keluar rumah, pulang ke rumah maupun saat berliburan keluar kota dapat dilakukan proses pengontrolan dengan jarak jauh.

BATASAN MASALAH

Dalam pembuatan aplikasi ini, perlu diberikan beberapa batasan agar alur pembicaraan tidak terlalu meluas. Batasan-batasan itu antara lain :

1. Pengontrolan peralatan listrik berbasis SMS ini tidak menyediakan arus listrik cadangan apabila aliran listrik padam atau arus listrik mati .
2. Sirkuit hanya dikontrol oleh program dengan operator dan SMS.
3. Pengontrolan program ini bisa dilakukan ke semua alat-alat listrik dirumah seperti (lemari es, tv, pagar remote dll, tetapi penyusun hanya memakai simulasi *miniatur rumah* yang terdiri dari beberapa led lampu, pagar kontrol dan kipas angin, sebagai bahan percobaan yang mana fungsinya sama untuk digunakan.

PENGERTIAN PERINTAH-PERINTAH ATCOMMAND

AT Command digunakan untuk melakukan komunikasi dengan modem. Ponsel pada dasarnya adalah modem AT Command pun berlaku pada ponsel, dengan menggunakan kabel data yang tersedia pada masing-masing merk ponsel, kita dapat berkomunikasi dengan ponsel melalui komputer.

Keuntungan menggunakan perintah AT Command adalah dapat mengotomasikan tugas pada Mobile Phone mulai dari penerimaan sampai dengan pengiriman balasan SMS. Untuk mengotomatisasi tugas pada ponsel, maka diperlukan juga bahasa pemograman yang dapat berkomunikasi dengan port pada komputer.

ANALISIS MASALAH

Permasalahan utama dalam merancang Aplikasi ini, yakni dengan menggunakan SMS sebagaimana untuk membuat sebuah program yang mampu menerima SMS dan dapat memberi informasi SMS tersebut ke rangkaian serta mengolahnya menjadi format data yang diinginkan, kemudian mengirimkan kembali SMS tersebut ke user.

Analisis kebutuhan Sistem

Sistem ini digunakan untuk memberikan kemudahan kepada user dimanapun user berada, hanya dengan memanfaatkan telepon selular melalui fasilitas *short message service* (SMS) dan memberilan proses perintah kerangkaian.

KEBUTUHAN PERANGKAT KERAS

Komponen-komponen yang termasuk dalam perangkat keras untuk kebutuhan aplikasi sistem yang dibangun adalah, sebagai berikut :

1. Prosesor P4 2,4 GHz
2. Memory 256 MB
3. Harddisk 40 GB
4. VGA 64 MB
5. Monitor 14”
6. Telepon selular dengan SIM *Card-nya* sebagai *transmitter* (pengiriman SMS) dan *receive*r pesan SMS (penerima SMS) dengan menggunakan SIM Card ProXL.
7. Kabel Data USB WP-SM02 Siemens
8. Mobile Phone Siemens 2128 (Modem)
9. Sirkuit atau Rangkaian Listrik yang terdiri dari 4 LD dimana untuk melengkapi percobaan program.

KEBUTUHAN PERANGKAT LUNAK

Berdasarkan kebutuhan sistem secara umum, perangkat lunak yang akan di bangun harus memenuhi kebutuhan-kebutuhan sebagai berikut :

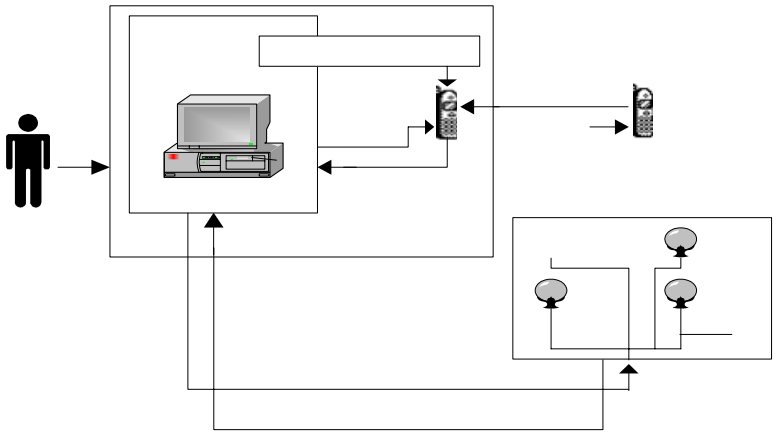
1. Mampu mengambil pesan SMS yang tersimpan dalam telepon selular yang berfungsi sebagai gateway.
2. Mampu mengidentifikasi apakah pesan SMS yang masuk berasal dari User yang telah terdaftar dalam database sistem (sebagai pendukung kemampuan sistem)

3. Mampu memproses permintaan user sesuai jenis permintaannya.
4. Mampu memberikan pesan hasil pemrosesan sistem (laporan).
5. Mampu mengirim pesan SMS hasil dari proses sistem melalui gateway.
6. Mampu menyimpan semua data penerimaan dan pengiriman pesan.

Komponen-komponen yang termasuk dalam perangkat lunak untuk kebutuhan aplikasi sistem yang dibangun adalah, sebagai berikut :

1. Borland Delphi 6
2. Microsoft Windows XP
3. Protel 99 SE
4. Microsoft Office Access
5. Software Kabel Data USB WP-SM02Siemens

DESAIN SISTEM SMS GATEWAY



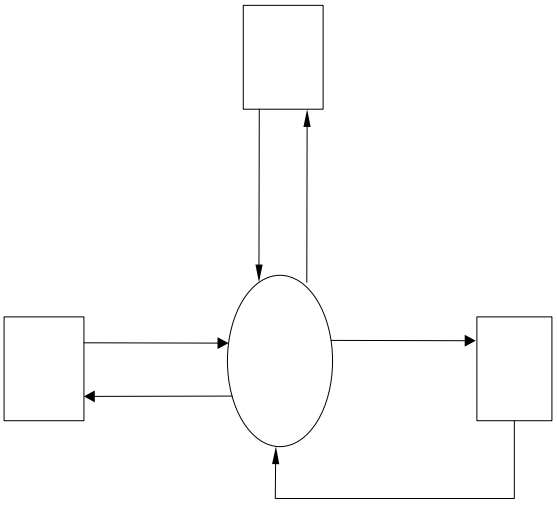
Tabel 3.1 Spesifikasi Kode Rangkaian

Kode	Register Bit	Keterangan

LD 0	D0	Dengan menggunakan Kipas Angin terdiri dari D0 saja, jika On maka LD 0 akan nyala dan jika LD 0 Off Kipas Angin akan mati.
LD 1	D1 dan D3	Dengan menggunakan lampu, yang terdiri dari Led D1 dan D3. jika On maka LD 1 akan menyala dan jika LD 1 Off maka akan mati.
LD 2	D2	Dengan menggunakan Kipas Angin yang terdiri D2 saja, jika On maka LD 2 akan menyala dan jika LD 2 Off Kipas angin akan mati
LD 3	D4 dan D5	Dengan menggunakan lampu, yang terdiri dari Led D4 dan D5. jika On maka LD 3 akan menyala dan jika LD 3 Off maka akan mati.
LD 4	C0 dan C0-	Dengan menggunakan Pagar Automatis yang terdiri dari C0 dan C0 -, jika On maka LD4 akan terbuka dan jika LD4 Off pagar akan tertutup.

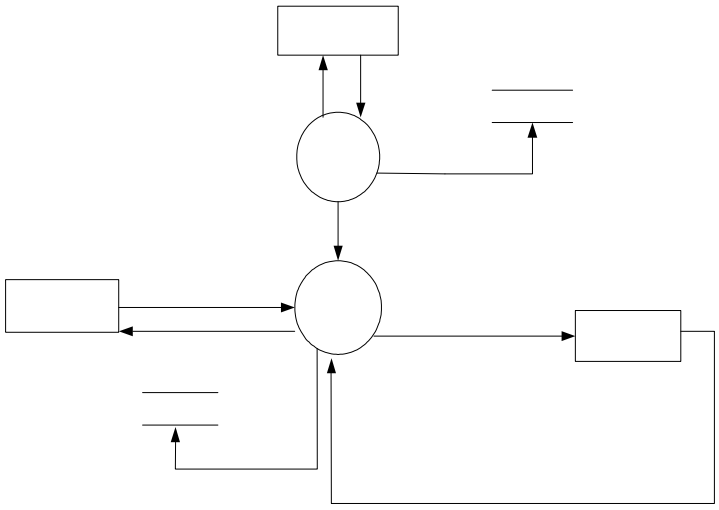
DIAGRAM KONTEKS

Adapun gambaran diagram konteks pada aplikasi SMS gateway ini adalah sebagai berik



Gambar 3.2 Diagram konteks

DATA FLOW DIAGRAM



Gambar 3.3 DFD level 1

PERINTAH-PERINTAH AT COMMAND

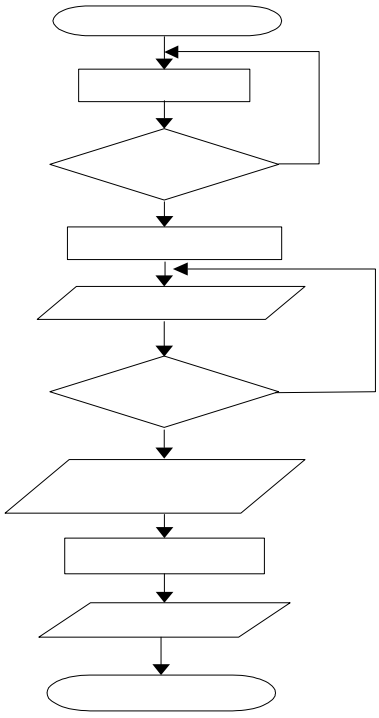
Tabel 3.6 Perintah AT Command

No	Perintah	Fungsi
1	ATE1	Aktifkan Perintah
2	AT+CGMI	Perintah aktif Manufaktur ID Kode
3	AT+CGMM	Perintah Model ID
4	AT+CGMR	Perintah Versi telepon GSM
5	AT+CGSN	Perintah Serial Number (IMEI)
6	AT+CSQ	Perintah Kualiti Sinyal
7	AT+CPMS	Perintauh data memori SMS
8	AT+CMGF	Perintah Format SMS
9	AT+CNMI	Perintah Tampilan SMS Masuk
10	AT+CMTI	Perintah Indikasi SMS Masuk
11	AT+CMGS	Perintah Mengirim SMS
12	AT+CMS	Perintah Pesan SMS
13	AT+CMGR	Perintah Membaca SMS

14	AT+CBC	Perintah Batre Charge
15	AT+CMGD	Perintah Menghapus SMS
16	AT+CHUP	Perintah Terminate Call

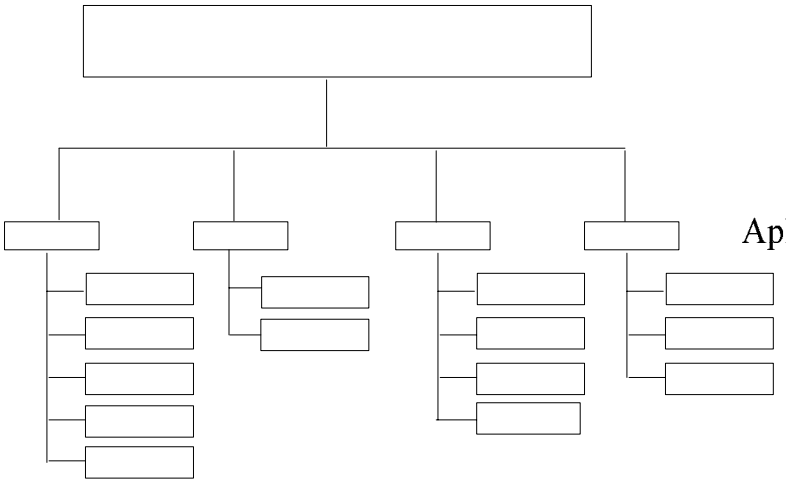
PERANCANGAN PROSEDURAL

perancangan prosedural untuk aplikasi yang akan dibangun adalah sebagai berikut :



Gambar 3.4 Perancangan Prosedural

STRUKTUR MENU PROGRAM

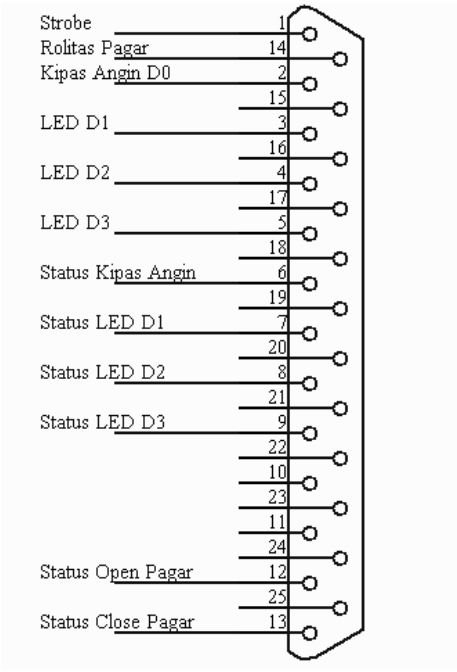


Gambar 3.5 Struktur Menu Program

TABEL KONEKTOR DB 25

DB 25	IN/OUT	NAMA SINYAL	REGBIT	KETERANGAN
1	Out	nStrobe	C0	Rolitas Pagar
2	Out	Data 0	D0	Kipas Angin
3	Out	Data 1	D1	LED
4	Out	Data 2	D2	LED
5	Out	Data 3	D3	LED
6	Out	Data 4	D4	Status Kipas Angin
7	Out	Data 5	D5	Status LED
8	Out	Data 6	D6	Status LED
9	Out	Data 7	D7	Status LED
10				
11				
12	In	Paperend	S 5+	Status Open Pagar
13	In	Select	S 4+	Status Close Pagar
14	Out	n Auto Feed	C1 -	Rolitas Pagar
15				
16				
17				
18-25	Ground	Ground		Ground

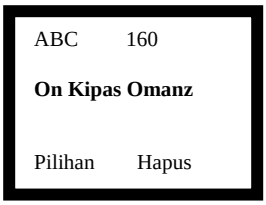
SKEMA KONEKTOR DB-25 PADA RANGKAIAN



Gambar.3.7 Bagian Konektor DB-25

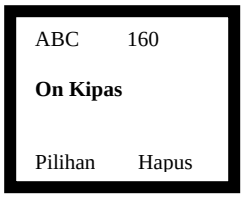
PENGUJIAN PROGRAM PERTAMA

Pada pengujian program pertama ini User mengirimkan kode yang benar ke komputer server gateway misalkan kode D0 untuk Kipas Angin. Berikut penulis mengambil contoh dengan mengirimkan kode D0 pada komputer server gateway. Berikut ini tampilan tampilan yang dikirimkan oleh User.



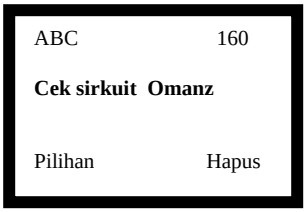
PENGUJIAN PROGRAM KEDUA

Pada pengujian program kedua ini User mengirimkan kode yang salah ke komputer server gateway misalkan User mengirimkan pesan SMS tapi tidak diakhiri dengan password atau password salah.



PENGUJIAN PROGRAM KETIGA

Pada pengujian ketiga User mengirimkan Cek sirkuit atau cek status sirkuit ke komputer server gateway, apakah sirkuit keadaan menyala atau mati.



ANALISIS HASIL PENGUJIAN

Berdasarkan hasil pengujian dengan kasus uji diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

KESIMPULAN

Berdasarkan hasil penelitian yang telah penulis lakukan, dapat diambil beberapa kesimpulan, diantaranya sebagai berikut:

1. Aplikasi pengontrolan peralatan listrik rumah melalui SMS sesuai yang diharapkan.
2. Aplikasi pengontrolan peralatan listrik rumah secara automatis berhasil dilakukan.

3. Aplikasi pengontrolan peralatan listrik rumah secara manual berhasil dilakukan.
4. Pada aplikasi pengontrolan peralatan listrik rumah ini sangat bermanfaat karena dapat mengurangai pemborosan dalam pemakaian peralatan listrik.
5. Pengiriman laporan status sirkuit ke user berhasil dilakukan.

SARAN

Penulis akan memberikan beberapa saran bagi pengembang yang hendak mengembangkan teknologi ini lebih lanjut, yaitu sebagai berikut :

1. Aplikasi pengontrolan peralatan listrik rumah berbasis SMS ini dapat dikembangkan dengan Micro Controller.
2. Aplikasi pengontrolan peralatan listrik rumah berbasis SMS ini hanya membuat simulasi miniatur rumah yang sederhana saja, sehingga dapat di kembangkan sebagai ajuan untuk memberikan tegangan yang lebih besar dan tinggi pada sirkuit.

DAFTAR PUSTAKA

1. Michel Roger.”Linear Data Book”,1976 *National SemiConduktor(USA).Ltd.*
2. Pressman,Roger.S.”Software Engineering : A Practioner’s Approach.” 5 th.McGrawHill.2001
3. Pressman Roger S Phd, 2000, *Rekayasa Perangkat Lunak,*Yogyakarta;Andi
4. Turing, Alan. “*Computing Machinery and Intelligence*”, 1950
5. *Text To Speech Technology*. (*http://www.mbrola.com*)

Extreme Programming. (*www.extremeprogramming.org*

LAMPIRAN

```
procedure TMainForm.FormCreate(Sender: TObject);
var
    bErr: byte;
begin
    IniFileName := ExtractFilePath(Application.ExeName)+'omanz.ini';
    ControlNo := 0;
    bErr:=(Out32(DATA,ControlNo));

    ListSMS := TStringList.Create;
    ListCMD := TStringList.Create;
    ListCMD.Add('ATE1');
    ListCMD.Add('AT+CGMI');
    ListCMD.Add('AT+CGMM');
    ListCMD.Add('AT+CGMR');
    ListCMD.Add('AT+CGSN');
    ListCMD.Add('AT+CBC');
    ListCMD.Add('AT+CSQ');
    ListCMD.Add('AT+CPMS="ME","ME","ME"');
    ListCMD.Add('AT+CMGF=0');
    ListCMD.Add('AT+CNMI=1,1,0,2,1');
end;

procedure TMainForm.btnMouseEnter(Sender: TObject);
begin
    FlatSound.Play;
end;

procedure TMainForm.btnMouseClick(Sender: TObject);
begin
    Midi.Play;
end;

procedure TMainForm.About(Sender: TObject);
begin
    FormAbout.ShowModal;
end;

procedure TMainForm.ClosePanel;
begin
    while MainForm.Width > pnControl.Width do
    begin
        MainForm.Width := MainForm.Width - 20;
        MainForm.Repaint;
    end;
end;

procedure TMainForm.OpenPanel;
begin
    while MainForm.Width < (pnControl.Width + 575) do
    begin
        MainForm.Width := MainForm.Width + 10;
        MainForm.Repaint;
    end;
end;

procedure TMainForm.HidePanel;
begin
    pnDevice.Hide;
```

```
    pnMessage.Hide;
    pnSettings.Hide;
    pnDatabase.Hide;
end;

procedure TMainForm.btnDeviceClick(Sender: TObject);
begin
    if (not pnDevice.Visible) then
    begin
      ClosePanel;
      HidePanel;
      pnDevice.Show;
      OpenPanel;
    end;
end;

procedure TMainForm.btnMessageClick(Sender: TObject);
begin
    if (not pnMessage.Visible) then
    begin
      ClosePanel;
      HidePanel;
      pnMessage.Show;
      OpenPanel;
    end;
end;

procedure TMainForm.btnSettingsClick(Sender: TObject);
begin
    if (not pnSettings.Visible) then
    begin
      ClosePanel;
      HidePanel;
      pnSettings.Show;
      OpenPanel;
    end;
end;

procedure TMainForm.btnDatabaseClick(Sender: TObject);
begin
    if (not pnDatabase.Visible) then
    begin
      ClosePanel;
      HidePanel;
      pnDatabase.Show;
      OpenPanel;
    end;
end;

procedure TMainForm.Pagar(Open: Boolean);
begin
    if (Open) then
        out32(CONTROL,2)
    else
        out32(CONTROL,0);
end;

procedure TMainForm.LightOnOff(BitPos: integer; Off: Boolean);
var
```

```
    MyByte : string;
    bErr:byte;
begin
    Out32(CONTROL, Inp32(CONTROL) and $DF);
    Out32(CONTROL, Inp32(CONTROL) and $F7);
    bErr:=(Out32(DATA,0));
    Sleep(100);
    MyByte := HexToBin(IntToHex(ControlNo,2));

    case BitPos of
        0 : begin
                if (Off) then
                    MyByte := '00000000'
                else
                    MyByte := '00001111';
            end;
        1 : begin
                if (Off) then
                    MyByte[8] := '0'
                else
                    MyByte[8] := '1';
            end;
        2 : begin
                if (Off) then
                    MyByte[7] := '0'
                else
                    MyByte[7] := '1';
            end;
        3 : begin
                if (Off) then
                    MyByte[6] := '0'
                else
                    MyByte[6] := '1';
            end;
        4 : begin
                if (Off) then
                    MyByte[5] := '0'
                else
                    MyByte[5] := '1';
            end;
    end;

    ControlNo := BinStrToInt(MyByte);
    bErr:=(Out32(DATA,ControlNo));
    out32(CONTROL,inp32(CONTROL) or $01);
end;

procedure TMainForm.btnOnOffD0Click(Sender: TObject);
begin
    Midi.Play;
    if (UpperCase(btnOnOffD0.Caption) = 'ON') then
    begin
        btnOnOffD0.Caption := 'OFF';
        btnOnOffD0.Font.Color := clRed;
        Led0.Value := true;
        LightOnOff(1,false);
    end else
    begin
        btnOnOffD0.Caption := 'ON';
```

```
        btnOnOffD0.Font.Color := clNavy;
        Led0.Value := false;
        LightOnOff(1,true);
    end;
end;

procedure TMainForm.btnOnOffD1Click(Sender: TObject);
begin
    Midi.Play;
    if (UpperCase(btnOnOffD1.Caption) = 'ON') then
    begin
        btnOnOffD1.Caption := 'OFF';
        btnOnOffD1.Font.Color := clRed;
        Led1.Value := true;
        LightOnOff(2,false);
    end else
    begin
        btnOnOffD1.Caption := 'ON';
        btnOnOffD1.Font.Color := clNavy;
        Led1.Value := false;
        LightOnOff(2,true);
    end;
end;

procedure TMainForm.btnOnOffD2Click(Sender: TObject);
begin
    Midi.Play;
    if (UpperCase(btnOnOffD2.Caption) = 'ON') then
    begin
        btnOnOffD2.Caption := 'OFF';
        btnOnOffD2.Font.Color := clRed;
        Led2.Value := true;
        LightOnOff(3,false);
    end else
    begin
        btnOnOffD2.Caption := 'ON';
        btnOnOffD2.Font.Color := clNavy;
        Led2.Value := false;
        LightOnOff(3,true);
    end;
end;

procedure TMainForm.btnOnOffD3Click(Sender: TObject);
begin
    Midi.Play;
    if (UpperCase(btnOnOffD3.Caption) = 'ON') then
    begin
        btnOnOffD3.Caption := 'OFF';
        btnOnOffD3.Font.Color := clRed;
        Led3.Value := true;
        LightOnOff(4,false);
    end else
    begin
        btnOnOffD3.Caption := 'ON';
        btnOnOffD3.Font.Color := clNavy;
        Led3.Value := false;
        LightOnOff(4,true);
    end;
end;
```