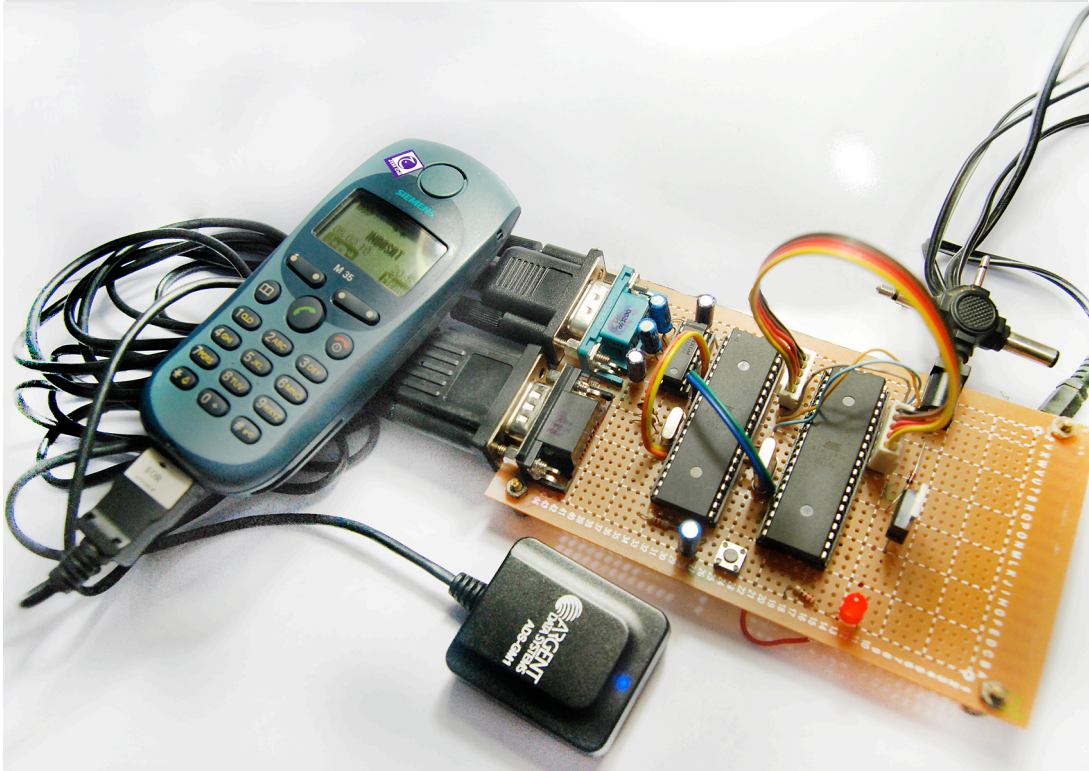
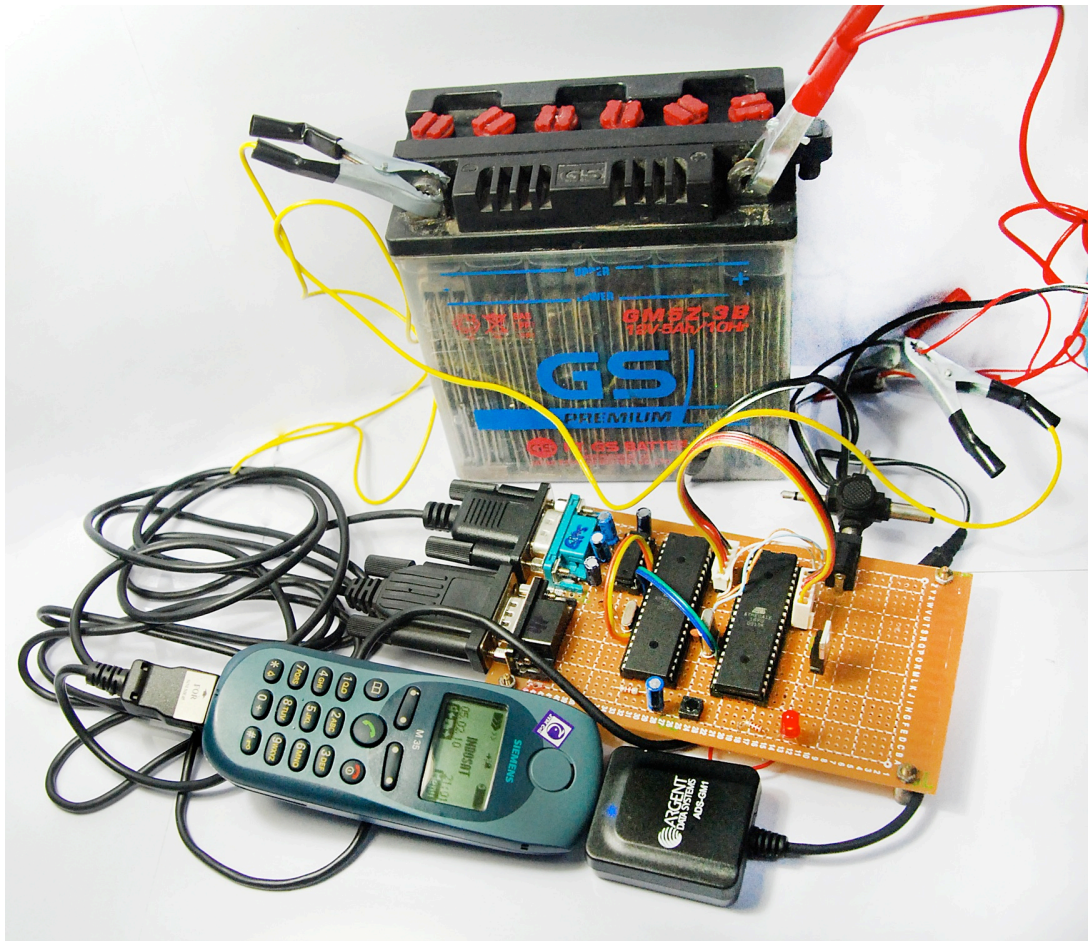


LAMPIRAN A
GAMBAR SISTEM



LAMPIRAN B
LISTING PROGRAM AVR ATMEGA 16

```
/******  
Chip type           : ATmega16 - 1  
Program type       : Application  
Clock frequency    : 11.059200 MHz  
Memory model       : Small  
External SRAM size : 0  
Data Stack size    : 256  
******/
```

```
#include <mega16.h>  
#include <stdio.h>  
#include <delay.h>
```

```
unsigned char x;
```

```
void main(void)  
{
```

```
PORTA=0xFF;  
DDRA=0xFF;
```

```
PORTB=0x00;  
DDRB=0x00;
```

```
PORTC=0x00;  
DDRC=0x00;
```

```
PORTD=0x00;  
DDRD=0x00;
```

```
TCCR0=0x00;  
TCNT0=0x00;  
OCR0=0x00;
```

```
TCCR1A=0x00;  
TCCR1B=0x00;  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;
```

```
ASSR=0x00;  
TCCR2=0x00;  
TCNT2=0x00;
```

```
OCR2=0x00;

MCUCR=0x00;
MCUCSR=0x00;

TIMSK=0x00;

UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x8F;

ACSR=0x80;
SFIOR=0x00;

x=0;

while (1)
{
    x=getchar();
    PORTA=x;
    delay_us(10);
    PORTA=0xFF;
};
}
```

```

/*****
Chip type           : ATmega16 - 2
Program type       : Application
Clock frequency    : 11.059200 MHz
Memory model       : Small
External SRAM size : 0
Data Stack size    : 256
*****/

#include <mega16.h>
#include <stdio.h>
#include <delay.h>

#define enter putchar(0x0D);

unsigned char in,out,x,index=0,n,ok=0,sip,panjang;
unsigned char data[80];

void chartohex(unsigned char input) // fungsi untuk merubah bilangan
ke bentuk ASCII hex
{
    unsigned char digit1,digit2,t;
    t=input/16;
    if (t<10) {digit1=t+48;} else digit1=t+55;
    t=input-(t*16);
    if (t<10) {digit2=t+48;} else digit2=t+55;
    putchar(digit1);
    delay_ms(5);
    putchar(digit2);
    delay_ms(5);
}

void encode( unsigned char *c) // fungsi untuk merubah string ke bentuk
PDU dan mengirimnya ke port serial
{
    unsigned char p=1,d,e,f,g=1;
    e=c[0];
    while (c[p]!='\0')
    {
        f=c[p]; // 'l' = 1101100
        d=f & ( 0b11111111>>(8-g)); // d=0
        e=d<<(8-g); // e= 11000011
        f=f>>g; //
        if (g>7) {g=1;e=c[p];}
        else {g++; chartohex(e); e=f; }
        p++;
    }
    if(g!=8){chartohex(e);}
}

```

```

}

void tulis(unsigned char flash *c)
{
    unsigned char indeks=0;
    while (c[indeks]!='\0')
    {
        putchar (c[indeks]);
        indeks++;
        delay_ms(5);
    }
}

unsigned char baca ( unsigned char flash *c)
{
    unsigned char a=1,indeks=0;
    while ((a!=0) && (c[indeks]!='\0'))
    {
        if (getchar()!= c[indeks]) a&=0;
        indeks++;
    }
    return a ;
}

void skip(unsigned char kali)
{
    unsigned char j;
    for(j=0;j<kali;j++) {getchar();}
}

void cek_port()
{
    sip=0;
    while (sip!=1)
    {
        if (PINA.0==0)
        {
            out=0;
            in=PINA;
            for (x=0;x<4;x++) out|= (in & (0b10000000 >> x) ) >> (7-
2*x);
            for (x=4;x<8;x++) out|= (in & (0b10000000 >> x) ) << (2*x-
7);
            if (out=='$') index=1;
            if (index==6) if (out=='C') ok=1;
            if (index>0) {data[index-1]=out; index++;}
            if ((out==0x0A) && (index>1) && (ok==1))

```

```

        {
            data[index-1]=0;
            index=0;
            //puts(data);
            sip=1;
            ok=0;
        }
        while (PINA.0==0);
    }
}

```

```

void main(void)
{

```

```

    PORTA=0xFF;
    DDRA=0x00;

```

```

    PORTB=0x00;
    DDRB=0x00;

```

```

    PORTC=0x00;
    DDRC=0x00;

```

```

    PORTD=0x00;
    DDRD=0x00;

```

```

    TCCR0=0x00;
    TCNT0=0x00;
    OCR0=0x00;

```

```

    TCCR1A=0x00;
    TCCR1B=0x00;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x00;
    ICR1L=0x00;
    OCR1AH=0x00;
    OCR1AL=0x00;
    OCR1BH=0x00;
    OCR1BL=0x00;

```

```

    ASSR=0x00;
    TCCR2=0x00;
    TCNT2=0x00;
    OCR2=0x00;

```



```

MCUCR=0x00;
MCUCSR=0x00;

TIMSK=0x00;

UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x23;

ACSR=0x80;
SFIOR=0x00;

ok=0;

delay_ms(5000);

tulis("ATE0");
enter;
while (!baca("OK"));
skip(2);
delay_ms(20);
tulis("AT+CLIP=1");
enter;
skip(6);
delay_ms(20);

while(1)
{
x=getchar();
if (x=='R')
{
skip(15);
if (baca("6285722545587"))
{
delay_ms(1000);
tulis("ATH");
enter;
while(getchar()!='K');
while(getchar()!='\n');
delay_ms(1000);
cek_port();
n=0;
data[0]=0x02;
while(data[n]!=0){n++;}
n--;
}
}
}

```

```

data[n]=0;
n--;
tulis("AT+CMGS=");
panjang=n;
n=n-n/8;
n=n+14;
if(n>9) putchar(n/10+0x30);
putchar(n%10+0x30);
enter;
while(getchar()!='>');
delay_ms(1000);
tulis("0001000D9126");
tulis("5827525485F7"); // no. hp penerima
tulis("0000");
chartohex(panjang);
delay_ms(5);
encode(data);
delay_ms(1000);
putchar(0x1A);
}
}
}
}

```

LAMPIRAN C
CODE VISUAL BASIC

Form 1

```
'untuk login memerlukan password
Dim login As Integer
Dim kesempatan As Integer
Private Sub Command1_Click()
    kesempatan = kesempatan + 1
    If (LCase(Text1.Text) = "login") And (kesempatan <= 3) Then
        Image1.Visible = True
        Text1.Enabled = False
        Command1.Enabled = False
    Else
        MsgBox "Passwordnya Salah !"
        If kesempatan < 3 Then
            Text1.Text = ""
            Text1.SetFocus
        Else
            End
        End If
    End If
End Sub
Private Sub Image1_Click()
Form2.Show
End Sub
Private Sub Lbl_Keluar_Click()
End
End Sub
```

Form 2

```
Form 2
Dim doit As Boolean
Dim buffer As String
'deklarasi untuk SMS
Dim objGsmProtocol As AXmsCtrl.SmsProtocolGsm
Dim objSmsMessage As AXmsCtrl.SmsMessage
Dim objSmsConstants As AXmsCtrl.SmsConstants
Dim ShowReference As Boolean
Dim mOk
Dim mErr
Sub delay()
    For x = 1 To 100
        For y = 1 To 1000
            For z = 1 To 3000
                Next z
            Next y
        Next x
    End Sub
```

```
Next x
End Sub
```

```
Private Sub fresh_Click1()
    List1.AddItem "connected..."
End Sub
```

```
Private Sub chkDebug_Click()
    Me.picDebug.Visible = Me.chkDebug.Value
End Sub
```

```
Private Sub cmdConnect_Click()
    On Error GoTo p1
    If Me.cmdConnect.Caption = "&Connect" Then
        If Len(Me.cmbPorts.Text) = 0 Then MsgBox "TOLONG dicek lagi ya portnya...":
Me.cmbPorts.SetFocus: Exit Sub
        Me.cmdConnect.Caption = "&Disconnect"
        setStatus "Sudah tersambung..."
        MSComm1.RThreshold = 1
        MSComm1.InputLen = 0
        MSComm1.Settings = "19200,N,8,1"
        MSComm1.DTREnable = True
        MSComm1.InBufferSize = 32
        MSComm1.OutBufferSize = 0
        MSComm1.CommPort = Me.cmbPorts.Text
        MSComm1.RTSEnable = True
        DoEvents
        MSComm1.PortOpen = True
        DoEvents
        setStatus "Sudah Tersambung ke Port: " & Me.cmbPorts.Text
        DoEvents
        Me.picInfo1.Enabled = True
        setStatus "Getting Status nich...."
        getMobileInfo
        setStatus "Berhasil...Tersambung Ke COM : " & Me.cmbPorts.Text
    ElseIf Me.cmdConnect.Caption = "&Disconnect" Then
        Me.cmdConnect.Caption = "&Connect"
        setStatus "Diconnected dari port : " & Me.cmbPorts.Text
        Me.txtOut = ""
        Me.MSComm1.PortOpen = False
        Me.picInfo1.Enabled = False
    End If
    Exit Sub
p1:
    MsgBox "MAAF, TIDAK BISA CONNECT NICH HPNYA, Tolong di cek lagi ya
Portnya...Coba lagi aja.. OK :D" & vbCrLf & Err.Description
End
```

```

End Sub
Function getProvider(ByVal s)
    s1 = ""
    If Len(s) > 0 Then
        p = InStr(s, Chr(34))
        s1 = Mid(s, p + 1)
        p1 = InStr(s1, Chr(34))
        If p1 > 0 Then
            s1 = Mid(s1, 1, p1 - 1)
        End If
    End If
    getProvider = s1
End Function
Function getManufacturer(ByVal s)
    s1 = ""
    If Len(s) > 0 Then
        s1 = Mid(s, 11)
        p = InStr(s1, Chr(13))
        If p = 0 Then p = InStr(s1, Chr(10))
        If p > 0 Then
            s1 = Mid(s1, 1, p - 1)
        End If
    End If
    getManufacturer = s1
End Function
Function getDevType(ByVal s)
    s1 = ""
    If Len(s) > 0 Then
        s1 = Mid(s, 7)
        p = InStr(s1, Chr(10))
        If p = 0 Then p = InStr(s1, Chr(13))
        If p > 0 Then
            s1 = Mid(s1, 1, p)
        End If
    End If
    getDevType = s1
End Function
Private Sub cmdRefresh_Click()
    setStatus "Memulai Ports list Nichh..."
    ListComPorts
    'setStatus ""
End Sub
Sub getMobileInfo()
    Dim st As Boolean
    Me.txtOut = ""
    st = sendIt("AT", "OK", "ERROR")

```

```

If st = True Then
    'Everything OKe
Else
    'Not tersambung
    MsgBox "GSM Modem Not found"
    End
End If
Me.txtOut = ""
st = sendIt("ATI", "OK", "ERROR")
If st = True Then
    Me.lblDevType.Caption = getDevType(Me.txtOut)
Else
    Me.lblDevType.Caption = ""
End If
Me.txtOut = ""
st = sendIt("AT+CGMI", "OK", "ERROR")
If st = True Then
    Me.lblManufacturer.Caption = getManufacturer(Me.txtOut)
Else
    Me.lblManufacturer.Caption = ""
End If
Me.txtOut = ""
st = sendIt("AT+COPS?", "OK", "ERROR")
If st = True Then
    Me.lblProvider.Caption = getProvider(Me.txtOut)
Else
    Me.lblProvider.Caption = ""
End If
'Me.txtOut = ""
End Sub
Private Sub ListComPorts()
    Dim i As Integer
    Me.cmbPorts.Clear
    setStatus "Getting Available Com Portsnya yaa..."
    For i = 1 To 16
        If COMAvailable(i) Then
            Me.cmbPorts.AddItem i
            setStatus "Com " & i & " found"
        End If
    Next
    Me.cmbPorts.ListIndex = 0
End Sub
Sub setStatus(ByVal s)
    Me.lblStatus.Caption = "" & s
End Sub
Function sendIt(ByVal s, ByVal ok, ByVal eror, Optional ByVal TOut = 2) As Boolean

```

```

mOK = ok
mErr = eror
Me.List1.AddItem "sending..." & s
MSComm1.Output = s & Chr(13)
Dim p As Double, p1 As Double
p = 0.0001 * TOut
doit = False
sdata = ""
Dim dt1 As Date, dt2 As Date
dt1 = Now
s1 = ""
While doit = False
    dt2 = Now
    p1 = (dt2 - dt1)
    If p1 >= p Then
        doit = True
        sendIt = False
        Exit Function
    End If
    DoEvents
Wend
sendIt = True
End Function
'Perintah untuk misscall ke bagian objek
Private Sub Command1_Click(Index As Integer)
Me.txtOut = ""
st = sendIt("ATD085721615485;", "OK", "ERROR")
Call delay
Me.txtOut = ""
st = sendIt("ATH", "OK", "ERROR")
End Sub

'Perintah untuk keluar dari aplikasi
Private Sub Command3_Click()
End
End Sub

'Prosedur untuk mengaktifkan mapinfo
Private Sub Command4_Click()
Shell "C:\Program Files\MapInfo\Professional\MAPINFOW.exe", vbNormalFocus
Me.SetFocus
End Sub

'Perintah untuk menampilkan form help? pada form3
Private Sub Command5_Click()
Form3.Show
End Sub

Public Function FileExists(sFileName As String) As Boolean

```



```

    FileExists = CBool(Len(Dir$(sFileName))) And CBool(Len(sFileName))
End Function
'Perintah untuk melihat logfile
Private Sub Logfile_Click()
    If FileExists(TextLogfile.Text) = True Then
        Shell "notepad " + TextLogfile.Text, vbNormalFocus
    End If
End Sub
Private Sub Form_Load()
    Dim numDevices
    Dim i
    Set objGsmProtocol = CreateObject("ActiveXperts.SmsProtocolGsm")
    Set objSmsMessage = CreateObject("ActiveXperts.SmsMessage")
    Set objSmsConstants = CreateObject("ActiveXperts.SmsConstants")
    Set objMm1Protocol = CreateObject("ActiveXperts.MmsProtocolMm1")
    numDevices = objMm1Protocol.GetDeviceCount
    For i = 0 To numDevices - 1
        ComboDevice.AddItem (objMm1Protocol.GetDevice(i))
        ComboDevice.ListIndex = 0
    Next
    SetDefaultLogFile
End Sub
' Untuk menunjukkan keterangan $GPRMC header
Private Sub Text1_Change()
    Dim a As String
    Dim b As String
    Dim c As String
    Dim d As String
    Dim e As String
    a = Mid(Text1.Text, 8, 6) ' indikasi UTM time
    Text2.Text = a
    b = Mid(Text1.Text, 19, 1) ' indikasi data valid atau tidak valid
    Text3.Text = b
    c = Mid(Text1.Text, 21, 9) ' indikasi latitude
    Text4.Text = c
    d = Mid(Text1.Text, 31, 1) ' indikasi N/S
    Text5.Text = d
    d = Mid(Text1.Text, 33, 10) ' indikasi lognitude
    Text6.Text = d
    e = Mid(Text1.Text, 44, 1) 'indikasi E/W
    Text7.Text = e
'Prosedur untuk menyimpan data.gps
    k = Text1.Text
    buffer = buffer & k
    filedata = "data.gps"
    Open filedata For Output As #1

```

```

Print #1, buffer
Close #1
'buffer = ""
' Set LogFile
objGsmProtocol.Logfile = TextLogfile.Text
End Sub
Private Sub MSComm1_OnComm()
Dim sdata1
Me.List1.AddItem "In OnComm"
If MSComm1.CommEvent = comEvReceive Then
sdata1 = MSComm1.Input
sdata = sdata & sdata1
If InStr(sdata, mOK) > 0 Then
doit = True
ElseIf InStr(sdata, mErr) > 0 Then
doit = True
ElseIf InStr(sdata, ">") > 0 Then
doit = True
End If
mResult = sdata
If Len(sdata) > 0 Then Me.List1.AddItem "--> " & sdata
txtOut = txtOut & sdata
End If
End Sub
'Perintah untuk memunculkan inbox di text2 dengan waktu 10 detik
Private Sub Timer1_Timer()
Dim NumMessages As Long
Dim i As Long
Set objGsmProtocol = CreateObject("ActiveXperts.SmsProtocolGsm")
Set objSmsMessage = CreateObject("ActiveXperts.SmsMessage")
Set objSmsConstants = CreateObject("ActiveXperts.SmsConstants")
Screen.MousePointer = vbHourglass
Text1.Text = ""
objGsmProtocol.Device = ComboDevice.Text 'Set Device
objGsmProtocol.MessageStorage = objSmsConstants.asSTORAGE_DEVICE 'ambil
dari inbox HP
NumMessages = objGsmProtocol.Receive 'Retrieve messages
If GetResult = 0 Then ' Success?
For i = 0 To NumMessages - 1
On Error Resume Next
Set objSmsMessage = objGsmProtocol.GetMessage(NumMessages - 1)
On Error GoTo 0
If GetResult = 0 Then
Text1.Text = objSmsMessage.Data ' Add data to list control
End If
Next

```

```

    End If
    'objGsmProtocol.DeleteMessage (0)
    Screen.MousePointer = vbDefault
End Sub
Private Function SetDefaultLogFile()
    TextLogfile.Text = "C:\sms.txt"
End Function

```

MODULES

```

Option Explicit
'// API calls
Private Declare Function EnumPorts Lib "winspool.drv" Alias "EnumPortsA" (ByVal
pName As String, ByVal Level As Long, ByVal lpbPorts As Long, ByVal cbBuf As
Long, pcbNeeded As Long, pcReturned As Long) As Long
Private Declare Function lstrlenW Lib "kernel32" (ByVal lpString As Long) As Long
Private Declare Sub CopyMem Lib "kernel32" Alias "RtlMoveMemory" (pTo As Any,
uFrom As Any, ByVal lSize As Long)
Private Declare Function HeapAlloc Lib "kernel32" (ByVal hHeap As Long, ByVal
dwFlags As Long, ByVal dwBytes As Long) As Long
Private Declare Function GetProcessHeap Lib "kernel32" () As Long
Private Declare Function HeapFree Lib "kernel32" (ByVal hHeap As Long, ByVal
dwFlags As Long, lpMem As Any) As Long
'// Public Data Structure - up to 100 Ports Information
Public Ports(0 To 100) As PORT_INFO_2
'// API Structures
Private Type PORT_INFO_2
    pPortName As String
    pMonitorName As String
    pDescription As String
    fPortType As Long
    Reserved As Long
End Type

Private Type API_PORT_INFO_2
    pPortName As Long
    pMonitorName As Long
    pDescription As Long
    fPortType As Long
    Reserved As Long
End Type

'// These declarations are used to detect what Com ports
'// are available
'// API Declarations
Public Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal
lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As

```

```

Long, lpSecurityAttributes As SECURITY_ATTRIBUTES, ByVal
dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal
hTemplateFile As Long) As Long
Public Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long

```

```

'// API Structures

```

```

Public Type SECURITY_ATTRIBUTES
    nLength As Long
    lpSecurityDescriptor As Long
    bInheritHandle As Long
End Type

```

```

'// API constants

```

```

Public Const FILE_SHARE_READ = &H1
Public Const FILE_SHARE_WRITE = &H2
Public Const OPEN_EXISTING = 3
Public Const FILE_ATTRIBUTE_NORMAL = &H80

```

```

'// This detects if a COM ports is available.

```

```

'// Used by the ListComPorts() procedure

```

```

'// Returns TRUE if the COM exists, FALSE if the COM does not exist

```

```

Public Function COMAvailable(COMNum As Integer) As Boolean

```

```

    Dim hCOM As Long

```

```

    Dim ret As Long

```

```

    Dim sec As SECURITY_ATTRIBUTES

```

```

    'try to open the COM port

```

```

    hCOM = CreateFile("COM" & COMNum & "", 0, FILE_SHARE_READ +
FILE_SHARE_WRITE, sec, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0)

```

```

    If hCOM = -1 Then

```

```

        COMAvailable = False

```

```

    Else

```

```

        COMAvailable = True

```

```

        'close the COM port

```

```

        ret = CloseHandle(hCOM)

```

```

    End If

```

```

End Function

```

```

'// This procedure returns all the available ports

```

```

'// Used by the cmdGetAllPorts_Click() procedure

```

```

'// Use ServerName to specify the name of a Remote Workstation i.e. "//WIN95WKST"

```

```

'// or leave it blank "" to get the ports of the local Machine

```

```

Public Function GetAvailablePorts(ServerName As String) As Long

```

```

    Dim ret As Long

```

```

    Dim PortsStruct(0 To 100) As API_PORT_INFO_2

```

```

    Dim pcbNeeded As Long

```

```

Dim pcReturned As Long
Dim TempBuff As Long
Dim i As Integer

// Get the amount of bytes needed to contain the data returned by the API call
ret = EnumPorts(ServerName, 2, TempBuff, 0, pcbNeeded, pcReturned)
// Allocate the Buffer
TempBuff = HeapAlloc(GetProcessHeap(), 0, pcbNeeded)
ret = EnumPorts(ServerName, 2, TempBuff, pcbNeeded, pcbNeeded, pcReturned)
If ret Then
    // Convert the returned String Pointer Values to VB String Type
    CopyMem PortsStruct(0), ByVal TempBuff, pcbNeeded
    For i = 0 To pcReturned - 1
        Ports(i).pDescription = LPSTRtoSTRING(PortsStruct(i).pDescription)
        Ports(i).pPortName = LPSTRtoSTRING(PortsStruct(i).pPortName)
        Ports(i).pMonitorName = LPSTRtoSTRING(PortsStruct(i).pMonitorName)
        Ports(i).fPortType = PortsStruct(i).fPortType
    Next
End If
GetAvailablePorts = pcReturned
// Free the Heap Space allocated for the Buffer
If TempBuff Then HeapFree GetProcessHeap(), 0, TempBuff
End Function

Public Function TrimStr(strName As String) As String
    // Finds a null then trims the string
    Dim x As Integer

    x = InStr(strName, vbNullChar)
    If x > 0 Then TrimStr = Left(strName, x - 1) Else TrimStr = strName
End Function

Public Function LPSTRtoSTRING(ByVal lngPointer As Long) As String
    Dim lngLength As Long

    // Get number of characters in string
    lngLength = lstrlenW(lngPointer) * 2
    // Initialize string so we have something to copy the string into
    LPSTRtoSTRING = String(lngLength, 0)
    // Copy the string
    CopyMem ByVal StrPtr(LPSTRtoSTRING), ByVal lngPointer, lngLength
    // Convert to Unicode
    LPSTRtoSTRING = TrimStr(StrConv(LPSTRtoSTRING, vbUnicode))
End Function

```

LAMPIRAN D
DATA SHEET

Penjelasan GPS NMEA 0813

NMEA-0183 adalah standar kalimat laporan yang dikeluarkan oleh GPS receiver. Standar NMEA memiliki banyak jenis bentuk kalimat laporan, di antaranya yang paling penting adalah koordinat lintang (latitude), bujur (longitude), ketinggian (altitude), waktu sekarang standar UTC (UTC time), dan kecepatan (speed over ground).

Output GPS : \$GPRMC

Contoh: \$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,*,*25

Field	Contoh isi	Deskripsi
Sentence ID	\$GPRMC	
UTC Time	092204.999	hhmmss.sss
Status	A	A = Valid, V = Invalid
Latitude	4250.5589	ddmm.mmmm
N/S Indicator	S	N = North, S = South
Longitude	14718.5084	dddmm.mmmm
E/W Indicator	E	E = East, W = West
Speed over ground	0.00	Knots
Course over ground	0.00	Degrees
UTC Date	211200	DDMMYY
Magnetic variation		Degrees
Checksum	*25	
Terminator	CR/LF	

Jenis kalimat NMEA-0183 adalah sebagai berikut.

Kalimat	Deskripsi
\$GPGGA	Global positioning system fixed data
\$GPGLL	Geographic position - latitude / longitude
\$GPGSA	GNSS DOP and active satellites
\$GPGSV	GNSS satellites in view
\$GPRMC	Recommended minimum specific GNSS data
\$GPVTG	Course over ground and ground speed

\$GPGGA

Contoh: \$GPGGA,092204.999,4250.5589,S,14718.5084,E,1,04,24.4,19.7,M,,0000*1F

Field	Contoh isi	Deskripsi
Sentence ID	\$GPGGA	
UTC Time	092204.999	hhmmss.sss
Latitude	4250.5589	ddmm.mmmm
N/S Indicator	S	N = North, S = South
Longitude	14718.5084	dddmm.mmmm
E/W Indicator	E	E = East, W = West
Position Fix	1	0 = Invalid, 1 = Valid SPS, 2 = Valid DGPS, 3 = Valid PPS
Satellites Used	04	Satellites being used (0-12)
HDOP	24.4	Horizontal dilution of precision
Altitude	19.7	Altitude in meters according to WGS-84 ellipsoid
Altitude Units	M	M = Meters
Geoid Separation		Geoid separation in meters according to WGS-84 ellipsoid
Separation Units		M = Meters
DGPS Age		Age of DGPS data in seconds
DGPS Station ID	0000	
Checksum	*1F	
Terminator	CR/LF	

PERINTAH – PERINTAH AT (AT Commands)

Perintah	Fungsi
AT+CACM	Akumulasi panggilan meter
AT+CALM	Peringatan tipe suara
AT+CAMM	Maksimum panggilan meter terakumulasi
AT+CAOC	Jumlah yang perlu diisi
AT+CBC	Pengisi baterai
AT+CBST	Pemilihan pembawa tipe service
AT+CCFC	Panggilan ke depan
AT+CCLK	Jam
AT+CCWA	Panggilan ditunggu
AT+CEER	Query alasan untuk memutuskan hubungan panggilan terakhir
AT+CGACT	PDP pengisi aktif atau nonaktif
AT+CGANS	Respon manual ke sebuah permintaan jaringan untuk PDP mengaktifasi konteks
AT+CGATT	GPRS attach atau detach
AT+CGAUTO	Respon otomatis ke sebuah permintaan jaringan untuk PDP mengaktifasi konteks
AT+CGCLASS	GPRS mobile station class
AT+CGDATA	Tempat pemasukan data
AT+CGDCONT	Penentuan isi PDP
AT+CGEREP	Laporan kejadian GPRS
AT+CGMI	Isu manufaktur kode ID
AT+CGMM	Isu model kode ID
AT+CGMR	Keluar GSM versi telepon
AT+CGPADDR	Petunjuk alamat PDP
AT+CGQMIN	Kualitas dari service profile (permintaan minimum)
AT+CGQREQ	Kualitas dari service profile (diminta)
AT+CGREQ	Jaringan pendaftaran status GPRS
AT+CGSMS	Pemilihan service untuk pesan MO SMS
AT+CGSN	Pengeluaran nomor serial (IMEI)
AT+CHLD	Panggilan di-hold dan multiparty
AT+CHUP	Panggilan terminal
AT+CIMI	Keluaran dari IMSI
AT+CKPD	Keypad kontrol
AT+CLCC	Daftar panggilan sekarang
AT+CLCK	Switch pengunci on dan off
AT+CLIP	Menampilkan nomor telepon pemanggil
AT+CLIR	Memilih modus incognito (pembatas panggilan line identifikasi)
AT+CLVL	Level volume pengeras suara
AT+CMEE	Perluasan pesan error menurut GSM 07.07

AT+CMUT	Mute Kontrol
AT+COLP	Terkoneksi baris lampiran identitas
AT+COPN	Membaca nama operator
AT+COPS	Perintah mengenai pilihan operator jaringan
AT+CPAS	Query status telepon
AT+CPBR	Membaca sebuah pemasukan buku telepon
AT+CPIN	Memasukan PIN dan kunci query
AT+CPOL	Mengajukan daftar operator
AT+CPUC	Harga perunit dan tabel mata uang
AT+CPWD	Mengubah password ke sebuah kunci
AT+CR	Kontrol servis pelaporan
AT+CRC	Hasil kode-kode seluler
AT+CREG	Pendaftaran jaringan
AT+CRLP	Memilih radio link protocol parameter untuk keaslian nontransparansi data panggilan
AT+CRSL	Level suara kring (ringer)
AT+CRSM	Keterbatasan akses SIM
AT+CSCS	Memilih karakter TE
AT+CSQ	Kualitas sinyal keluaran
AT+CSSN	Supplementary service notifications
AT+CVIB	Modus getaran
AT+GSN	Pengeluaran nomor serial (IMEI)
AT+VTS	Penerimaan sebuah bunyi DTMF
AT+VTD	Menentukan lamanya DTMF tone
AT+WS46	Memilih jaringan wirelesss
AT+CMGC	Mengirim sebuah perintah SMS
AT+CMGD	Menghapus sebuah SMS dalam SMS memori
AT+CMGF	SMS format
AT+CMGL	Daftar SMS
AT+CMGR	Membaca dalam sebuah SMS
AT+CMGS	Mengirim sebuah SMS
AT+CMGW	Mengisi sebuah SMS ke SMS memori
AT+CMSS	Mengirim sebuah SMS dari SMS memori
AT+CNMA	Pemberitahuan pasan singkat langsung pengeluarannya
AT+CNMI	Menampilkan datangnya SMS baru
AT+CPMS	Preferred SMS message storage
AT+CSCA	Alamat dari pusat SMS service
AT+CSCB	Memilih sel pesan-pesan broadcast
AT+CSMS	Pemilihan dari service pesan
AT+GCAP	Permintaan daftar kemampuan
AT+IPR	Terpasang DTE rate
AT+FBADLIN	Menentukan atau membaca nomor line-line yang rusak (bad line)
AT+FBADMUL	Menentukan, membaca, atau test nomor dari line-line yang rusak

AT+FBOR	Query urutan bit untuk modus penerimaan
AT+FCIG	Query atau set pemilihan lokal id
AT+FCLASS	Memilih baca atau tes fax servis class
AT+FCQ	Kontrol kualitas duplikasi
AT+FCR	Kemampuan untuk menerima
AT+FDCC	Memilih services untuk pesan MOSMS
AT+FDFFC	Kompresi data penukaran format
AT+FDIS	Query atau set session parameter
AT+FDR	Memulai atau melanjutkan fase C penerimaan data
AT+FDT	Transmisi data
AT+FET	Akhir dari halaman atau dokumen
AT+FK	Menghentikan operasi seperti membatalkan fax
AT+FLID	Query or set session parameters
AT+FMDL	Pengenalan model produk
AT+FMFR	Meminta identifikasi manufaktur
AT+FOPT	Set bit order secara independen
AT+FPHCTO	DTE Phase C respons waktu-habis
AT+FREV	Mengidentifikasi produk revisi
AT+FRH	Penerimaan data menggunakan bingkai HDLC
AT+FRM	Penerimaan data
AT+FRS	Penerimaan sepi
AT+FTH	Pemindahan data menggunakan bingkai HDLC
AT+FTM	Pemindahan data
AT+FTS	Penghentian transmisi dan tunggu
AT+FVRFC	Vertical resolution format conversion
AT+SACM	Keluaran ACM (terakumulasi penggilan meter) dan ACMmax
AT+SBNR	Pembacaan binary
AT+SBNW	Pengisian binary
AT+SCID	Keluaran kartu ID
AT+SCKS	Keluaran kartu SIM status
AT+SCNI	Keluaran nomor panggilan informasi
AT+SDBR	pembacaan database
AT+SDLD	Menghapus dial nomor terakhir di memori
AT+SGAUTH	Select Type of Authent cation for PPP connection
AT+SICO	Kontrol gambar icon
AT+SLCK	Switch locks (including user define locks) on dan off
AT+SLNG	Setting bahasa
AT+SMGO	SMS indikasi overflow
AT+SMGL	Daftar SMS (tanpa perubahan status dari yang tak membaca ke yang membaca)
AT+SMGR	Membaca SMS (tanpa status berubah dari tak terbaca ke yang baca)
AT+SMSO	Mengganti peralatan off
AT+SNFS	Memilih perangkat keras

AT+SNFV	Set volume
AT+SPBC	Melihat ke masukan pertama dalam urutan buku telepon yang dimulai dari surat yang terpilih (atau keberadaan selanjutnya)
AT+SPBG	Membaca ke masukan dari urutan buku telepon melalui urutan indeks
AT+SPBS	Memilih sebuah buku telepon (termasuk buku spesifikasi)
AT+SPIC	Pengeluaran counter PIN
AT+SPLM	Membaca PLMN
AT+SPLR	Membaca sebuah pemasukan dari operator yang ditunjuk
AT+SPLW	Menulis sebuah pemasukan ke operator yang ditunjuk
AT+SPST	Memainkan sinyal tone
AT+SPWD	Mengubah password ke sebuah kunci (termasuk kunci-kunci yang ditentukan pengguna)
AT+SRTC	Set bunyi kring (ringer)
AT+SSTK	Peralatan SIM
A/	Mengulang perintah penting sesudahnya
AT . . .	Awalan untuk semua perintah lainnya
ATA	Menerima panggilan (V.25)
ATB[n]	Perintah modem ini digunakan menset pembawa service untuk data koneksi (cf.AT+CBST). <n> dapat menggunakan salah satu nilai berikut : 7 2400 bps, asynchronous, V.22bis 11 4800bps, asynchronous, V.32 13 9600bps, asynchronous, 32 15 14400bps, asynchronous, V.34 25 2400bps, asynchronous, V.110 ISDN 27 4800bps, asynchronous, V.110 ISDN 29 9600bps, asynchronous, V.110 ISDN 31 14400bps, asynchronous, V.110 ISDN
ATD<str>	Dial abjad <str> dengan utiliti suara valid pengubah dial: I pembatasan AT+CLIR i meniadakan AT+CLIR untuk panggilan selanjutnya T tone dial P Pulse dial dibiarkan Karakteristik terakhir ";" menentukan ke telepon yang panggilannya harus di-setup dengan utiliti suara. Jika tidak, sebuah uji coba yaitu membuat setup sebuah data panggilan, yang telepon untuk segera memberitahu "ERROR". Perintah dial akan mengembalikan OK ke pengguna dengan segera sesudah memulai sebuah panggilan suara.

	Jika tidak, tanda# akan diurutkan dalam perintah dial, dan panggilan data tinggal tak berubah.
ATD<n>;	Dial nomor telepon dari buku telepon nomor lokasi <n>
ATD<mem><n>;	Dial nomor telepon dari buku telepon nomor lokasi <mem><n>
ATDx[:]	Dial nomor telepon x i ISDN Panggilan telepon akan membuat seperti sebuah panggilan UDI. Sebuah koneksi ISDN ke sebuah V.110 terminal adapter akan diselenggarakan. Kecepatan data transmisi sama seperti untuk sebuah panggilan analog (2400/4800/9600/14400bps). PP Plus: sama seperti + karakter
ATDL	Dial nomor telepon terakhir
ATE0	Pengurangan aktivasi perintah echo
ATE1	Mengaktivasi perintah echo
ATH[0]	Melepaskan koneksi yang ada
ATI[n]	Perintah modem sesuai [3]; Menampilkan kode produksi : 0 042 1 042 2 Ok, (check firmware checksum) 8 Menampilkan modus operasi yang disupport (lihat ATB) 9 Modem identifikasi dan mobile phone
ATL[n]	Monitor kekerasan speaker (perintah modem menurut ke [3])
ATM[m]	Monitor speaker mode (perintah modem menurut ke [3])
ATO[n]	Switch kembali ke transparent mode setelah +++ intrupsi (perintah modem menurut ke [3])
ATQ0	Menampilkan penerimaan (respons atau pesan)
ATQ1	Meniadakan pemberitahuan (respons atau pesan)
ATSn=x	Menulis nilai x ke S register n (perintah modem menurut[3])
ATSn?	Memperlihatkan nilai S register n (perintah modem menurut [3]) catatan : Tipe mobile tidak mengizinkan nilai dari S register untuk ditampilkan dengan sebuah perintah single
ATV0	Menampilkan penerimaan seperti angka-angka
ATV1	Menampilkan penerimaan seperti naskah
ATX<n>	Laporan link dengan hanya CONNECT biarkan sinyal sibuk <n> dapat menggunakan salah satu nilai berikut: 1 Laporan link dengan CONNECT tambah

	<p>baudrate, biarkan sinyal sibuk</p> <ol style="list-style-type: none"> 2 Sama seperti ATX1 3 Sama seperti ATX, tetapi laporan BUSY 4 Sama seperti ATX, tetapi laporan BUSY
ATZ	Set ke konfigurasi semula
AT&C[n]	<p>Sirkuit 109 (penerimaan baris sinyal pendeteksi/DCD) perilaku</p> <p><n> dapat mengambil salah satu nilai berikut :</p> <ol style="list-style-type: none"> 0 DCD selalu on 1 DCD ON jika pembawa terdeteksi
AT&D[n]	<p>Sirkuit 108 (Data Terminal Ready/DTR) perilaku</p> <p>Catatan : Perintah AT&D<n> dicirikan berikut sehingga tidak berpengaruh sejak sirkuit 108 tidak mendukung dalam tipe mobil phone ini. <n> dapat mengambil salah satu nilai berikut :</p> <ol style="list-style-type: none"> 0 DTR ignored 1 On DTR On to OFF: ke perintah online mode, tidak dikoneksi 2 On DTR ON to OFF: tidak dikoneksi ke perintah mode. Menjawab otomatis yaitu dimatikan ketika DTR OFF
AT&F[0]	<p>Reset semua parameter sementara dari beberapa perintah AT untuk profil pabrik</p> <p>ATE, ATQ, ATV, ATX</p> <p>AT+CBST, AT+CRLP, AT+CRC, AT+CR,</p> <p>AT+CNMI, AT+CMEE, AT+CSMS, AT+SCKS,</p> <p>AT+SACM, AT+CREG, AT+CLIP</p> <p>- S parameter</p> <p>- AT&D;AT&C;AT&S</p> <p>Beberapa koneksi yang dihadirkan akan terputus. Bukan perintah lain yang diterima pada baris perintah yang sama.</p>
\N	<p>Tidak ada aksi (\N2-\N6)</p> <p>\N2</p> <p>\N3</p> <p>\N4</p>
\Q<n	<p>Pilihan aliran kontrol lokal (DTE-DCE); dapat dikustom</p> <p><n> dapat diambil satu dari nilai berikut :</p> <ol style="list-style-type: none"> 0 Tanpa aliran kontrol 1 XON-XOFF aliran software kontrol 2 Hanya CTS aliran kontrol 3 RTS/CTS aliran kontrol
\V[n]	<p>Perintah modem :</p> <p>No /REL or /RLP lampiran dengan pesan koneksi /REL or /RLP lampiran dengan pesan koneksi</p>



+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Next-Generation Device Features

- ◆ For Low-Voltage, Integrated ESD Applications
MAX3222E/MAX3232E/MAX3237E/MAX3241E/
MAX3246E: +3.0V to +5.5V, Low-Power, Up to
1Mbps, True RS-232 Transceivers Using Four
0.1 μ F External Capacitors (MAX3246E Available
in a UCSP™ Package)
- ◆ For Low-Cost Applications
MAX221E: $\pm 15kV$ ESD-Protected, +5V, 1 μ A,
Single RS-232 Transceiver with AutoShutdown™

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μ F)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1480
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package



Maxim Integrated Products 1

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

MAX220–MAX249

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	..880mW
V+ (Note 1)	(V _{CC} - 0.3V) to +14V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	..440mW
V- (Note 1)	+0.3V to +14V	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	..696mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)	..762mW
T _{IN}	-0.3V to (V _{CC} - 0.3V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	..762mW
R _{IN} (Except MAX220)	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	..800mW
R _{IN} (MAX220)	±25V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	..640mW
T _{OUT} (Except MAX220) (Note 2)	±15V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)	..800mW
T _{OUT} (MAX220)	±13.2V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)	..842mW
Output Voltages		Operating Temperature Ranges	
T _{OUT}	±15V	MAX2__AC__ MAX2__C__0°C to +70°C
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	MAX2__AE__ MAX2__E__-40°C to +85°C
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2__AM__ MAX2__M__-55°C to +125°C
Continuous Power Dissipation (T _A = +70°C)		Storage Temperature Range-65°C to +160°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	..842mW	Lead Temperature (soldering, 10s) (Note 3)+300°C

Note 1: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Note 2: Input voltage measured with T_{OUT} in high-impedance state, SHDN = 0V or V_{CC} = 0V.

Note 3: Maximum reflow temperature for the MAX233A is +225°C.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1-C4 = 0.1µF, MAX220, C1 = 0.047µF, C2-C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS					
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND	±5	±8		V
Input Logic Threshold Low			1.4	0.8	V
Input Logic Threshold High	All devices except MAX220	2	1.4		V
	MAX220: V _{CC} = 5.0V	2.4			
Logic Pullup/Input Current	All except MAX220, normal operation		5	40	µA
	SHDN = 0V, MAX222/MAX242, shutdown, MAX220		±0.01	±1	
Output Leakage Current	V _{CC} = 5.5V, SHDN = 0V, V _{OUT} = ±15V, MAX222/MAX242		±0.01	±10	µA
	V _{CC} = SHDN = 0V		±0.01	±10	
	V _{OUT} = ±15V			±25	
	MAX220, V _{OUT} = ±12V				
Data Rate			200	116	kbps
Transmitter Output Resistance	V _{CC} = V+ = V- = 0V, V _{OUT} = ±2V	300	10M		Ω
Output Short-Circuit Current	V _{OUT} = 0V	±7	±22		mA
	MAX220			±60	
RS-232 RECEIVERS					
RS-232 Input Voltage Operating Range				±30	V
	MAX220			±25	
RS-232 Input Threshold Low	V _{CC} = 5V	All except MAX243 R _{2IN}	0.8	1.3	V
		MAX243 R _{2IN} (Note 4)	-3		
RS-232 Input Threshold High	V _{CC} = 5V	All except MAX243 R _{2IN}	1.8	2.4	V
		MAX243 R _{2IN} (Note 4)	-0.5	-0.1	

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX222-MAX243

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

($V_{CC} = +5V \pm 10\%$, $C1-C4 = 0.1\mu F$, MAX220, $C1 = 0.047\mu F$, $C2-C4 = 0.33\mu F$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Hysteresis	All except MAX220/MAX243, $V_{CC} = 5V$, no hysteresis in SHDN		0.2	0.5	1.0	V
	MAX220		0.3			
	MAX243		1			
RS-232 Input Resistance	$T_A = +25^\circ C$ (MAX220)		3	5	7	k Ω
			3	5	7	
TTL/CMOS Output Voltage Low	$I_{OUT} = 3.2mA$		0.2		0.4	V
	$I_{OUT} = 1.6mA$ (MAX220)		0.4			
TTL/CMOS Output Voltage High	$I_{OUT} = -1.0mA$		3.5	$V_{CC} - 0.2$		V
TTL/CMOS Output Short-Circuit Current	Sourcing $V_{OUT} = GND$		-2	-10		mA
	Sinking $V_{OUT} = V_{CC}$		10	30		
TTL/CMOS Output Leakage Current	$\overline{SHDN} = V_{CC}$ or $\overline{EN} = V_{CC}$ ($\overline{SHDN} = 0V$ for MAX222), $0V \leq V_{OUT} \leq V_{CC}$		± 0.05		± 10	μA
EN Input Threshold Low	MAX242		1.4		0.9	V
EN Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5	5.5		V
V _{CC} Supply Current ($\overline{SHDN} = V_{CC}$), figures 5, 6, 11, 19	No load	MAX220	0.5		2	μA
		MAX222/MAX232A/MAX233A/MAX242/MAX243	4		10	
	3k Ω load both inputs	MAX220	12			
		MAX222/MAX232A/MAX233A/MAX242/MAX243	15			
Shutdown Supply Current	MAX222/MAX242	$T_A = +25^\circ C$	0.1	10		μA
		$T_A = 0^\circ C$ to $+70^\circ C$	2	50		
		$T_A = -40^\circ C$ to $+85^\circ C$	2	50		
		$T_A = -55^\circ C$ to $+125^\circ C$	35	100		
\overline{SHDN} Input Leakage Current	MAX222/MAX242				± 1	μA
\overline{SHDN} Threshold Low	MAX222/MAX242		1.4		0.9	V
\overline{SHDN} Threshold High	MAX222/MAX242		2.0	1.4		V
Transition Slew Rate	$C_L = 50pF$ to $2500pF$, $R_L = 3k\Omega$ to $7k\Omega$, $V_{CC} = 5V$, $T_A = +25^\circ C$, measured from $+3V$ to $-3V$ or $-3V$	MAX222/MAX232A/MAX233A/MAX242/MAX243	6	12	30	V/ μs
		MAX220	1.5	3	30.0	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t_{PHLT}	MAX222/MAX232A/MAX233A/MAX242/MAX243	1.3		3.5	μs
		MAX220	4		10	
	t_{PLHT}	MAX222/MAX232A/MAX233A/MAX242/MAX243	1.5		3.5	
		MAX220	5		10	

Note 4: MAX243 R_{2OUT} is guaranteed to be low when R_{2IN} is $\geq 0V$ or is floating.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

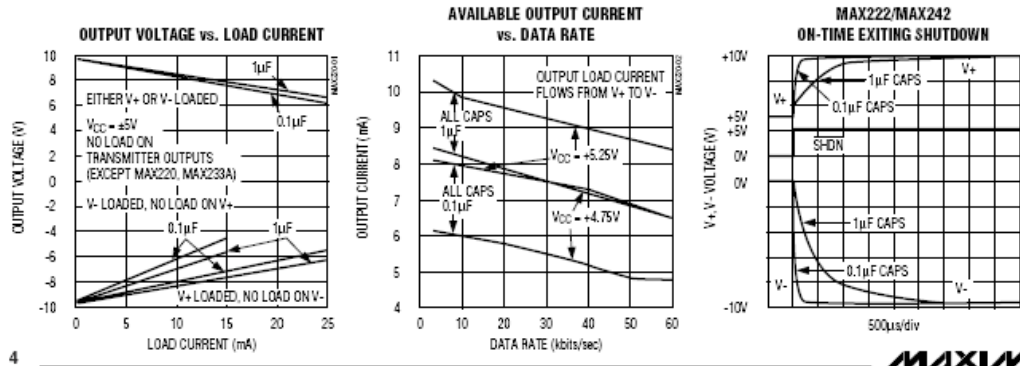
ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

($V_{CC} = +5V \pm 10\%$, C1–C4 = 0.1 μF , MAX220, C1 = 0.047 μF , C2–C4 = 0.33 μF , $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Receiver Propagation Delay RS-232 to TLL (Normal Operation), Figure 2	t _{PHLR}	MAX222/MAX232A/MAX233/ MAX242/MAX243	0.5	1		μs
		MAX220	0.6	3		
	t _{PLHR}	MAX222/MAX232A/MAX233/ MAX242/MAX243	0.6	1		
		MAX220	0.8	3		
Receiver Propagation Delay RS-232 to TLL (Shutdown), Figure 2	t _{PHLS}	MAX242	0.5	10		μs
	t _{PHLS}	MAX242	2.5	10		
Receiver-Output Enable Time, Figure 3	t _{ER}	MAX242	125	500		ns
Receiver-Output Disable Time, Figure 3	t _{DR}	MAX242	160	500		ns
Transmitter-Output Enable Time (\overline{SHDN} Goes High), Figure 4	t _{ET}	MAX222/MAX242, 0.1 μF caps (includes charge-pump start-up)		250		μs
Transmitter-Output Disable Time (\overline{SHDN} Goes Low), Figure 4	t _{DT}	MAX222/MAX242, 0.1 μF caps		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t _{PHLT} - t _{PLHT}	MAX222/MAX232A/MAX233/ MAX242/MAX243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	t _{PHLR} - t _{PLHR}	MAX222/MAX232A/MAX233/ MAX242/MAX243		100		ns
		MAX220		225		

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243



Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 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
- High Endurance Non-volatile Memory segments
 - 16K Bytes of In-System Self-programmable Flash program memory
 - 512 Bytes EEPROM
 - 1K Byte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C¹
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- 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 in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - 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, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega16L
 - 4.5 - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
 - Active: 1.1 mA
 - Idle Mode: 0.35 mA
 - Power-down Mode: < 1 µA



8-bit AVR[®]
Microcontroller
with 16K Bytes
In-System
Programmable
Flash

ATmega16
ATmega16L

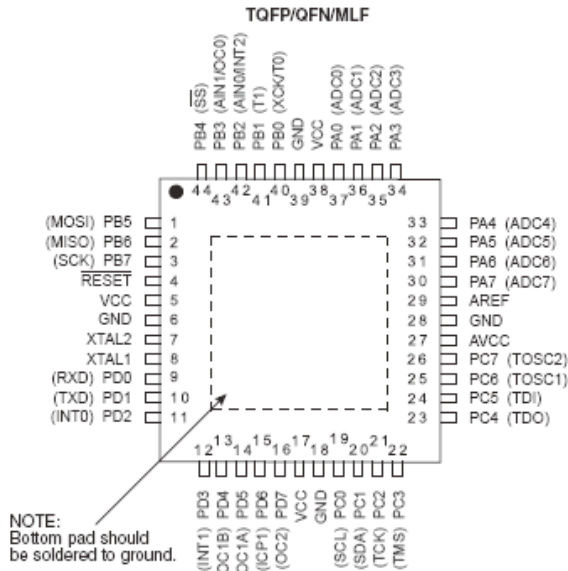
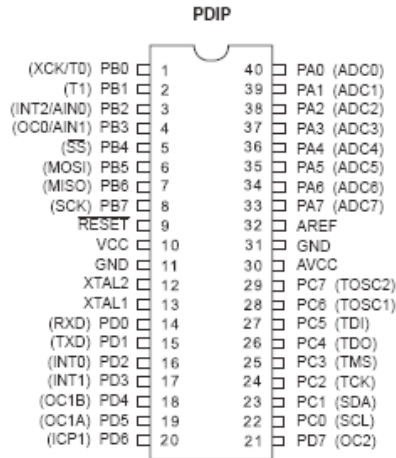
Summary





Pin Configurations

Figure 1. Pinout ATmega16



Disclaimer

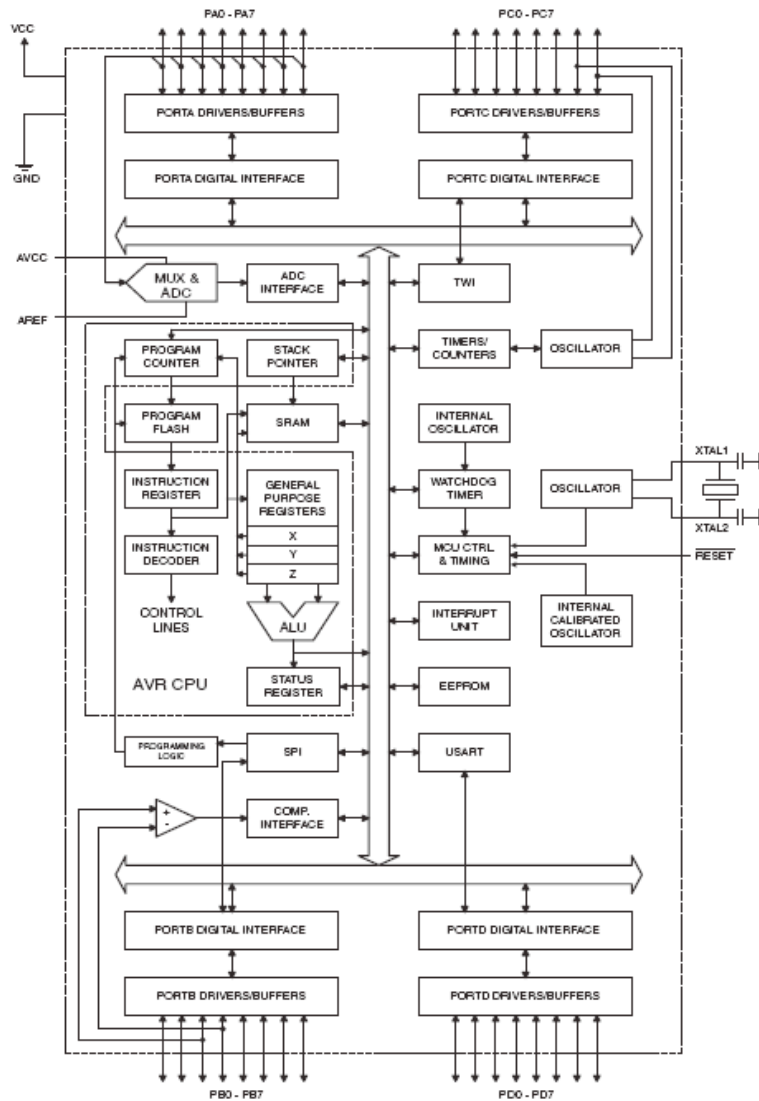
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$3F (\$5F)	SREG	I	T	H	S	V	N	Z	C	9
\$3E (\$5E)	SPH	-	-	-	-	-	SP10	SP9	SP8	12
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
\$3C (\$5C)	OCRO	Timer/Counter0 Output Compare Register								86
\$3B (\$5B)	GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	48, 69
\$3A (\$5A)	GIFR	INTF1	INTF0	INTF2	-	-	-	-	-	70
\$39 (\$59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	86, 116, 133
\$38 (\$58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	86, 116, 133
\$37 (\$57)	SPMCR	SPMIE	RWWSE	-	RWWRE	BLBSET	PGWRT	PGERS	SPMEN	260
\$36 (\$56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	180
\$35 (\$55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	32, 68
\$34 (\$54)	MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF	41, 69, 231
\$33 (\$53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	83
\$32 (\$52)	TCNT0	Timer/Counter0 (8 Bits)								86
\$31 ⁽¹⁾ (\$51 ⁽¹⁾)	OSCCAL	Oscillator Calibration Register								30
	ODDR	On-Chip Debug Register								227
\$30 (\$50)	SPOR	ADTS2	ADTS1	ADTS0	-	ACME	PUD	POR2	POR10	67, 86, 134, 201, 221
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	110
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	113
\$2D (\$4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								114
\$2C (\$4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								114
\$2B (\$4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								114
\$2A (\$4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								114
\$29 (\$49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								114
\$28 (\$48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								114
\$27 (\$47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								114
\$26 (\$46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								114
\$25 (\$45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	128
\$24 (\$44)	TCNT2	Timer/Counter2 (8 Bits)								130
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register								130
\$22 (\$42)	ADSR	-	-	-	-	AS2	TON2UB	OCR2UB	TOR2UB	131
\$21 (\$41)	WDTCR	-	-	-	WDTOE	WDE	WDF2	WDF1	WDF0	43
\$20 ⁽²⁾ (\$40 ⁽²⁾)	UBRRH	URSEL	-	-	-	UBRR(11:8)				167
	UCSR0	URSEL	UMSEL	UPM1	UPM0	USB0	UCS21	UCS20	UCPOL	166
\$1F (\$3F)	EEARH	-	-	-	-	-	-	-	EEAR8	19
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte								19
\$1D (\$3D)	EEDR	EEPROM Data Register								19
\$1C (\$3C)	EEDR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	19
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	66
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	66
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	66
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	66
\$17 (\$37)	DDRB	ddb7	ddb6	ddb5	ddb4	ddb3	ddb2	ddb1	ddb0	66
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	66
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	67
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	67
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	67
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	67
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	67
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	67
\$0F (\$2F)	SPDR	SPI Data Register								142
\$0E (\$2E)	SPR	SPIF	WCOL	-	-	-	-	-	SPI2X	142
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	140
\$0C (\$2C)	UDR	USART I/O Data Register								163
\$0B (\$2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	164
\$0A (\$2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCS22	RXB8	TXB8	165
\$09 (\$29)	UBRRL	USART Baud Rate Register Low Byte								167
\$08 (\$28)	ADSR	ACD	ACBG	AD0	AD1	AD2	AD3	AD4	AD5	202
\$07 (\$27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	217
\$06 (\$26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	219
\$05 (\$25)	ADCH	ADC Data Register High Byte								220
\$04 (\$24)	ADCL	ADC Data Register Low Byte								220
\$03 (\$23)	TWDR	Two-wire Serial Interface Data Register								182
\$02 (\$22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWSE	182



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	Rd,K	Add Immediate to Word	$Roh\ Rd \leftarrow Roh\ Rd + K$	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	Rd,K	Subtract immediate from Word	$Roh\ Rd \leftarrow Roh\ Rd - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \wedge Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \wedge K$	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	$Rd \leftarrow \text{SFF} - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \text{SDD} - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set BR(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear BR(s) in Register	$Rd \leftarrow Rd \wedge (\text{SFF} - K)$	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	$Rd \leftarrow Rd \wedge Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow \text{SFF}$	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
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow \text{STACK}$	None	4
RETI		Interrupt Return	$PC \leftarrow \text{STACK}$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	$\text{if } (Rd = Rr) \text{ PC} \leftarrow PC + 2 \text{ or } 3$	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	$\text{if } (Rr[b]=0) \text{ PC} \leftarrow PC + 2 \text{ or } 3$	None	1 / 2 / 3
SBRB	Rr, b	Skip if Bit in Register is Set	$\text{if } (Rr[b]=1) \text{ PC} \leftarrow PC + 2 \text{ or } 3$	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	$\text{if } (P[b]=0) \text{ PC} \leftarrow PC + 2 \text{ or } 3$	None	1 / 2 / 3
SBIB	P, b	Skip if Bit in I/O Register is Set	$\text{if } (P[b]=1) \text{ PC} \leftarrow PC + 2 \text{ or } 3$	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	$\text{if } (SREG[s] = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	$\text{if } (SREG[s] = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRBEQ	k	Branch if Equal	$\text{if } (Z = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRNE	k	Branch if Not Equal	$\text{if } (Z = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRCS	k	Branch if Carry Set	$\text{if } (C = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRCC	k	Branch if Carry Cleared	$\text{if } (C = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRSH	k	Branch if Same or Higher	$\text{if } (C = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRLO	k	Branch if Lower	$\text{if } (C = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRMI	k	Branch if Minus	$\text{if } (N = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRPL	k	Branch if Plus	$\text{if } (N = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	$\text{if } (N \oplus V = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	$\text{if } (N \oplus V = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	$\text{if } (H = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	$\text{if } (H = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRTS	k	Branch if T Flag Set	$\text{if } (T = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRTC	k	Branch if T Flag Cleared	$\text{if } (T = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	$\text{if } (V = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	$\text{if } (V = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1 / 2



Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch If Interrupt Enabled	$\text{if } (I = 1) \text{ then } PC \leftarrow PC - k + 1$	None	1/2
BRID	k	Branch If Interrupt Disabled	$\text{if } (I = 0) \text{ then } PC \leftarrow PC - k + 1$	None	1/2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
MOVW	Rd, Rr	Copy Register Word	$Rd \leftarrow Rr; Rr \leftarrow Rr + 1$	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	$RD \leftarrow (Z)$	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
SFM		Store Program Memory	$(Z) \leftarrow R1; RD$	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
SSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
SBT	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

ATmega16(L)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
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