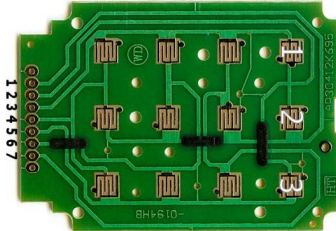
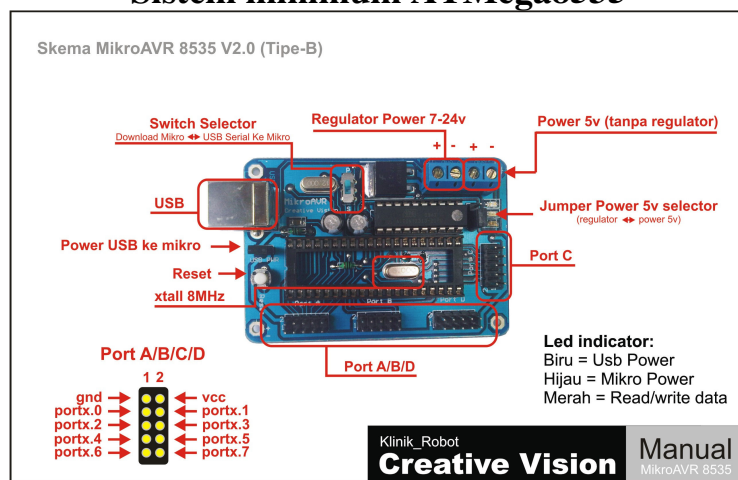


LAMPIRAN

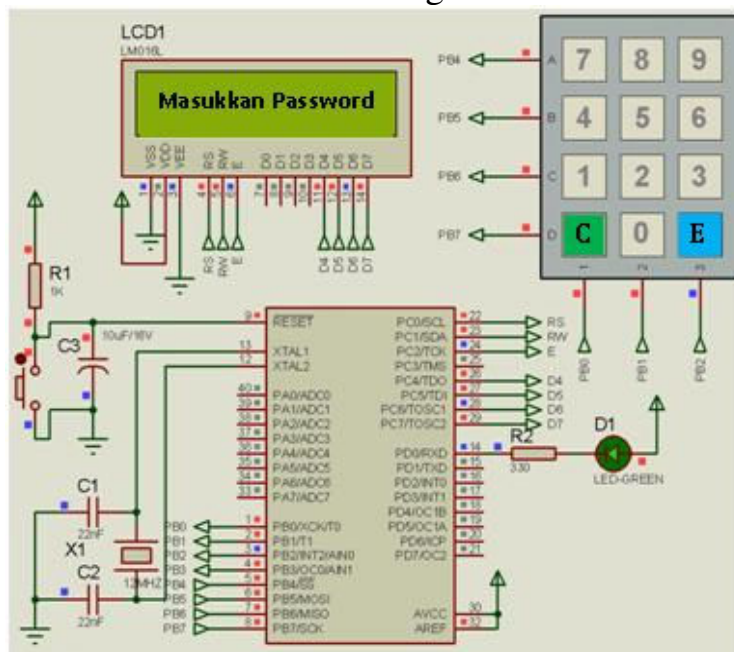
Layout PCB keypad



Sistem minimum ATmega8535



Skematik rangkaian



PROGRAM MIKROKONTROLLER

/*****

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 18/07/2012
Author : Sh.mukti
Company : Jangan gunakan software bajakan!
Comments:

Chip type : ATmega8535
Program type : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 128

*****/

```
#include <mega8535.h>
#include <alcd.h>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>
```

```
float nilai;
unsigned char temp[12], array[10], i=0;
int data_rx;
```

```
void simpan_dlm_1variabel()
{
    if (i==1){nilai=array[i];}
```

```

if (i>=2 && i<=7)
{
    nilai=(nilai*10)+array[i];
}

if (i>7)
{
    lcd_clear();
    delay_ms(100);

    lcd_gotoxy(0,0);
    lcd_putsf("Pasword Maksimal");

    lcd_gotoxy(4,1);
    lcd_putsf("7 Digit");

    delay_ms(3000);

    i=0;
    nilai=0;
    lcd_clear();
    delay_ms(200);
}
}

void scanning_keypad()
{
    lcd_gotoxy(0,0);
    lcd_putsf("Masukan Pasword:");

    PORTA = 0b11111110;
    delay_ms(30);
    if (PINA.4 == 0) {i++; array[i]=1; simpan_dlm_1variabel();
delay_ms(100);}
    if (PINA.5 == 0) {i++; array[i]=2; simpan_dlm_1variabel();
delay_ms(100);}
    if (PINA.6 == 0) {i++; array[i]=3; simpan_dlm_1variabel();
delay_ms(100);}
}

```

```

PORTA = 0b11111101;
delay_ms(30);
if (PINA.4 == 0) {i++; array[i]=4; simpan_dlm_lvariabel();
delay_ms(100);}
if (PINA.5 == 0) {i++; array[i]=5; simpan_dlm_lvariabel();
delay_ms(100);}
if (PINA.6 == 0) {i++; array[i]=6; simpan_dlm_lvariabel();
delay_ms(100);}

PORTA = 0b11111011;
delay_ms(30);
if (PINA.4 == 0) {i++; array[i]=7; simpan_dlm_lvariabel();
delay_ms(100);}
if (PINA.5 == 0) {i++; array[i]=8; simpan_dlm_lvariabel();
delay_ms(100);}
if (PINA.6 == 0) {i++; array[i]=9; simpan_dlm_lvariabel();
delay_ms(100);}

PORTA = 0b11110111;
delay_ms(30);
if (PINA.4 == 0) {nilai=0; i=0; lcd_clear(); delay_ms(100);}
if (PINA.5 == 0) {i++; array[i]=0; simpan_dlm_lvariabel();
delay_ms(100);}
if (PINA.6 == 0) {printf ("%0.0f",nilai); delay_ms(300);
data_rx=getchar(); }
}

void tampil_lcd()
{
if (nilai>0)
{
ftoa(nilai,0,temp);
lcd_gotoxy(0,1);
lcd_puts(temp);
}
}

```

```

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In Func2=In
Func1=In Func0=In
// State7=1 State6=1 State5=1 State4=1 State3=P State2=P State1=P
State0=P
PORTA=0xFF;
DDRA=0x0F;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=Out Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=1 State4=T State3=T State2=P State1=T
State0=T
PORTD=0x24;
DDRD=0x20;

// Timer/Counter 0 initialization

```

```
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
```

```
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;
```



```

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 8
lcd_init(16);
lcd_gotoxy(0,0);
  lcd_putsf("Tomi Wijaya");
  delay_ms(2500);
  lcd_clear();

while (1)
{
  PORTD=0x24;

  scanning_keypad();
  tampil_lcd();

  if (data_rx==49)
  {
    PORTD.5=0;
    lcd_clear();
    lcd_gotoxy(1,0);
    lcd_putsf("Pasword Benar");
    lcd_gotoxy(1,1);
    lcd_putsf("Silahkan Masuk");
    delay_ms(3000);
  }
}

```

```
lcd_clear();  
delay_ms(200);
```

```
lcd_gotoxy(3,0);  
lcd_putsf("Waktu Anda");  
lcd_gotoxy(2,1);  
lcd_putsf("Hanya 5 Detik");  
delay_ms(1000);
```

```
lcd_gotoxy(3,0);  
lcd_putsf("Waktu Anda");  
lcd_gotoxy(2,1);  
lcd_putsf("Hanya 4 Detik");  
delay_ms(1000);
```

```
lcd_gotoxy(3,0);  
lcd_putsf("Waktu Anda");  
lcd_gotoxy(2,1);  
lcd_putsf("Hanya 3 Detik");  
delay_ms(1000);
```

```
lcd_gotoxy(3,0);  
lcd_putsf("Waktu Anda");  
lcd_gotoxy(2,1);  
lcd_putsf("Hanya 2 Detik");  
delay_ms(1000);
```

```
lcd_gotoxy(3,0);  
lcd_putsf("Waktu Anda");  
lcd_gotoxy(2,1);  
lcd_putsf("Hanya 1 Detik");  
delay_ms(1000);
```

```
lcd_gotoxy(3,0);  
lcd_putsf("Pintu Akan");  
lcd_gotoxy(0,1);  
lcd_putsf("Terkunci Kembali");
```

```
    delay_ms(2000);

    nilai=0;
    i=0;
    data_rx=0;
    lcd_clear();
    delay_ms(100);
}

if (data_rx==50)
{
    lcd_clear();
    lcd_gotoxy(2,0);
    lcd_putsf("Pasword Anda");

    lcd_gotoxy(5,1);
    lcd_putsf("SALAH");
    delay_ms(3000);

    nilai=0;
    i=0;
    data_rx=0;
    lcd_clear();
    delay_ms(100);
}
if (PIND.2==0)
{
    PORTD.5=0;
    delay_ms(5000);
    PORTD=0x24;
}
}
```

Program Delphi 7 Enterprise

//Program Project1

Program Project1;

Uses

```
Forms,  
Unit1 In 'Unit1.Pas' {Form1},  
Unit2 In 'Unit2.Pas' {Form2},  
Unit3 In 'Unit3.Pas' {Form3},  
Unit4 In 'Unit4.Pas' {Form4},  
Unit5 In 'Unit5.Pas' {Form5},  
Unit6 In 'Unit6.Pas' {Form6};
```

{\$R *.Res}

Begin

```
Application.Initialize;  
Application.Createform(Tform1, Form1);  
Application.Createform(Tform2, Form2);  
Application.Createform(Tform3, Form3);  
Application.Createform(Tform4, Form4);  
Application.Createform(Tform5, Form5);  
Application.Createform(Tform6, Form6);  
Application.Run;
```

End.

//Program Form Utama

Unit Unit1;

Interface

Uses

```
Windows, Messages, Sysutils, Variants, Classes, Graphics,  
Controls, Forms,  
Dialogs, Dbctrls, Stdctrls, Jpeg, Extctrls, Cport, Grids,  
Buttons,  
Cportctl;
```

Type

```
Tform1 = Class(Tform)  
  Image1: Timage;  
  Label1: Tlabel;  
  Button1: Tbutton;  
  Comled1: Tcomled;  
  Button2: Tbutton;  
  Button3: Tbutton;  
  Button4: Tbutton;  
  Button5: Tbutton;  
  Procedure Button1click(Sender: Tobject);  
  Procedure Button2click(Sender: Tobject);  
  Procedure Button3click(Sender: Tobject);  
  Procedure Button4click(Sender: Tobject);  
  Procedure Button5click(Sender: Tobject);
```

```

Private
  { Private Declarations }
Public
  { Public Declarations }
End;

Var
  Form1: TForm1;

Implementation

Uses Unit2, Unit3;

{$R *.Dfm}

Procedure TForm1.Button1click(Sender: TObject);
Begin
  If      Messagedlg('Anda      Yakin?',Mtconfirmation,[Mbytes,
Mbno],0)=Mryes Then
  Begin
    Form2.Comport1.Connected := False;
    Application.Terminate;
  End;
End;

Procedure TForm1.Button2click(Sender: TObject);
Begin
  Form2.Comport1.Showsetupdialog;
End;

Procedure TForm1.Button3click(Sender: TObject);
Begin
  If (Form2.Comport1.Connected = False) Then
  Begin
    Form2.Comport1.Connected := True;
    Button3.Caption := 'Disconnect';
  End
  Else
  Begin
    Form2.Comport1.Connected := False;
    Button3.Caption := 'Connect';
  End
End;

Procedure TForm1.Button4click(Sender: TObject);
Begin
  Form1.Hide;
  Form3.Show;
End;

Procedure TForm1.Button5click(Sender: TObject);
Begin
  Form1.Hide;
  Form2.Show;
End;
End.

```

//Program Form Umum

Unit Unit2;

Interface

Uses

Windows, Messages, Sysutils, Variants, Classes, Graphics,
Controls, Forms,
Dialogs, Grids, Dbgrids, Jpeg, ExtCtrls, Db, Adodb, StdCtrls,
Cport;

Type

```
Tform2 = Class(Tform)
  Dbgrid1: Tdbgrid;
  Button1: Tbutton;
  Comport1: Tcomport;
  Mem1: Tmemo;
  Datasource2: Tdatasource;
  Datasource1: Tdatasource;
  Adoconnection1: Tadoconnection;
  Adoquery1: Tadoquery;
  Adoquerylno: Tautoincfield;
  Adoqueryltanggal: Tdatetimefield;
  Adoqueryljammasuk: Tdatetimefield;
  Adoquerylid_Karyawan: Tintegerfield;
  Adoquerylnama: Twidestringfield;
  Adoquerylbagian: Twidestringfield;
  Adoquerylnotelp: Twidestringfield;
  Adoquerylalamat: Twidestringfield;
  Adoquerylketerangan: Twidestringfield;
  Adoqueryljabatan: Twidestringfield;
  Image1: Timage;
  Procedure Button1click(Sender: Tobject);
  Procedure Comport1rxchar(Sender: Tobject; Count: Integer);
```

Private

```
{ Private Declarations }
```

Public

```
{ Public Declarations }
```

End;

Var

```
Form2: Tform2;
```

Implementation

Uses Unit1, Unit5, Unit4;

```
{ $R *.Dfm }
```

```
Procedure Tform2.Button1click(Sender: Tobject);
```

```
Begin
```

```
  Form2.Hide;
```

```
  Form1.Show;
```

```
End;
```

```

Procedure TForm2.Comport1rxchar(Sender: TObject; Count: Integer);
Var
Str: String;
Id : Integer;
Begin
Comport1.Readstr(Str,Count);
Memol.Text := Memol.Text + Str;
If (Length(Memol.Text) >=6) Then
Begin
If Form4.Adokaryawan.Locate('Password', Memol.Text,
[Locaseinsensitive]) Then
Begin
Id := Form4.Adokaryawan.Fieldbyname('Id').Value;
Showmessage('Ada Pengguna Baru Dengan Id '+ Inttostr(Id) +
'!!!!....');
Adoquery1.Append;
Adoquery1.Fieldbyname('Id_Karyawan').Value:= Id;
Adoquery1.Fieldbyname('Tanggal').Value :=Datetostr(Now);
Adoquery1.Fieldbyname('Jam Masuk').Value :=Timetostr(Now);
Adoquery1.Post;
Adoquery1.Close;
Adoquery1.Open;
Comport1.Writestr('1');
Memol.Clear;
End
Else
Begin
Showmessage('Ada Pengguna Baru Gagal Masuk!!!!....');
Comport1.Writestr('2');
Memol.Clear;
End;
End;
End;
End.

```

//Program Form Log-In

```

Unit Unit3;

Interface

Uses
Windows, Messages, Sysutils, Variants, Classes, Graphics,
Controls, Forms,
Dialogs, Jpeg, Extctrls, Stdctrls, Db, Adodb;

Type
Tform3 = Class(Tform)
Image1: Timage;
Edit1: Tedit;
Edit2: Tedit;
Button1: Tbutton;
Button2: Tbutton;
Label1: Tlabel;
Label2: Tlabel;
Datasource1: Tdatasource;
Procedure Button2click(Sender: TObject);

```

```

    Procedure Button1click(Sender: Tobject);
Private
    { Private Declarations }
Public
    { Public Declarations }
End;

Var
    Form3: TForm3;

Implementation

Uses Unit1, Unit4;

{$R *.Dfm}

Procedure TForm3.Button2click(Sender: Tobject);
Begin
    Edit1.Clear;
    Edit2.Clear;
    Form1.Show;
    Form3.Hide;
End;

Procedure TForm3.Button1click(Sender: Tobject);
Begin
    If (Length(Edit1.Text) >=1) Then
    If (Length(Edit2.Text) >=1) Then
    Begin
        If Form4. Adokaryawan. Locate ('Username', Edit1. Text,
        [Locaseinsensitive]) Then
        If Form4.Adokaryawan.Fieldbyname('Password Admin').Asstring =
        Edit2.Text Then
        Begin
            Showmessage('Login Berhasil');
            Edit1.Clear;
            Edit2.Clear;
            Hide;
            Form4.Show;
        End
        Else
        Begin
            Showmessage('Password Salah');
            Edit2.Clear;
        End
        Else
            Showmessage('Username Tidak Terdaftar');
            Edit1.Clear;
            Edit2.Clear;
        End;
    End;
End.

//Program Form Admin
Unit Unit4;

```


Interface

Uses

```
Windows, Messages, Sysutils, Variants, Classes, Graphics,  
Controls, Forms,  
Dialogs, Grids, Dbgrids, Db, Adodb, Jpeg, Extctrls, Menus,  
Stdctrls,  
Extdlgs, Dbctrls;
```

Type

```
Tform4 = Class(Tform)  
  Adoconnection1: Tadoconnection;  
  Adokaryawan: Tadoable;  
  Datasource1: Tdatasource;  
  Dbgrid1: Tdbgrid;  
  Button5: Tbutton;  
  Button1: Tbutton;  
  Button2: Tbutton;  
  Button3: Tbutton;  
  Label1: TLabel;  
  Edit1: Tedit;  
  Image1: Timage;  
  Button4: Tbutton;  
  Procedure Keluar1click(Sender: TObject);  
  Procedure Button2click(Sender: TObject);  
  Procedure Formactivate(Sender: TObject);  
  Procedure Formclose(Sender: TObject; Var Action:  
Tcloseaction);  
  Procedure Button5click(Sender: TObject);  
  Procedure Button1click(Sender: TObject);  
  Procedure Button3click(Sender: TObject);  
  Procedure Dbgrid1dblclick(Sender: TObject);  
  Procedure Button4click(Sender: TObject);  
  
  Private  
    { Private Declarations }  
  Public  
    { Public Declarations }  
End;
```

Var

```
Form4: Tform4;  
Ada: Integer;
```

Implementation

```
Uses Unit1, Unit5, Unit6;
```

```
{ $R *.Dfm }
```

```
Procedure Tform4.Keluar1click(Sender: TObject);
```

```
Begin
```

```
  If Messagedlg('Yakin Anda Ingin Keluar Dari Form Ini? ?',  
Mtconfirmation, [Mbyes, Mbno], 0) = Mryes Then  
  Begin
```

```

        Form4.Hide;
        Form1.Show;
    End;
End;

Procedure TForm4.Button2click(Sender: TObject);
Begin
    If Messagedlg('Hapus Data ?', Mtconfirmation,
        [Mbytes,Mbno],0)=Mryes Then
        Adokaryawan.Delete;
    End;
End;

Procedure TForm4.Formactivate(Sender: TObject);
Begin
    Form4.Adokaryawan.Open;
End;

Procedure TForm4.Formclose(Sender: TObject; Var Action:
Tcloseaction);
Begin
    Form4.Adokaryawan.Close;
End;

Procedure TForm4.Button5click(Sender: TObject);
Begin
    If Messagedlg('Yakin Ingin Keluar Dari Form Ini ?',
        Mtconfirmation, [Mbytes,Mbno],0)=Mryes Then
        Begin
            Form4.Hide;
            Form1.Show;
        End;
    End;
End;

Procedure TForm4.Button1click(Sender: TObject);
Begin
    Form5.Show;
    Form4.Hide;
End;

Procedure TForm4.Button3click(Sender: TObject);
Begin
    If Ada =0 Then
        Adokaryawan.First
    Else
        Adokaryawan.Next;
    While Not Adokaryawan.Eof Do
        Begin
            Ada:=Pos(Uppercase(Edit1.Text),Uppercase(Adokaryawan['Nama']
                ));
            If Ada>0 Then
                Break;
            Adokaryawan.Next;
        End;
    If Ada =0 Then
        Begin
            Beep;
        End;
    End;
End;

```

```

        Adokaryawan.First;
    End;
End;

Procedure TForm4.Dbgrid1dblclick(Sender: TObject);
Begin
    If Messagedlg('Apakah Anda Ingin Merubah Data Ini?',
Mtconfirmation, [Mbytes, Mbno], 0)=Mryes Then
    Begin
        Form4.Hide;
        Form6.Show;
        Form6.Edit1.Text := Adokaryawan.Fields[1].AsString;
        Form6.Edit2.Text := Adokaryawan.Fields[2].AsString;
        Form6.Edit3.Text := Adokaryawan.Fields[3].AsString;
        Form6.Edit4.Text := Adokaryawan.Fields[4].AsString;
        Form6.Edit5.Text := Adokaryawan.Fields[5].AsString;
        Form6.Edit6.Text := Adokaryawan.Fields[7].AsString;
        Form6.Edit7.Text := Adokaryawan.Fields[8].AsString;
        Form6.Edit8.Text := Adokaryawan.Fields[9].AsString;
        Form6.Edit9.Text := Adokaryawan.Fields[6].AsString;
    End;
End;

Procedure TForm4.Button4click(Sender: TObject);
Begin
    If Messagedlg('Apakah Anda Ingin Merubah Data Ini?',
Mtconfirmation, [Mbytes, Mbno], 0)=Mryes Then
    Begin
        Form4.Hide;
        Form6.Show;
        Form6.Edit1.Text := Adokaryawan.Fields[1].AsString;
        Form6.Edit2.Text := Adokaryawan.Fields[2].AsString;
        Form6.Edit3.Text := Adokaryawan.Fields[3].AsString;
        Form6.Edit4.Text := Adokaryawan.Fields[4].AsString;
        Form6.Edit5.Text := Adokaryawan.Fields[5].AsString;
        Form6.Edit6.Text := Adokaryawan.Fields[7].AsString;
        Form6.Edit7.Text := Adokaryawan.Fields[8].AsString;
        Form6.Edit8.Text := Adokaryawan.Fields[9].AsString;
        Form6.Edit9.Text := Adokaryawan.Fields[6].AsString;
    End;
End;

End.

```

//Program Form Tambah

```
Unit Unit5;
```

```
Interface
```

```
Uses
```

```
Windows, Messages, Sysutils, Variants, Classes, Graphics,
Controls, Forms,
```

```
Dialogs, StdCtrls, Grids, Dbgrids, Db, Adodb, Jpeg, ExtCtrls;
```

```
Type
```

```
Tform5 = Class(Tform)
```

```
    Button1: Tbutton;
```

```

    Datasource1: Tdatasource;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Edit1: Tedit;
    Edit2: Tedit;
    Edit3: Tedit;
    Edit4: Tedit;
    Edit5: Tedit;
    Edit6: Tedit;
    Edit7: Tedit;
    Edit8: Tedit;
    Edit9: Tedit;
    Edit10: Tedit;
    Button2: Tbutton;
    Button3: Tbutton;
    Image1: Timage;
    Procedure Button2click(Sender: TObject);
    Procedure Button1click(Sender: TObject);
    Procedure Button3click(Sender: TObject);
Private
    { Private Declarations }
Public
    { Public Declarations }
End;

Var
    Form5: TForm5;

Implementation

Uses Unit4;

{$R *.Dfm}

Procedure TForm5.Button2click(Sender: TObject);
Begin
    Edit1.Clear;
    Edit2.Clear;
    Edit3.Clear;
    Edit4.Clear;
    Edit5.Clear;
    Edit6.Clear;
    Edit7.Clear;
    Edit8.Clear;
    Edit9.Clear;
    Edit10.Clear;
End;

```

```

Procedure TForm5.Button1click(Sender: TObject);
Begin
  If Messagedlg('Yakin Ingin Keluar Dari Form Ini ?',
    Mtconfirmation, [Mbyes,Mbno],0)=Mryes Then
    Begin
      Form5.Hide;
      Form4.Show;
    End;
End;

```

```

Procedure TForm5.Button3click(Sender: TObject);
Begin
  Form4.Adokaryawan.Append;
  Form4.Adokaryawan['Id']:=Edit1.Text;
  Form4.Adokaryawan['Nama']:=Edit2.Text;
  Form4.Adokaryawan['Bagian']:=Edit3.Text;
  Form4.Adokaryawan['Jabatan']:=Edit4.Text;
  Form4.Adokaryawan['No Telp']:=Edit5.Text;
  Form4.Adokaryawan['Alamat']:=Edit6.Text;
  Form4.Adokaryawan['Username']:=Edit7.Text;
  Form4.Adokaryawan['Password Admin']:=Edit8.Text;
  Form4.Adokaryawan['Password']:=Edit9.Text;
  Form4.Adokaryawan['Keterangan']:=Edit10.Text;
  Form4.Adokaryawan.Post;
  Refresh;
  Button2click(Self);
End;

```

End.

//Program Form Ubah

```
Unit Unit6;
```

```
Interface
```

```
Uses
```

```
  Windows, Messages, Sysutils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Db, Jpeg, ExtCtrls;
```

```
Type
```

```
  TForm6 = Class(Tform)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Edit1: Tedit;
    Edit2: Tedit;
    Edit3: Tedit;
    Edit4: Tedit;
```

```

    Edit5: Tedit;
    Edit6: Tedit;
    Edit7: Tedit;
    Edit8: Tedit;
    Edit9: Tedit;
    Button1: Tbutton;
    Button2: Tbutton;
    Button3: Tbutton;
    Button4: Tbutton;
    Datasource1: Tdatasource;
    Image1: Timage;
    Procedure Button3click(Sender: Tobject);
    Procedure Button2click(Sender: Tobject);
    Procedure Button1click(Sender: Tobject);
    Procedure Button4click(Sender: Tobject);
Private
  { Private Declarations }
Public
  { Public Declarations }
End;

Var
  Form6: TForm6;

Implementation

Uses Unit4;

{$R *.Dfm}

Procedure TForm6.Button3click(Sender: Tobject);
Begin
  Form6.Hide;
  Form4.Show;
End;

Procedure TForm6.Button2click(Sender: Tobject);
Begin
  Form6.Edit1.Text := Form4.Adokaryawan.Fields[1].AsString;
  Form6.Edit2.Text := Form4.Adokaryawan.Fields[2].AsString;
  Form6.Edit3.Text := Form4.Adokaryawan.Fields[3].AsString;
  Form6.Edit4.Text := Form4.Adokaryawan.Fields[4].AsString;
  Form6.Edit5.Text := Form4.Adokaryawan.Fields[5].AsString;
  Form6.Edit6.Text := Form4.Adokaryawan.Fields[7].AsString;
  Form6.Edit7.Text := Form4.Adokaryawan.Fields[8].AsString;
  Form6.Edit8.Text := Form4.Adokaryawan.Fields[9].AsString;
  Form6.Edit9.Text := Form4.Adokaryawan.Fields[6].AsString;
End;

Procedure TForm6.Button1click(Sender: Tobject);
Begin
  If MessageDlg('Apakah Anda Ingin Menyimpan Data Ini?',
Mtconfirmation, [Mbyes, Mbno], 0) = Mryes Then
  Begin
    Form4.Adokaryawan.Edit;
    Form4.Adokaryawan['Nama'] := Edit1.Text;

```

```
Form4.Adokaryawan['Bagian']:=Edit2.Text;
Form4.Adokaryawan['Jabatan']:=Edit3.Text;
Form4.Adokaryawan['No Telp']:=Edit4.Text;
Form4.Adokaryawan['Alamat']:=Edit5.Text;
Form4.Adokaryawan['Username']:=Edit6.Text;
Form4.Adokaryawan['Password Admin']:=Edit7.Text;
Form4.Adokaryawan['Password']:=Edit8.Text;
Form4.Adokaryawan['Keterangan']:=Edit9.Text;
Form4.Adokaryawan.Post;
Refresh;
Button4click(Self);
Form4.Show;
Form6.Hide;
End;
End;

Procedure TForm6.Button4click(Sender: TObject);
Begin
Form6.Edit1.Clear;
Form6.Edit2.Clear;
Form6.Edit3.Clear;
Form6.Edit4.Clear;
Form6.Edit5.Clear;
Form6.Edit6.Clear;
Form6.Edit7.Clear;
Form6.Edit8.Clear;
Form6.Edit9.Clear;
End;

End.
```

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 8K Bytes of In-System Self-Programmable Flash
Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
 - 512 Bytes EEPROM
Endurance: 100,000 Write/Erase Cycles
 - 512 Bytes Internal SRAM
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels for TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega8535L
 - 4.5 - 5.5V for ATmega8535
- Speed Grades
 - 0 - 8 MHz for ATmega8535L
 - 0 - 16 MHz for ATmega8535



**8-bit AVR[®]
Microcontroller
with 8K Bytes
In-System
Programmable
Flash**

**ATmega8535
ATmega8535L**

**Preliminary
Summary**

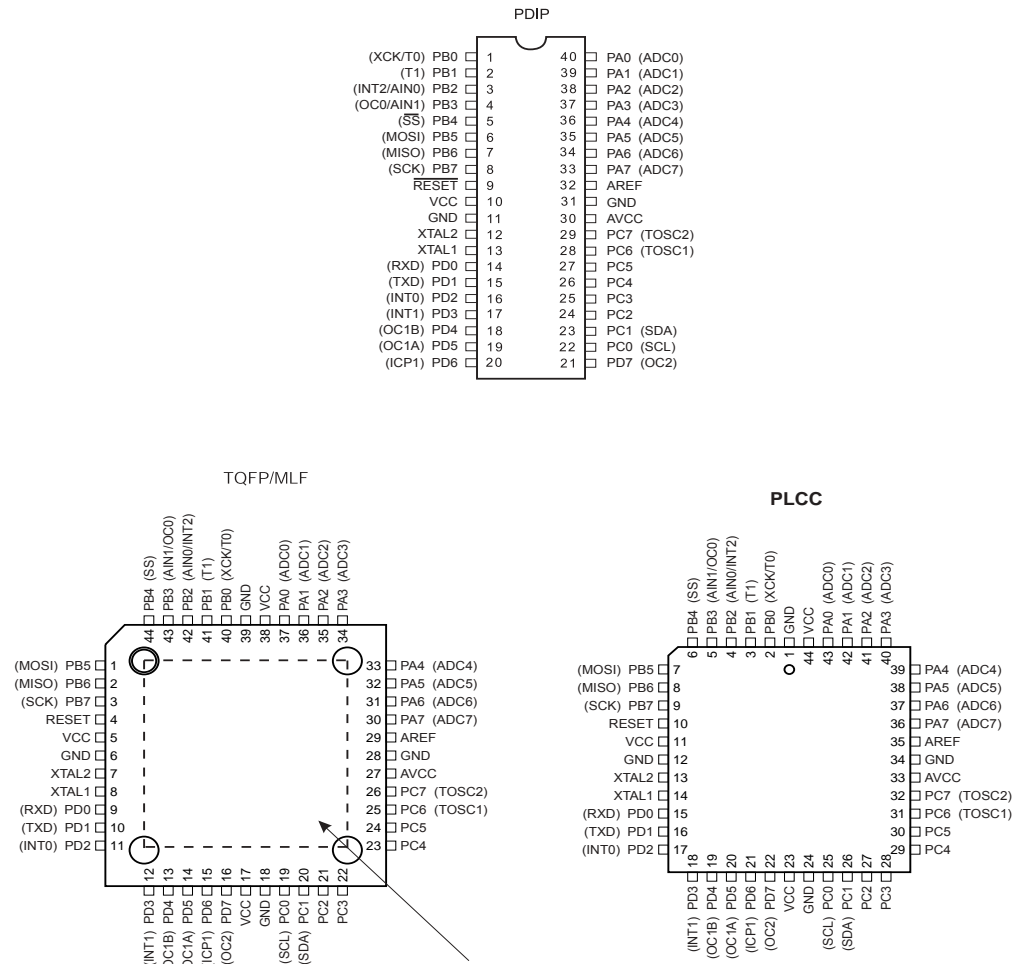
Rev. 2502ES-AVR-12/03



Note: This is a summary document. A complete document is available on our Web site at www.atmel.com.

Pin Configurations

Figure 1. Pinout ATmega8535



NOTE: MLF Bottom pad should be soldered to ground.

Disclaimer

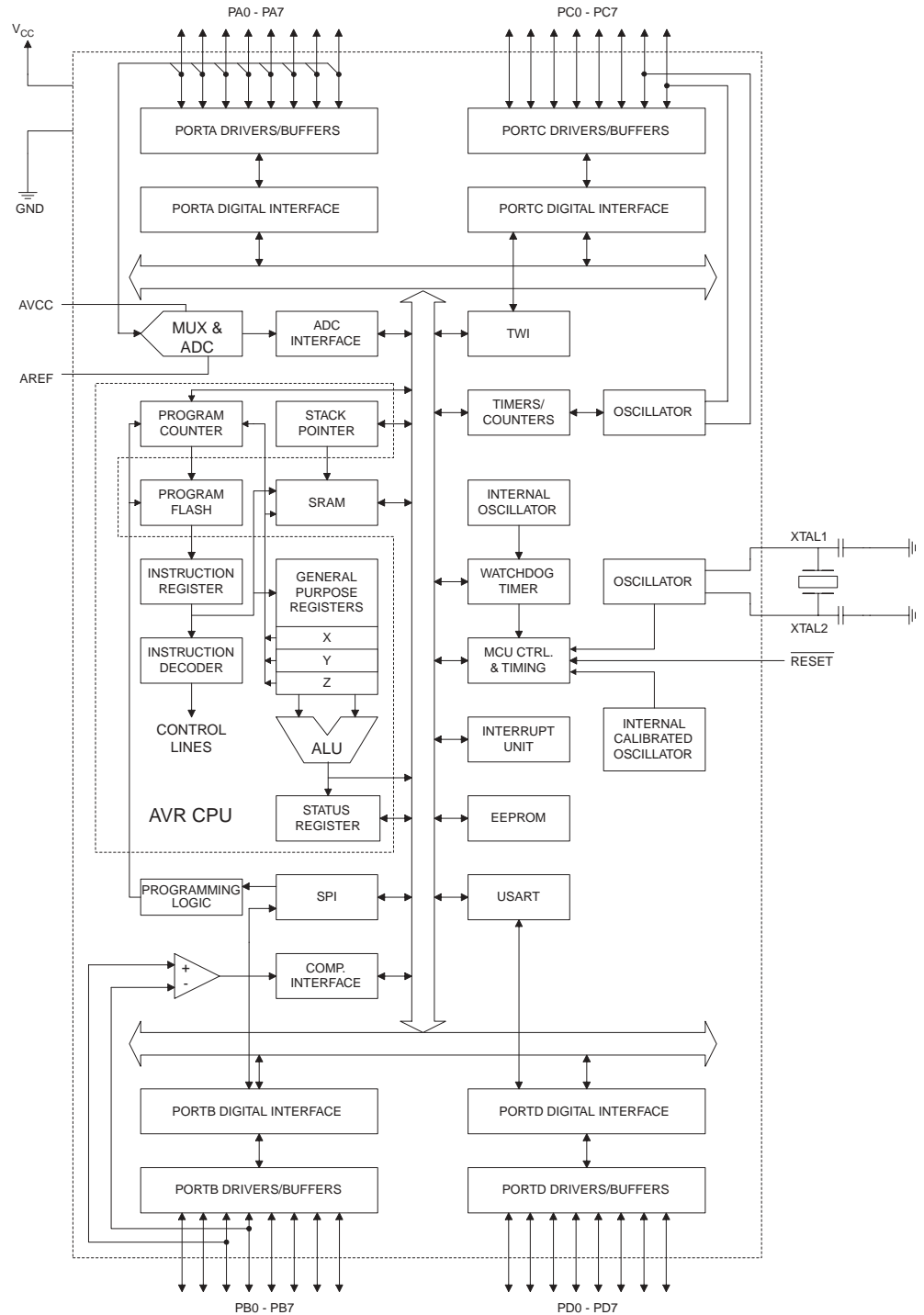
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8535 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain in TQFP package, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the asynchronous timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega8535 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

AT90S8535 Compatibility

The ATmega8535 provides all the features of the AT90S8535. In addition, several new features are added. The ATmega8535 is backward compatible with AT90S8535 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S8535 compatibility mode can be selected by programming the S8535C fuse. ATmega8535 is pin compatible with AT90S8535, and can replace the AT90S8535 on current Printed Circuit Boards. However, the location of fuse bits and the electrical characteristics differs between the two devices.

AT90S8535 Compatibility Mode

Programming the S8535C fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 43 for details.
- The double buffering of the USART Receive Register is disabled. See "AVR USART vs. AVR UART – Compatibility" on page 143 for details.

Pin Descriptions

V_{CC}	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	<p>Port A serves as the analog inputs to the A/D Converter.</p> <p>Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega8535 as listed on page 58.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega8535 as listed on page 62.</p>
<u>RESET</u>	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 35. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V _{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V _{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	8
0x3E (0x5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	10
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	10
0x3C (0x5C)	OCR0	Timer/Counter0 Output Compare Register								83
0x3B (0x5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	47, 67
0x3A (0x5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	68
0x39 (0x59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	83, 113, 131
0x38 (0x58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	84, 114, 132
0x37 (0x57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	225
0x36 (0x56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	178
0x35 (0x55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	30, 66
0x34 (0x54)	MCUCSR	–	ISC2	–	–	WDRF	BORF	EXTRF	PORF	38, 67
0x33 (0x53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	81
0x32 (0x52)	TCNT0	Timer/Counter0 (8 Bits)								83
0x31 (0x51)	OSCCAL	Oscillator Calibration Register								28
0x30 (0x50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	57,86,133,200,220
0x2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	108
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	111
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								112
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								112
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								112
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								112
0x29 (0x49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								112
0x28 (0x48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								112
0x27 (0x47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								112
0x26 (0x46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								112
0x25 (0x45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	126
0x24 (0x44)	TCNT2	Timer/Counter2 (8 Bits)								128
0x23 (0x43)	OCR2	Timer/Counter2 Output Compare Register								129
0x22 (0x42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	129
0x21 (0x41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	40
0x20 ⁽¹⁾ (0x40 ⁽¹⁾)	UBRRH	URSEL	–	–	–	–	UBRR[11:8]			166
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	164
0x1F (0x3F)	EEARH	–	–	–	–	–	–	–	EEAR8	17
0x1E (0x3E)	EEARL	EEPROM Address Register Low Byte								17
0x1D (0x3D)	EEDR	EEPROM Data Register								17
0x1C (0x3C)	EECR	–	–	–	–	EERIE	EEMWE	EWE	EERE	17
0x1B (0x3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	64
0x1A (0x3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	64
0x19 (0x39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	64
0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	64
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	64
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	65
0x15 (0x35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
0x14 (0x34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
0x13 (0x33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
0x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
0x11 (0x31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
0x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
0x0F (0x2F)	SPDR	SPI Data Register								140
0x0E (0x2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	140
0x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	138
0x0C (0x2C)	UDR	USART I/O Data Register								161
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	162
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	163
0x09 (0x29)	UBRRL	USART Baud Rate Register Low Byte								166
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	200
0x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	216
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	218
0x05 (0x25)	ADCH	ADC Data Register High Byte								219
0x04 (0x24)	ADCL	ADC Data Register Low Byte								219
0x03 (0x23)	TWDR	Two-wire Serial Interface Data Register								180
0x02 (0x22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	180
0x01 (0x21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	180

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x00 (0x20)	TWBR	Two-wire Serial Interface Bit Rate Register								178

- Notes:
1. Refer to the USART description for details on how to access UBRRH and UCSRC.
 2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	RdI,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	RdI,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBSI	P, b	Skip if Bit in I/O Register is Set	if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BREQ	k	Branch if Equal	if $(Z = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRNE	k	Branch if Not Equal	if $(Z = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCS	k	Branch if Carry Set	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCC	k	Branch if Carry Cleared	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRSH	k	Branch if Same or Higher	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLO	k	Branch if Lower	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRMI	k	Branch if Minus	if $(N = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRPL	k	Branch if Plus	if $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTS	k	Branch if T Flag Set	if $(T = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if $(T = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if $(V = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if $(V = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if $(I = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if $(I = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
DATA TRANSFER INSTRUCTIONS					

Mnemonics	Operands	Description	Operation	Flags	#Clocks
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
MOVW	Rd, Rr	Copy Register Word	$Rd+1:Rd \leftarrow Rr+1:Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LD	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LD	Rd, Y	Load Indirect	$Rd \leftarrow (Y)$	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	$Rd \leftarrow (Y + q)$	None	2
LD	Rd, Z	Load Indirect	$Rd \leftarrow (Z)$	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (k)$	None	2
ST	X, Rr	Store Indirect	$(X) \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
STD	Y+q, Rr	Store Indirect with Displacement	$(Y + q) \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$(Z) \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
STD	Z+q, Rr	Store Indirect with Displacement	$(Z + q) \leftarrow Rr$	None	2
STS	k, Rr	Store Direct to SRAM	$(k) \leftarrow Rr$	None	2
LPM		Load Program Memory	$R0 \leftarrow (Z)$	None	3
LPM	Rd, Z	Load Program Memory	$Rd \leftarrow (Z)$	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	3
SPM		Store Program Memory	$(Z) \leftarrow R1:R0$	None	-
IN	Rd, P	In Port	$Rd \leftarrow P$	None	1
OUT	P, Rr	Out Port	$P \leftarrow Rr$	None	1
PUSH	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2
POP	Rd	Pop Register from Stack	$Rd \leftarrow STACK$	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P.b	Set Bit in I/O Register	$I/O(P.b) \leftarrow 1$	None	2
CBI	P.b	Clear Bit in I/O Register	$I/O(P.b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	1
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit load from T to Register	$Rd(b) \leftarrow T$	None	1
SEC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1
SES		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLS		Clear Signed Test Flag	$S \leftarrow 0$	S	1
SEV		Set Twos Complement Overflow.	$V \leftarrow 1$	V	1
CLV		Clear Twos Complement Overflow	$V \leftarrow 0$	V	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEH		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/Timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Ordering Information

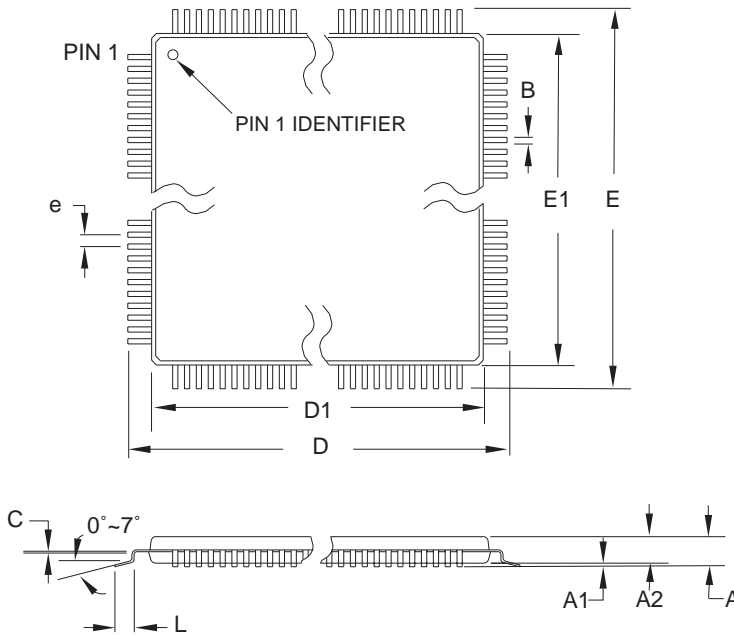
Speed (MHz)	Power Supply	Ordering Code	Package ⁽¹⁾	Operation Range
8	2.7 - 5.5V	ATmega8535L-8AC	44A	Commercial (0°C to 70°C)
		ATmega8535L-8PC	40P6	
		ATmega8535L-8JC	44J	
		ATmega8535L-8MC	44M1	
		ATmega8535L-8AI	44A	Industrial (-40°C to 85°C)
		ATmega8535L-8PI	40P6	
		ATmega8535L-8JI	44J	
		ATmega8535L-8MI	44M1	
16	4.5 - 5.5V	ATmega8535-16AC	44A	Commercial (0°C to 70°C)
		ATmega8535-16PC	40P6	
		ATmega8535-16JC	44J	
		ATmega8535-16MC	44M1	
		ATmega8535-16AI	44A	Industrial (-40°C to 85°C)
		ATmega8535-16PI	40P6	
		ATmega8535-16JI	44J	
		ATmega8535-16MI	44M1	

Note: 1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
44M1-A	44-pad, 7 x 7 x 1.0 mm body, lead pitch 0.50 mm, Micro Lead Frame Package (MLF)

Packaging Information

44A



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	1.20	
A1	0.05	-	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	-	0.45	
C	0.09	-	0.20	
L	0.45	-	0.75	
e	0.80 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
 2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
 3. Lead coplanarity is 0.10 mm maximum.

10/5/2001



2325 Orchard Parkway
San Jose, CA 95131

TITLE

44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness,
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

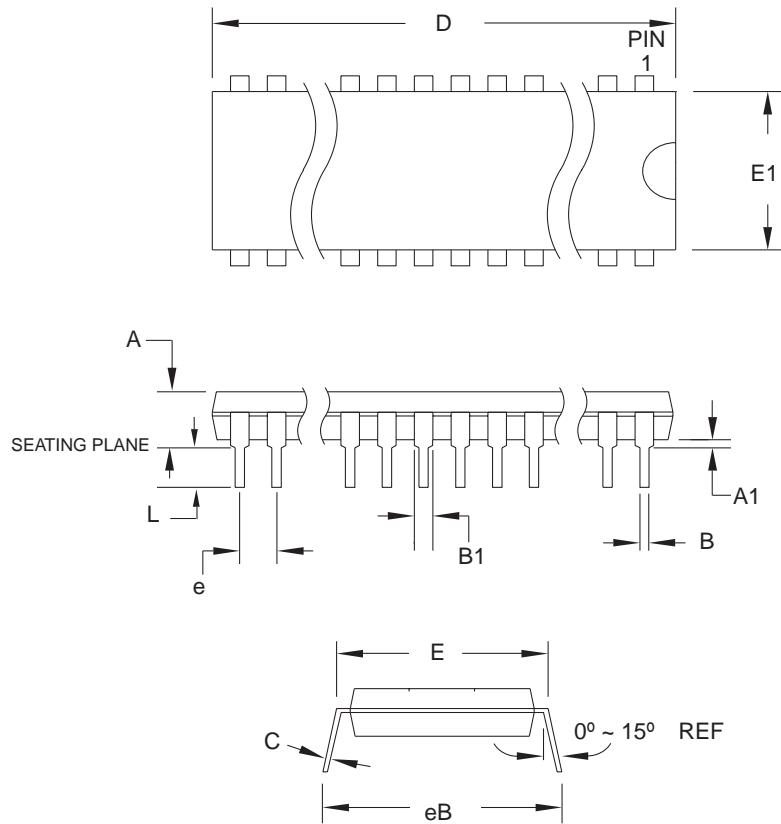
DRAWING NO.

44A

REV.

B

40P6



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.826	
A1	0.381	-	-	
D	52.070	-	52.578	Note 2
E	15.240	-	15.875	
E1	13.462	-	13.970	Note 2
B	0.356	-	0.559	
B1	1.041	-	1.651	
L	3.048	-	3.556	
C	0.203	-	0.381	
eB	15.494	-	17.526	
e	2.540 TYP			

- Notes: 1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01



2325 Orchard Parkway
San Jose, CA 95131

TITLE

40P6, 40-lead (0.600"/15.24 mm Wide) Plastic Dual
Inline Package (PDIP)

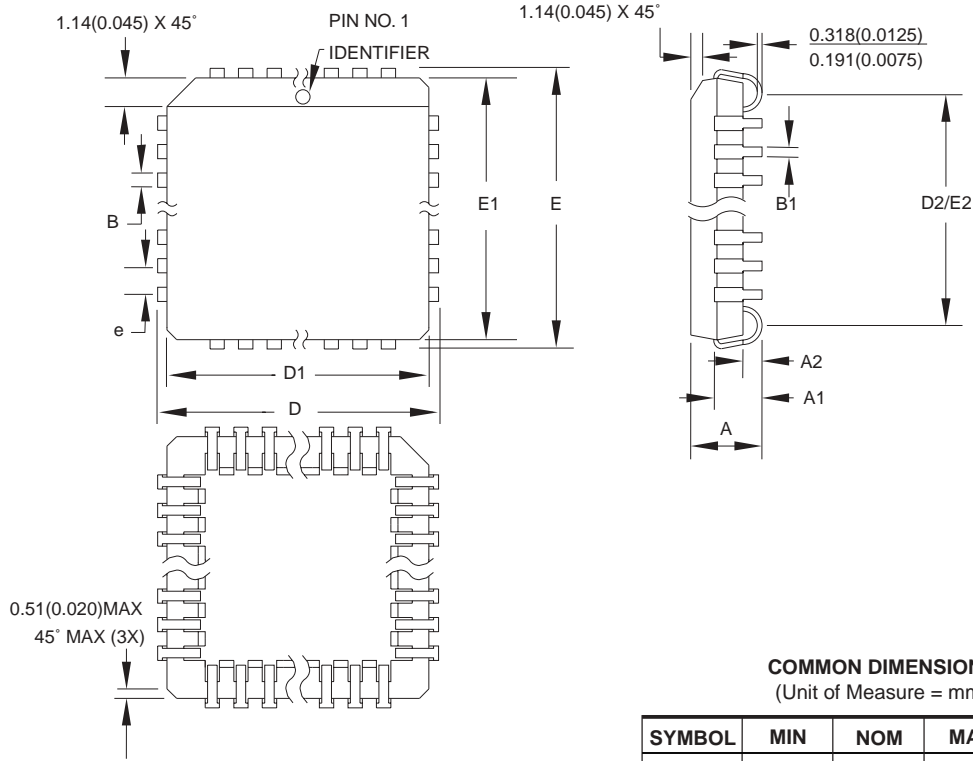
DRAWING NO.

40P6

REV.

B





COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	4.191	-	4.572	
A1	2.286	-	3.048	
A2	0.508	-	-	
D	17.399	-	17.653	
D1	16.510	-	16.662	Note 2
E	17.399	-	17.653	
E1	16.510	-	16.662	Note 2
D2/E2	14.986	-	16.002	
B	0.660	-	0.813	
B1	0.330	-	0.533	
e	1.270 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-018, Variation AC.
 2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010" (0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
 3. Lead coplanarity is 0.004" (0.102 mm) maximum.

10/04/01



2325 Orchard Parkway
San Jose, CA 95131

TITLE

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)

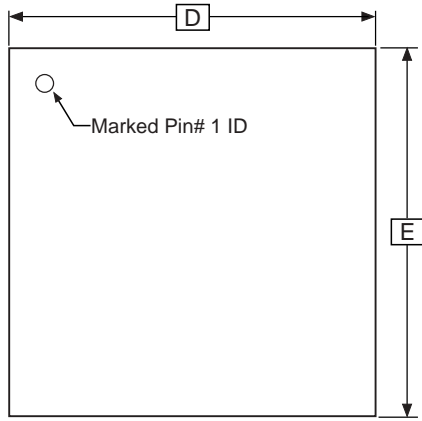
DRAWING NO.

44J

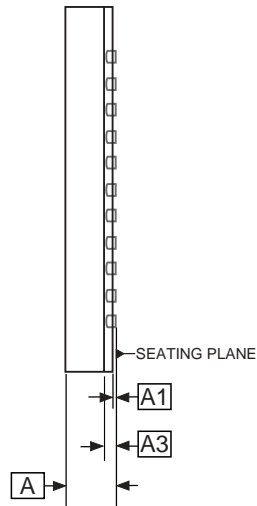
REV.

B

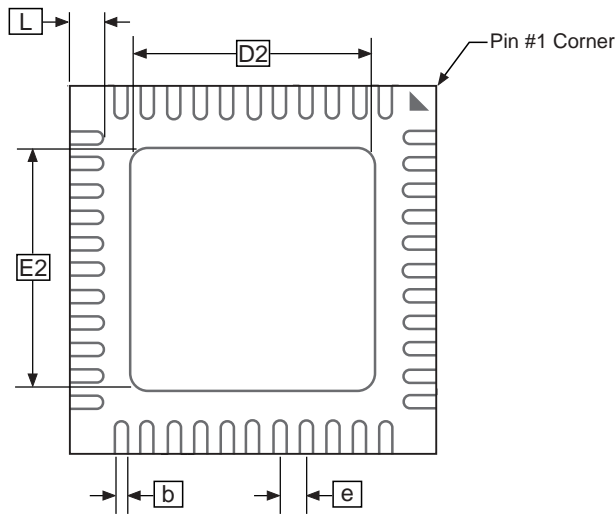
44M1-A



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A3	0.25 REF			
b	0.18	0.23	0.30	
D	7.00 BSC			
D2	5.00	5.20	5.40	
E	7.00 BSC			
E2	5.00	5.20	5.40	
e	0.50 BSC			
L	0.35	0.55	0.75	

Notes: 1. JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-1.

01/15/03



2325 Orchard Parkway
San Jose, CA 95131

TITLE

44M1, 44-pad, 7 x 7 x 1.0 mm Body, Lead Pitch 0.50 mm
Micro Lead Frame Package (MLF)

DRAWING NO.

44M1

REV.

C





Errata

There are no errata for this revision of ATmega8535.

**Datasheet Change
Log for ATmega8535**

Please note that the referring page numbers in this section are referring to this document. The referring revision in this section are referring to the document revision.

**Changes from Rev.
2502D-09/03 to Rev.
2502E-12/03**

1. Updated “Calibrated Internal RC Oscillator” on page 27.
2. Added section “Errata” on page 16.

**Changes from Rev.
2502C-04/03 to Rev.
2502D-09/03**

1. Removed “Advance Information” and some TBD’s from the datasheet.
2. Added note to “Pinout ATmega8535” on page 2.
3. Updated “Reset Characteristics” on page 35.
4. Updated “Absolute Maximum Ratings” and “DC Characteristics” in “Electrical Characteristics” on page 252.
5. Updated Table 111 on page 255.
6. Updated “ADC Characteristics – Preliminary Data” on page 260.
7. Updated “ATmega8535 Typical Characteristics – Preliminary Data” on page 263.
8. Removed CALL and JMP instructions from code examples and “Instruction Set Summary” on page 8.

**Changes from Rev.
2502B-09/02 to Rev.
2502C-04/03**

1. Updated “Packaging Information” on page 12.
2. Updated Figure 1 on page 2, Figure 84 on page 176, Figure 85 on page 182, Figure 87 on page 188, Figure 98 on page 204.
3. Added the section “EEPROM Write During Power-down Sleep Mode” on page 20.
4. Removed the references to the application notes “Multi-purpose Oscillator” and “32 kHz Crystal Oscillator”, which do not exist.
5. Updated code examples on page 42.
6. Removed ADHSM bit.
7. Renamed Port D pin ICP to ICP1. See “Alternate Functions of Port D” on page 62.
8. Added information about PWM symmetry for Timer 0 on page 77 and Timer 2 on page 124.
9. Updated Table 68 on page 166, Table 75 on page 187, Table 76 on page 190, Table 77 on page 193, Table 108 on page 250, Table 113 on page 258.
10. Updated description on “Bit 5 – TWSTA: TWI START Condition Bit” on page 179.

11. Updated the description in “Filling the Temporary Buffer (Page Loading)” and “Performing a Page Write” on page 228.
12. Removed the section description in “SPI Serial Programming Characteristics” on page 251.
13. Updated “Electrical Characteristics” on page 252.
14. Updated “ADC Characteristics – Preliminary Data” on page 260.
14. Updated “Register Summary” on page 6.
15. Various Timer 1 corrections.
16. Added WD_FUSE period in Table 108 on page 250.

**Changes from Rev.
2502A-06/02 to Rev.
2502B-09/02**

1. Canged the Endurance on the Flash to 10,000 Write/Erase Cycles.



Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenalux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests

www.atmel.com/literature

Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2003. All rights reserved. Atmel® and combinations thereof, AVR®, and AVR Studio® are the registered trademarks of Atmel Corporation or its subsidiaries. Microsoft®, Windows®, Windows NT®, and Windows XP® are the registered trademarks of Microsoft Corporation. Other terms and product names may be the trademarks of others



Printed on recycled paper.

2502ES-AVR-12/03

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.