LAMPIRAN - LAMPIRAN

LAMPIRAN 1

LISTING PROGRAM VB.NET

Form Utama

```
Imports System.Runtime.InteropServices
Imports AXmsCtrl
Imports Dundas.Gauges.WinControl
Imports System.Text
Imports System.Data.SqlClient

PublicClass FrmUtama

Inherits System.Windows.Forms.Form
Const WM_CAP_START = &H400S
Const WS_CHILD = &H40000000
Const WS_VISIBLE = &H10000000

Const WM_CAP_DRIVER_CONNECT = WM_CAP_START + 10
Const WM_CAP_DRIVER_DISCONNECT = WM_CAP_START + 11
Const WM_CAP_EDIT_COPY = WM_CAP_START + 30
Const WM_CAP_SEQUENCE = WM_CAP_START + 62
Const WM_CAP_FILE_SAVEAS = WM_CAP_START + 23

Const WM_CAP_SET_SCALE = WM_CAP_START + 53
Const WM_CAP_SET_PREVIEWRATE = WM_CAP_START + 52
Const WM_CAP_SET_PREVIEW = WM_CAP_START + 50

Const SWP_NOMOVE = &H2S
Const SWP_NOSIZE = 1
Const SWP_NOZORDER = &H4S
Const HWND_BOTTOM = 1

'--The capGetDriverDescription function retrieves the version
' description of the capture driver--
DeclareFunction capGetDriverDescriptionA Lib"avicap32.dll" _
       (ByVal wDriverIndex AsShort, _
ByVal lpszName AsString, ByVal cbName AsInteger, _
ByVal lpszVer AsString, _
ByVal cbVer AsInteger) AsBoolean

'--The capCreateCaptureWindow function creates a capture window--
DeclareFunction capCreateCaptureWindowA Lib"avicap32.dll" _
       (ByVal lpszWindowName AsString, ByVal dwStyle AsInteger, _
ByVal x AsInteger, ByVal y AsInteger, ByVal nWidth AsInteger, _
ByVal nHeight AsShort, ByVal hWnd AsInteger, _
ByVal nID AsInteger) AsInteger

'--This function sends the specified message to a window or windows--
DeclareFunction SendMessage Lib"user32"Alias"SendMessageA" _
       (ByVal hwnd AsInteger, ByVal Msg AsInteger, _
ByVal wParam AsInteger, _
<MarshalAs(UnmanagedType.AsAny)>ByVal lParam AsObject) AsInteger

'--Sets the position of the window relative to the screen buffer--
DeclareFunction SetWindowPos Lib"user32"Alias"SetWindowPos" _
       (ByVal hwnd AsInteger, ByVal hWndInsertAfter AsInteger, ByVal x
AsInteger, _
ByVal y AsInteger, ByVal cx AsInteger, ByVal cy AsInteger, _
ByVal wFlags AsInteger) AsInteger

'--This function destroys the specified window--
DeclareFunction DestroyWindow Lib"user32" _
```

```
        (ByVal hndw AsInteger) AsBoolean

'---used as a window handle---
Private hWnd AsInteger

'Serial
PrivateWithEvents serialPort AsNew IO.Ports.SerialPort

'sms gateway
Private objGsmProtocol As SmsProtocolGsm
Private objConstants As SmsConstants
Dim objSmsMessage As SmsMessage
Dim perintah AsString

Dim buffer_susu(100) AsString
Dim buffer_ldr1(100) AsString
Dim buffer_ldr2(100) AsString
Dim buffer_pir(100) AsString
Dim sampel AsInteger

Dim s AsString
Dim suhu AsString
Dim pir AsString
Dim pir2 AsString
Dim ldr1 AsString
Dim ldr12 AsString
Dim ldr2 AsString

'database
Dim sqlAdapter AsNew SqlClient.SqlDataAdapter
Dim sqlConnection AsNew SqlClient.SqlConnection
Dim dataset AsNew DataSet

Sub ambil_gambar()
Dim data As IDataObject
Dim bmap As Image

'---copy the image to the clipboard---
        SendMessage(hWnd, WM_CAP_EDIT_COPY, 0, 0)

'---retrieve the image from clipboard and convert it
' to the bitmap format
        data = Clipboard.GetDataObject()
If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
            bmap = _
CType(data.GetData(GetType(System.Drawing.Bitmap)),  _
                Image)
            bmap.Save("C:\"& Now.ToFileTime &".bmp")
EndIf
EndSub
Sub termometer(ByVal suhunya)
' Get linear pointer object.
Dim pointer As LinearPointer =
GaugeContainer1.LinearGauges("GalleryGauge").Pointers("Default")

' Set thermometer style to Flask.
        pointer.ThermometerStyle = ThermometerStyle.Flask

' Set thermometer bulb size.
        pointer.ThermometerBulbSize = 30
        pointer.Width = 15
```

```
' Set thermometer appearance.
        pointer.FillColor = Color.Red
        pointer.ThermometerBackColor = Color.White
        GaugeContainer1.LinearGauges("GalleryGauge").Size.Height = 100
        GaugeContainer1.LinearGauges("GalleryGauge").Size.Width = 100
' Set the value of the pointer to the generated random number.

GaugeContainer1.LinearGauges("GalleryGauge").Pointers("Default").Value =
suhunya

' Set the value of the numeric indicator to the generated random number.
        GaugeContainer1.NumericIndicators("NumericIndicator1").Value =
suhunya
EndSub
Sub ldrIndikator1(ByVal ldr1nya)
' Set the value of the pointer to the generated random number.
        GaugeContainer2.CircularGauges("Default").Pointers("Default").Value
= ldr1nya

' Set the value of the numeric indicator to the generated random number.
        GaugeContainer2.NumericIndicators("NumericIndicator1").Value =
ldr1nya
EndSub
Sub ldrIndikator2(ByVal ldr2nya)
' Set the value of the pointer to the generated random number.
        GaugeContainer3.CircularGauges("Default").Pointers("Default").Value
= ldr2nya

' Set the value of the numeric indicator to the generated random number.
        GaugeContainer3.NumericIndicators("NumericIndicator1").Value =
ldr2nya
EndSub
Sub R1_nyala()
Try
            serialPort.Write("1"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Relay1 Nyala")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R1_mati()
Try
            serialPort.Write("a"& vbCrLf)
            masukankegiatan("Relay1 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R2_nyala()
Try
            serialPort.Write("2"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Rela2 Nyala")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R2_mati()
Try
            serialPort.Write("b"& vbCrLf)
```

```
            masukankegiatan("Relay2 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R3_nyala()
Try
            serialPort.Write("3"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Relay3 Nyala")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R3_mati()
Try
            serialPort.Write("c"& vbCrLf)
            masukankegiatan("Relay3 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R4_nyala()
Try
            serialPort.Write("4"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Relay4 Nyala")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R4_mati()
Try
            serialPort.Write("d"& vbCrLf)
            masukankegiatan("Relay4 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R5_nyala()
Try
            serialPort.Write("5"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Relay5 Nyala")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R5_mati()
Try
            serialPort.Write("e"& vbCrLf)
            masukankegiatan("Relay5 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R6_nyala()
Try
            serialPort.Write("6"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Relay6 Nyala")
```

```
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R6_mati()
Try
            serialPort.Write("f"& vbCrLf)
            masukankegiatan("Relay6 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R7_nyala()
Try
            serialPort.Write("7"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Relay7 Nyala")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R7_mati()
Try
            serialPort.Write("g"& vbCrLf)
            masukankegiatan("Relay7 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R8_nyala()
Try
            serialPort.Write("8"& vbCrLf)
'serialPort.Write(txtDataToSend.Text)
            masukankegiatan("Relay8 Nyala")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub R8_mati()
Try
            serialPort.Write("h"& vbCrLf)
            masukankegiatan("Relay8 Mati")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub motor1()
Try
            serialPort.Write("m"& vbCrLf)
            masukankegiatan("Gerakan Motor1")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub
Sub motor2()
Try
            serialPort.Write("n"& vbCrLf)
            masukankegiatan("Gerakan Motor2")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
```

```
EndSub
Sub kirimsms(ByVal tujuan AsString)
Try
Dim objSmsMessage As SmsMessage = New SmsMessage
Dim strMessageReference AsString

            objSmsMessage.Clear()
            objSmsMessage.Recipient = tujuan
            objSmsMessage.Data = "lokasi"
            objSmsMessage.Format = objConstants.asMESSAGEFORMAT_TEXT


            objGsmProtocol.Clear()
            objGsmProtocol.Device = CbPortHP.SelectedItem.ToString()
If CbSpeedHP.SelectedIndex = 0 Then
                objGsmProtocol.DeviceSpeed = 0
Else
                objGsmProtocol.DeviceSpeed =
System.Int32.Parse(CbSpeedHP.SelectedItem.ToString())
EndIf
' Set Cursor
            System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.WaitCursor

' Send the message !
            strMessageReference = objGsmProtocol.Send(objSmsMessage)

' Reset Cursor
            System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.Default
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry

EndSub

PrivateSub KeluarToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles KeluarToolStripMenuItem.Click
End
EndSub

PrivateSub FrmUtama_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) HandlesMyBase.Load
        FrmLogin.ShowDialog()
        PreviewVideo(PictureBox3)
'koneksi serial
        CbPortSerial.Items.Clear()
For i AsInteger = 0 ToMy.Computer.Ports.SerialPortNames.Count - 1
            CbPortSerial.Items.Add(My.Computer.Ports.SerialPortNames(i))
Next
        BtnDSerial.Enabled = False
        CbRateSerial.Items.Add("9600")



'koneksi HP
Dim numDeviceCount, l As System.Int32
Dim strDevice AsString

        objGsmProtocol = New SmsProtocolGsm
        objConstants = New SmsConstants
```

```
' Fill devices combo
        CbPortHP.Items.Clear()
        numDeviceCount = objGsmProtocol.GetDeviceCount()
For l = 0 To numDeviceCount - 1
            strDevice = objGsmProtocol.GetDevice(l)
            CbPortHP.Items.Add(strDevice)
Next
        CbPortHP.Items.Add("COM1")
        CbPortHP.Items.Add("COM2")
        CbPortHP.Items.Add("COM3")
        CbPortHP.Items.Add("COM4")
        CbPortHP.Items.Add("COM5")
        CbPortHP.Items.Add("COM6")
        CbPortHP.Items.Add("COM7")
        CbPortHP.Items.Add("COM8")
'CbPortHP.SelectedIndex = 0

' Fill devicespeed combo
        CbSpeedHP.Items.Clear()
        CbSpeedHP.Items.Add("Default")
        CbSpeedHP.Items.Add("1200")
        CbSpeedHP.Items.Add("2400")
        CbSpeedHP.Items.Add("4800")
        CbSpeedHP.Items.Add("9600")
        CbSpeedHP.Items.Add("19200")
        CbSpeedHP.Items.Add("38400")
        CbSpeedHP.Items.Add("57600")
        CbSpeedHP.Items.Add("115200")
'CbSpeedHP.SelectedIndex = 0
        BtnDHP.Enabled = False

'database
        sqlConnection.ConnectionString = "Data Source=JIMMYREKSO-
PC\SQLEXPRESS;Initial Catalog=DbTa;Integrated Security=True"
'sqlAdapter.SelectCommand = New SqlClient.SqlCommand("Select * From
Customer", sqlConnection)

EndSub
PrivateSub PreviewVideo(ByVal pbCtrl As PictureBox)
        hWnd = capCreateCaptureWindowA(0, _
            WS_VISIBLE Or WS_CHILD, 0, 0, 0, _
            0, pbCtrl.Handle.ToInt32, 0)
If SendMessage( _
           hWnd, WM_CAP_DRIVER_CONNECT, _
           0, 0) Then
'---set the preview scale---
            SendMessage(hWnd, WM_CAP_SET_SCALE, True, 0)
'---set the preview rate (ms)---
            SendMessage(hWnd, WM_CAP_SET_PREVIEWRATE, 30, 0)
'---start previewing the image---
            SendMessage(hWnd, WM_CAP_SET_PREVIEW, True, 0)
'---resize window to fit in PictureBox control---
            SetWindowPos(hWnd, HWND_BOTTOM, 0, 0, _
               pbCtrl.Width, pbCtrl.Height, _
               SWP_NOMOVE Or SWP_NOZORDER)
Else
'--error connecting to video source---
            DestroyWindow(hWnd)
EndIf
EndSub
```

```
PrivateSub BtnCSerial_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnCSerial.Click
If serialPort.IsOpen Then
            serialPort.Close()
EndIf
Try
With serialPort
                .PortName = CbPortSerial.Text
                .BaudRate = Val(CbRateSerial.Text)
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
EndWith
            serialPort.Open()
            System.Threading.Thread.Sleep(10)



            LbSSerial.Text = CbPortSerial.Text &" connected."
            BtnCSerial.Enabled = False
            BtnDSerial.Enabled = True
            TGauge.Enabled = True
            CbPortSerial.Enabled = False
            CbRateSerial.Enabled = False

'TDatabase.Enabled = True
'TSerial.Enabled = True

            GroupBox4.Enabled = True
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndSub

PrivateSub BtnDSerial_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnDSerial.Click
Try
            serialPort.Close()
            LbSSerial.Text = serialPort.PortName &" disconnected."
            BtnCSerial.Enabled = True
            BtnDSerial.Enabled = False
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry

        TGauge.Enabled = False
        GroupBox4.Enabled = False
'TDatabase.Enabled = False
'TSerial.Enabled = False
        CbPortSerial.Enabled = True
        CbRateSerial.Enabled = True
EndSub

PrivateSub DataReceived(ByVal sender AsObject, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
serialPort.DataReceived
        System.Threading.Thread.Sleep(10)
        sampel = 0
```

```
        txtDataReceived.BeginInvoke(New myDelegate(AddressOf
updateTextBox), NewObject() {})

        LbSerial.BeginInvoke(New myDelegate2(AddressOf updateTextBox2),
NewObject() {})


'System.Threading.Thread.Sleep(10)

        System.Threading.Thread.Sleep(5)

EndSub
PublicDelegateSub myDelegate()
PublicSub updateTextBox()

        System.Threading.Thread.Sleep(5)
        System.Threading.Thread.Sleep(5)
'---for receiving plan ASCII text---
With txtDataReceived
            .AppendText(serialPort.ReadExisting)
            .ScrollToCaret()
EndWith

If pir2 = "1"Then
            LbPir.Text = "Ada orang masuk"
            LbPir.ForeColor = Color.Red
            ambil_gambar()
Else
            LbPir.Text = "Kosong"
            LbPir.ForeColor = Color.Black
EndIf

      Ifval(suhu)>50Then
Try
Dim objSmsMessage As SmsMessage = New SmsMessage
Dim strMessageReference AsString

            objSmsMessage.Clear()
            objSmsMessage.Recipient = tujuan
            objSmsMessage.Data = "Bahaya, Suhu diatas 50 derajat celsius"
            objSmsMessage.Format = objConstants.asMESSAGEFORMAT_TEXT


            objGsmProtocol.Clear()
            objGsmProtocol.Device = CbPortHP.SelectedItem.ToString()
If CbSpeedHP.SelectedIndex = 0 Then
                objGsmProtocol.DeviceSpeed = 0
Else
                objGsmProtocol.DeviceSpeed =
System.Int32.Parse(CbSpeedHP.SelectedItem.ToString())
EndIf
' Set Cursor
            System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.WaitCursor

' Send the message !
            strMessageReference = objGsmProtocol.Send(objSmsMessage)

' Reset Cursor
            System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.Default
```

```
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry
EndIf

EndSub

PublicDelegateSub myDelegate2()
PublicSub updateTextBox2()
        System.Threading.Thread.Sleep(5)

Dim data_baris AsString =
txtDataReceived.Lines(txtDataReceived.Lines.Length - 2)

        LbSerial.Text = data_baris
        System.Threading.Thread.Sleep(5)

        s = Microsoft.VisualBasic.Right(LbSerial.Text, "17")
        suhu = Microsoft.VisualBasic.Left(s, "5")
        pir = Microsoft.VisualBasic.Left(s, "7")
        pir2 = Microsoft.VisualBasic.Right(pir, "1")
        ldr1 = Microsoft.VisualBasic.Right(s, "9")
        ldr12 = Microsoft.VisualBasic.Left(ldr1, "4")
        ldr2 = Microsoft.VisualBasic.Right(s, "4")


EndSub

PrivateSub TGauge_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TGauge.Tick
        System.Threading.Thread.Sleep(5)
        termometer(Val(suhu))
        ldrIndikator1(Val(ldr12))
        ldrIndikator2(Val(ldr2))
        masukansensor()
'Application.DoEvents()
EndSub


PrivateSub BtnOn1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn1.Click
        R1_nyala()
EndSub

PrivateSub BtnOff1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff1.Click
        R1_mati()
EndSub

PrivateSub BtnOn2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn2.Click
        R2_nyala()
EndSub

PrivateSub BtnOff2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff2.Click
        R2_mati()
EndSub

PrivateSub BtnOn3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn3.Click
```

```
        R3_nyala()
EndSub

PrivateSub BtnOff3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff3.Click
        R3_mati()
EndSub

PrivateSub BtnOn4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn4.Click
        R4_nyala()
EndSub

PrivateSub BtnOff4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff4.Click
        R4_mati()
EndSub

PrivateSub BtnOn5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn5.Click
        R5_nyala()
EndSub

PrivateSub BtnOff5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff5.Click
        R5_mati()
EndSub

PrivateSub BtnOn6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn6.Click
        R6_nyala()
EndSub

PrivateSub BtnOff6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff6.Click
        R6_mati()
EndSub

PrivateSub BtnOn7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn7.Click
        R7_nyala()
EndSub

PrivateSub BtnOff7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff7.Click
        R7_mati()
EndSub

PrivateSub BtnOn8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOn8.Click
        R8_nyala()
EndSub

PrivateSub BtnOff8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOff8.Click
        R8_mati()
EndSub

PrivateSub BtnCHP_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnCHP.Click
        TSms.Enabled = True
```

```
        BtnCHP.Enabled = False
        BtnDHP.Enabled = True
        Application.DoEvents()
        CbPortHP.Enabled = False
        CbSpeedHP.Enabled = False
EndSub

PrivateSub BtnDHP_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnDHP.Click
        TSMS.Enabled = False
        BtnCHP.Enabled = True
        BtnDHP.Enabled = False
        CbPortHP.Enabled = True
        CbSpeedHP.Enabled = True
EndSub

PrivateSub TSMS_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TSMS.Tick
Dim objSmsMessage As SmsMessage
        objSmsMessage = objGsmProtocol.GetNextMessage()

' Clear all items in list
        TextBox3.Text = ""


' Device
        objGsmProtocol.Clear()
        objGsmProtocol.Device = CbPortHP.SelectedItem.ToString()

' Device speed
If CbSpeedHP.SelectedIndex = 0 Then
            objGsmProtocol.DeviceSpeed = 0
Else
            objGsmProtocol.DeviceSpeed =
System.Int32.Parse(CbSpeedHP.SelectedItem.ToString())
EndIf


        objGsmProtocol.DeleteAfterReceive = True
        objGsmProtocol.MessageStorage = objConstants.asSTORAGE_ALL
' Set memory to all available storage locations
        objGsmProtocol.Receive()
' Receive all messages from all memory locations

If (objGsmProtocol.LastError <> 0) Then
            Label16.Text = ("Failed to receive, error: "&
objGsmProtocol.LastError)
EndIf
If objGsmProtocol.LastError = 0 Then


            objSmsMessage = objGsmProtocol.GetFirstMessage()
' Get first message

If (objGsmProtocol.LastError = 0) Then
'WScript.Echo("Message from: " & objSmsMessage.Sender)
' Show sender's mobile number
                TextBox4.Text = objSmsMessage.Sender
                TextBox3.Text = objSmsMessage.Data                          '
Show the SMS message text
                perintah = objSmsMessage.Data
```

```
'objSmsMessage = objGsmProtocol.GetNextMessage()                    ' Get next
message
EndIf
EndIf
EndSub

PrivateSub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox3.TextChanged
If TextBox3.Text <>""Then
            Lbsender.Text = TextBox4.Text
            Lbsms.Text = TextBox3.Text
            TextBox4.Text = ""
EndIf
EndSub
PrivateSub Lbsms_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Lbsms.TextChanged
Dim sistem AsString
Dim objek AsString
Dim perintah AsString
Dim LokStat AsString

Dim smslat AsString
Dim smslong AsString
Dim titikdua AsString
Dim koma AsString
Dim akhir AsString

        sistem = Microsoft.VisualBasic.Left(Lbsms.Text, "10")
        objek = Microsoft.VisualBasic.Mid(Lbsms.Text, 12, 2)
        perintah = Microsoft.VisualBasic.Mid(Lbsms.Text, 15, 2)
        LokStat = Microsoft.VisualBasic.Left(Lbsms.Text, "6")



If sistem = "Jimmyrekso"Then

If objek = "R1"Then
If perintah = "on"Then
                    R1_nyala()
Else
                    R1_mati()
EndIf
EndIf

If objek = "R2"Then
If perintah = "on"Then
                    R2_nyala()
Else
                    R2_mati()
EndIf
EndIf

If objek = "R3"Then
If perintah = "on"Then
                    R3_nyala()
Else
                    R3_mati()
EndIf
EndIf

If objek = "R4"Then
```

```
If perintah = "on"Then
                    R4_nyala()
Else
                    R4_mati()
EndIf
EndIf

If objek = "R5"Then
If perintah = "on"Then
                    R5_nyala()
Else
                    R5_mati()
EndIf
EndIf

If objek = "R6"Then
If perintah = "on"Then
                    R6_nyala()
Else
                    R6_mati()
EndIf
EndIf

If objek = "R7"Then
If perintah = "on"Then
                    R7_nyala()
Else
                    R7_mati()
EndIf
EndIf

If objek = "R8"Then
If perintah = "on"Then
                    R8_nyala()
Else
                    R8_mati()
EndIf
EndIf

If objek = "M1"Then
                motor1()
EndIf

If objek = "M2"Then
                motor2()
EndIf

Else

If LokStat = "lokasi"Then
                TabKoneksi.TabPages(5).Show()
                titikdua = Lbsms.Text.IndexOf(":")
                koma = Lbsms.Text.IndexOf(",")
                akhir = Lbsms.Text.IndexOf(";")
                smslat = Microsoft.VisualBasic.Mid(Lbsms.Text, titikdua +
2, koma - (titikdua + 1))
                smslong = Microsoft.VisualBasic.Mid(Lbsms.Text, koma + 2,
akhir - (koma + 1))

                txtLat.Text = smslat
                txtLong.Text = smslong
```

```
                lokasinya(smslat, smslong)
Else
If LokStat = "status"Then
Try
Dim objSmsMessage As SmsMessage = New SmsMessage
Dim strMessageReference AsString

                        objSmsMessage.Clear()
                        objSmsMessage.Recipient = Lbsender.Text
                        objSmsMessage.Data = s
                        objSmsMessage.Format =
objConstants.asMESSAGEFORMAT_TEXT


                        objGsmProtocol.Clear()
                        objGsmProtocol.Device =
CbPortHP.SelectedItem.ToString()
If CbSpeedHP.SelectedIndex = 0 Then
                            objGsmProtocol.DeviceSpeed = 0
Else
                            objGsmProtocol.DeviceSpeed =
System.Int32.Parse(CbSpeedHP.SelectedItem.ToString())
EndIf
' Set Cursor
                        System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.WaitCursor

' Send the message !
                        strMessageReference =
objGsmProtocol.Send(objSmsMessage)

' Reset Cursor
                        System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.Default
Catch ex As Exception
                        MsgBox(ex.ToString)
EndTry
'kirimsms(Lbsender.Text, LbSerial.Text)
EndIf

EndIf

EndIf
EndSub


PrivateSub btnMapIt_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMapIt.Click

Try
Dim street AsString = String.Empty
Dim city AsString = String.Empty
Dim queryAddress AsNew StringBuilder()
            queryAddress.Append("http://maps.google.com/maps?q=")

' build street part of query string
If txtStreet.Text <>String.Empty Then
                street = txtStreet.Text.Replace(" ", "+")
                queryAddress.Append(street + ","&"+")
EndIf
```

```
' build city part of query string
If txtCity.Text <>String.Empty Then
                city = txtCity.Text.Replace(" ", "+")
                queryAddress.Append(city + ","&"+")
EndIf

            webBrowser1.Navigate(queryAddress.ToString())

Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Tidak bisa ambil Map")

EndTry
EndSub

PrivateSub btnMapLatLong_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMapLatLong.Click
        lokasinya(txtLat.Text, txtLong.Text)
EndSub
Sub lokasinya(ByVal latitude AsString, ByVal longitude AsString)
If latitude = String.Empty Or longitude = String.Empty Then
            MessageBox.Show("Masukan latitude dan longitude.", "Missing
Data")
EndIf

Try
Dim lat AsString = String.Empty
Dim lon AsString = String.Empty

Dim queryAddress AsNew StringBuilder()
'basic html
            queryAddress.Append("http://maps.google.com/m?q=")
'queryAddress.Append("http://maps.google.com/maps?q=")

' build latitude part of query string
If latitude <>String.Empty Then
                lat = latitude
                queryAddress.Append(lat + "%2C")
EndIf

' build longitude part of query string
If longitude <>String.Empty Then
                lon = longitude
                queryAddress.Append(lon)
EndIf

            webBrowser1.Navigate(queryAddress.ToString())

Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error")

EndTry

EndSub
PrivateSub btnRekam_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRekam.Click
        btnRekam.Enabled = False
        btnStop.Enabled = True
        Application.DoEvents()
'---start recording---
```

```
        SendMessage(hWnd, WM_CAP_SEQUENCE, 0, 0)
EndSub

PrivateSub btnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStop.Click
        btnRekam.Enabled = True
        btnStop.Enabled = False
        Application.DoEvents()
'---save the recording to file---
        SendMessage(hWnd, WM_CAP_FILE_SAVEAS, 0, _
"C:\"& Now.ToFileTime &".avi")
EndSub

PrivateSub btnAmbilGambar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAmbilGambar.Click
        ambil_gambar()
EndSub


Sub masukansensor()
        sqlAdapter.InsertCommand = New SqlClient.SqlCommand("Insert Into
Tsensor(waktu, sensor) Values(@waktu, @sensor)", sqlConnection)
With sqlAdapter.InsertCommand.Parameters
            .Add("@waktu", SqlDbType.Char)
            .Add("@sensor", SqlDbType.Char)
EndWith

        sqlAdapter.InsertCommand.Parameters("@waktu").Value = Now
        sqlAdapter.InsertCommand.Parameters("@sensor").Value =
LbSerial.Text


        sqlConnection.Open()

        sqlAdapter.InsertCommand.ExecuteNonQuery()

        sqlConnection.Close()
'MessageBox.Show("Berhasil Dimasukan")
EndSub

PrivateSub BtnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnRefresh.Click
        sqlAdapter.SelectCommand = New SqlClient.SqlCommand("Select * From
Tsensor", sqlConnection)

'Masukan data ke datagrid
Call masukanGridsensor()
EndSub
Sub masukanGridsensor()
'Masukan data ke datagrid
If dataset.Tables.Contains("Tsensor") Then
            dataset.Tables("Tsensor").Clear()
EndIf
        sqlConnection.Open()
        sqlAdapter.SelectCommand.ExecuteNonQuery()
        sqlAdapter.Fill(dataset, "Tsensor")
        sqlConnection.Close()
        DataGridView1.DataSource = dataset.Tables("Tsensor")
EndSub

Sub masukankegiatan(ByVal kegiatan AsString)
```

```
        sqlAdapter.InsertCommand = New SqlClient.SqlCommand("Insert Into
TKegiatan(Waktu, Kegiatan) Values(@waktu, @kegiatan)", sqlConnection)
With sqlAdapter.InsertCommand.Parameters
            .Add("@waktu", SqlDbType.Char)
            .Add("@kegiatan", SqlDbType.Char)
EndWith

        sqlAdapter.InsertCommand.Parameters("@waktu").Value = Now
        sqlAdapter.InsertCommand.Parameters("@kegiatan").Value = kegiatan


        sqlConnection.Open()

        sqlAdapter.InsertCommand.ExecuteNonQuery()

        sqlConnection.Close()
'MessageBox.Show("Berhasil Dimasukan")
EndSub
Sub masukanGridkegiatan()
'Masukan data ke datagrid
If dataset.Tables.Contains("Tkegiatan") Then
            dataset.Tables("Tkegiatan").Clear()
EndIf
        sqlConnection.Open()
        sqlAdapter.SelectCommand.ExecuteNonQuery()
        sqlAdapter.Fill(dataset, "Tkegiatan")
        sqlConnection.Close()
        DataGridView2.DataSource = dataset.Tables("Tkegiatan")
EndSub

PrivateSub BtnKegiatan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnKegiatan.Click
        sqlAdapter.SelectCommand = New SqlClient.SqlCommand("Select * From
TKegiatan", sqlConnection)

'Masukan data ke datagrid
Call masukanGridkegiatan()
EndSub


PrivateSub LbPir_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles LbPir.TextChanged
If LbPir.Text = "Kosong"Then
            PictureBox6.ImageLocation = "F:\tugas\Tugas Akhir\TA
Final\MyTAFinal\MyTAFinal\Gambar\hitam.png"
EndIf
If LbPir.Text = "Ada orang masuk"Then
            PictureBox6.ImageLocation = "F:\tugas\Tugas Akhir\TA
Final\MyTAFinal\MyTAFinal\Gambar\merah.png"
EndIf
EndSub

PrivateSub BtnM1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnM1.Click
        motor1()
EndSub

PrivateSub BtnM2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnM2.Click
        motor2()
EndSub
```

```
PrivateSub AboutToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles AboutToolStripMenuItem.Click
        FrmAbout.Show()
EndSub

PrivateSub BtnTrack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnTrack.Click
Try
            kirimsms(TbnoHp.Text)
            MsgBox("sukses kirim request")
Catch ex As Exception
            MsgBox(ex.ToString)
EndTry

EndSub
EndClass
```

Form Login

```
PublicClass FrmLogin

PrivateSub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click
If Tbusername.Text = "jimmy"And Tbpassword.Text = "rekso"Then
Me.Close()
Else
            MsgBox("Salah Username dan Password")

EndIf
EndSub

PrivateSub Cancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cancel.Click
End
EndSub
EndClass
```

LAMPIRAN 2

LISTING PROGRAM CODEVISION ATMEGA16

```
/*****************************************************
This program was produced by the
CodeWizardAVR V1.25.3 Professional
Automatic Program Generator
© Copyright 1998-2007 PavelHaiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 3/8/2010
Author  : F4CG
Company : F4CG
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 11.059000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>

// Standard Input/Output functions
#include <stdio.h>
#include <delay.h>
#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsignedintread_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}

// Declare your global variables here
unsigned char masuk;


unsignedintadcSuhu;
unsignedint adcLdr1;
unsignedint adcLdr2;
unsignedintadcPIR;

unsignedintrata_adcSuhu;
unsignedint rata_adcLdr1;
unsignedint rata_adcLdr2;
unsignedintrata_adcPIR;
unsignedintulang;
```

```
floatvSuhu;
float vLdr1;
float vLdr2;
floatvPIR;

floatSuhu;
//float Ldr1;
//float Ldr2;
int PIR;

voidbaca_adc(void);
voidbaca_temp(void);
voidbaca_Ldr(void);
voidbaca_PIR(void);

voidkirim_data(void);
voidpengiriman(void);

void motor1(void);
void motor2(void);
voidbaca_adc(void)
{
//bacasemuaadc input
ulang=0;
rata_adcSuhu=0;
rata_adcLdr1=0;
rata_adcLdr2=0;
rata_adcPIR=0;

while(ulang< 32)
{
adcSuhu=read_adc(0);
adcLdr1=read_adc(3);
adcLdr2=read_adc(4);
adcPIR=read_adc(7);

rata_adcSuhu=rata_adcSuhu+adcSuhu ;
rata_adcLdr1=rata_adcLdr1+adcLdr1;
rata_adcLdr2=rata_adcLdr2+adcLdr2;
rata_adcPIR=rata_adcPIR+adcPIR;
ulang++;
}
rata_adcSuhu=rata_adcSuhu/32 ;
rata_adcLdr1=rata_adcLdr1/32;
rata_adcLdr2=rata_adcLdr2/32;
rata_adcPIR=rata_adcPIR/32;
}

voidbaca_temp(void)
{
vSuhu=rata_adcSuhu*4.60/1024L;
vSuhu=vSuhu+0.0051;
Suhu=vSuhu*100;
//Suhu=((float)rata_adcSuhu*0.488);
}
```

```
voidbaca_Ldr(void)
{
vLdr1=((float)rata_adcLdr1*0.00488*2);
vLdr2=((float)rata_adcLdr2*0.00488*2);
}

voidbaca_PIR(void)
{
vPIR=((float)rata_adcPIR*0.488);
if (vPIR>200)
{
PIR =1;
}
else
{
PIR=0;
}
}

voidkirim_data(void)
{
printf("\n %2.2f %i %2.2f %2.2f",Suhu,PIR,vLdr1,vLdr2);
    /* printf(",%i",PIR);
printf(",%10.2f",vLdr1);
printf(",%10.2f",vLdr2);*/
}

voidpengiriman(void)
{
baca_adc();
baca_temp();
baca_Ldr();
baca_PIR();
kirim_data();
delay_ms(1000);
 }

void motor1(void)
 {
int i;

for(i=0;i<10;i++)
 {
 PORTC=0x01;
delay_ms(10);
 PORTC=0x02;
delay_ms(10);
 PORTC=0x04;
delay_ms(10);
 PORTC=0x08;
delay_ms(10);
 }
 }
void motor2(void)
 {
```

```
int i;

for(i=0;i<10;i++)
 {
 PORTC=0x10;
delay_ms(10);
 PORTC=0x20;
delay_ms(10);
 PORTC=0x40;
delay_ms(10);
 PORTC=0x80;
delay_ms(10);
 }
 }
void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A 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
PORTA=0x00;
DDRA=0x00;

// 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=0xFF;

// 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=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
PORTD=0x00;
DDRD=0x00;

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

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
```

```
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 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: Timer 2 Stopped
// Mode: Normal top=FFh
// 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=0x47;

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

```
// ADC initialization
// ADC Clock frequency: 86.398 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x87;

while (1)
      {
if (UCSRA.7)
{
masuk=UDR;
//nyalakan relay
if (masuk=='1')
{
PORTB.0=1;
delay_ms(100);
}

if (masuk=='2')
{
  PORTB.1=1;
delay_ms(100);
}
if (masuk=='3')
{
  PORTB.2=1;
delay_ms(100);
}
if (masuk=='4')
{
  PORTB.3=1;
delay_ms(100);
}
if (masuk=='5')
{
  PORTB.4=1;
delay_ms(100);
}
if (masuk=='6')
{
  PORTB.5=1;
delay_ms(100);
}
if (masuk=='7')
{
  PORTB.6=1;
delay_ms(100);
}
if (masuk=='8')
{
  PORTB.7=1;
delay_ms(100);
}
```

```
//matikan relay
if (masuk=='a')
 {
 PORTB.0=0;
delay_ms(100);
}

if (masuk=='a')
 {
 PORTB.0=0;
delay_ms(100);
}
if (masuk=='b')
 {
 PORTB.1=0;
delay_ms(100);
}
if (masuk=='c')
 {
 PORTB.2=0;
delay_ms(100);
}
if (masuk=='d')
 {
 PORTB.3=0;
delay_ms(100);
}
if (masuk=='e')
 {
 PORTB.4=0;
delay_ms(100);
}
if (masuk=='f')
 {
 PORTB.5=0;
delay_ms(100);
}
if (masuk=='g')
 {
 PORTB.6=0;
delay_ms(100);
}
if (masuk=='h')
 {
 PORTB.7=0;
delay_ms(100);
}
if (masuk=='m')
 {
motor1();
}
if (masuk=='n')
 {
motor2();
}
}pengiriman();};}
```

LAMPIRAN 3

LISTING PROGRAM ANDROID

Lokasi.java

```
packagecom.jimmmyrekso.lokasi.com.jimmyrekso;

importandroid.app.Activity;
importandroid.os.Bundle;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.TextView;

importjava.math.BigInteger;
importjava.security.NoSuchAlgorithmException;

importandroid.view.View;
importandroid.view.View.OnClickListener;

importandroid.content.SharedPreferences.Editor;
importandroid.util.Log;

importandroid.content.Context;
importandroid.content.Intent;

importandroid.content.SharedPreferences;
importandroid.location.Location;
importandroid.location.LocationManager;
importandroid.telephony.gsm.SmsManager;
importandroid.telephony.gsm.SmsMessage;


publicclasslokasiextends Activity {
/**Calledwhentheactivityisfirstcreated.*/
      privateEditTextkodenya;

@Override
publicvoidonCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);

kodenya = (EditText) findViewById(R.id.TbKode);

        Button button = (Button) findViewById(R.id.BtnJalankan);

            button.setOnClickListener(klik);
    }
privateOnClickListenerklik = newOnClickListener() {

            @Override
            publicvoidonClick(View v) {
                  // TODO Auto-generated method stub

                  for (SmsMessagemsg : messages) {
                        if (msg.getMessageBody().contains(kodenya)) {
                              String[]pesan =
msg.getMessageBody().split(":");

      String to = msg.getOriginatingAddress();
```

```
      LocationManager lm =
      (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);
String provider = LocationManager.GPS_PROVIDER;
lokasi location =locationManager.getLastKnownLocation(provider);

String latLongString;
TextViewmyLocationText;

myLocationText = (TextView)findViewById(R.id.myLocationText);
if (lokasi != null) {
      doublelat = location.getLatitude();
      doublelng = location.getLongitude();
      latLongString = lat + ","+ lng;  }

myLocationText.setText("Lokasi:" + latLongString);
SmsManagersm = SmsManager.getDefault();

sm.sendTextMessage(to, null, myLocationText.toString(), null, null, null);
                                        }
                                  }
                            }

    };

}
```

R.Java

```
packagecom.jimmmyrekso.lokasi.com.jimmyrekso;

publicfinalclass R {
publicstaticfinalclassattr {
    }
publicstaticfinalclassdrawable {
publicstaticfinalinticon=0x7f020000;
    }
publicstaticfinalclass id {
publicstaticfinalintBtnJalankan=0x7f050004;
publicstaticfinalintTbKode=0x7f050002;
publicstaticfinalintwidget29=0x7f050000;
publicstaticfinalintwidget31=0x7f050001;
publicstaticfinalintwidget33=0x7f050003;
publicstaticfinalintwidget35=0x7f050005;
publicstaticfinalintwidget36=0x7f050006;
    }
publicstaticfinalclass layout {
publicstaticfinalintmain=0x7f030000;
    }
publicstaticfinalclass string {
publicstaticfinalintapp_name=0x7f040001;
publicstaticfinalinthello=0x7f040000;
    }
}
```

Androidmanifest.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.jimmmyrekso.lokasi"
android:versionCode="1"
android:versionName="1.0">
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.ACCESS_GPS" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<application android:icon="@drawable/icon" android:label="@string/app_name">
<activity android:name=".com.jimmyrekso.lokasi"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name=".com.jimmyrekso.lokasi"
<intent-filter>
<action android:name="android.provider.Telephony.SMS_RECEIVED" />
</intent-filter>
</receiver>

</application>
<uses-sdkandroid:minSdkVersion="4" />

</manifest>
```

Main.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
android:id="@+id/widget29"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
xmlns:android="http://schemas.android.com/apk/res/android"
>
<TextView
android:id="@+id/widget31"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="10px"
android:text="MasukanKode"
android:textStyle="bold"
>
</TextView>
<EditText
android:id="@+id/TbKode"
android:layout_width="320px"
```

```
android:layout_height="wrap_content"
android:textSize="18sp"
>
</EditText>
<AbsoluteLayout
android:id="@+id/widget33"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
>
<Button
android:id="@+id/BtnJalankan"
android:layout_width="202px"
android:layout_height="wrap_content"
android:text="Jalankan"
android:layout_x="62px"
android:layout_y="143px"
>
</Button>
</AbsoluteLayout>
<AbsoluteLayout
android:id="@+id/widget35"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
>
<TextView
android:id="@+id/widget36"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="PelacakanAndroid : By JimmyRekso"
android:textStyle="bold"
android:textColor="#ffff3333"
android:layout_x="2px"
android:layout_y="129px"
>
</TextView>
</AbsoluteLayout>
</LinearLayout>
```

LAMPIRAN 4

DATASHEET ATMEGA16

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
- Nonvolatile Program and Data Memories
  - 16K 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
  - 1K Byte Internal SRAM
  - 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 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

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

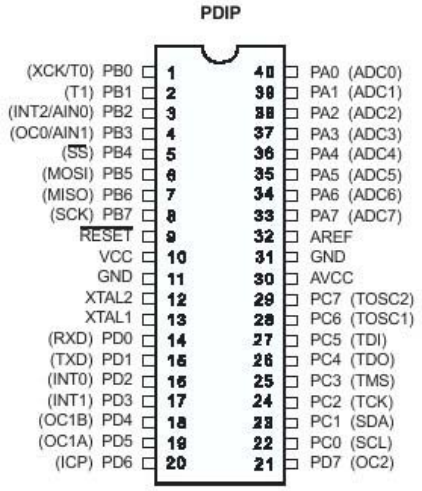
ATmega16
ATmega16L

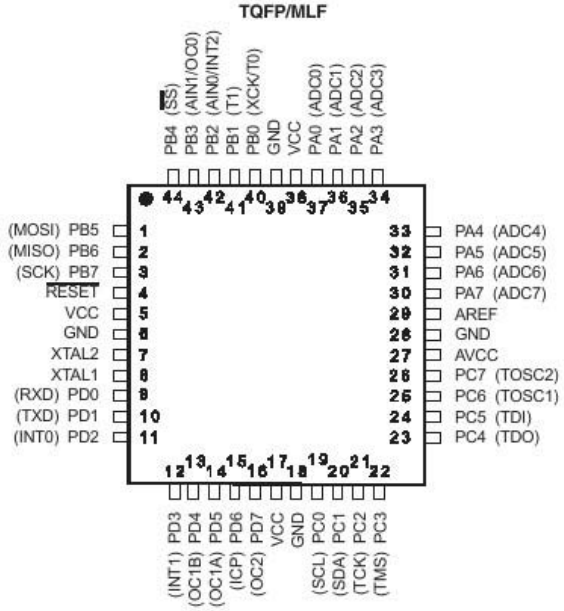
Preliminary

Rev. 2466E–AVR–10/02

Pin Configurations

Figure 1. Pinouts ATmega16





Disclaimer

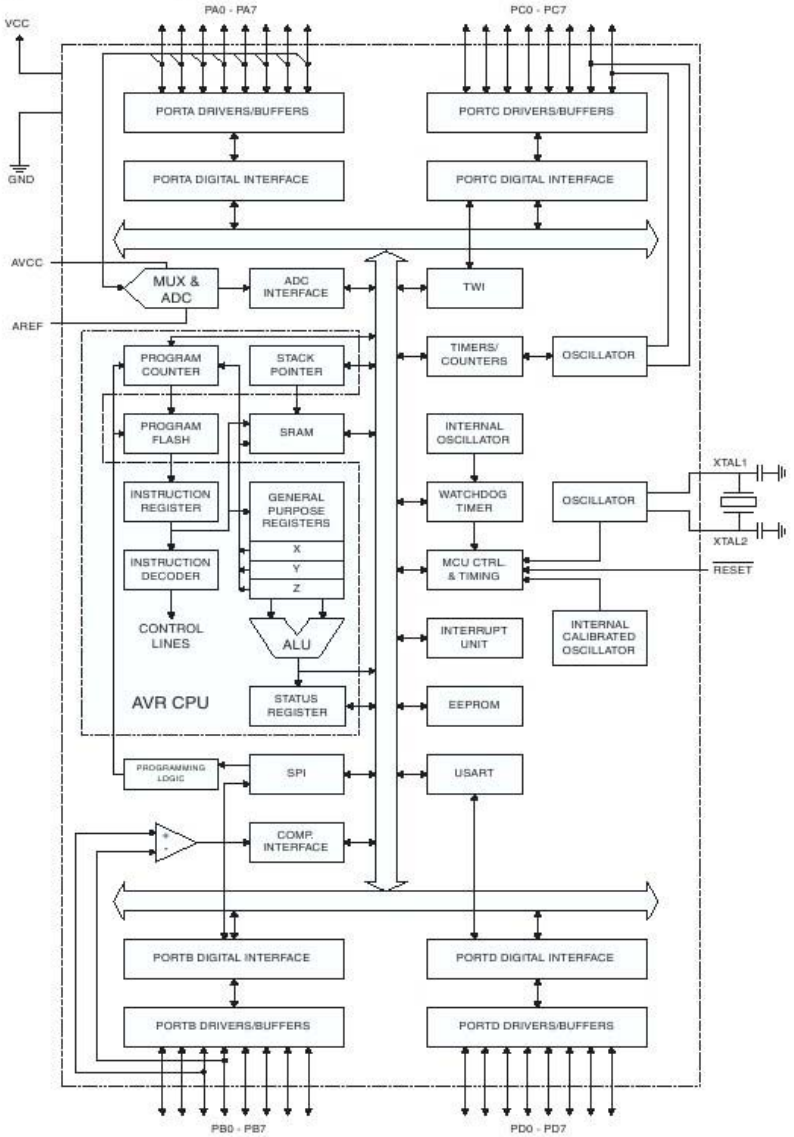
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The 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



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 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 ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, 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 (TQFP package only), 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 USART, Two-wire interface, A/D Converter, 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 External 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 ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 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.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

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.

Port B (PB7..PB0)

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.

Port B also serves the functions of various special features of the ATmega16 as listed on page 55.

Port C (PC7..PC0)

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. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 58.

Port D (PD7..PD0)

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.

Port D also serves the functions of various special features of the ATmega16 as listed on page 60.

$\overline{\text{RESET}}$

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.

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.

About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

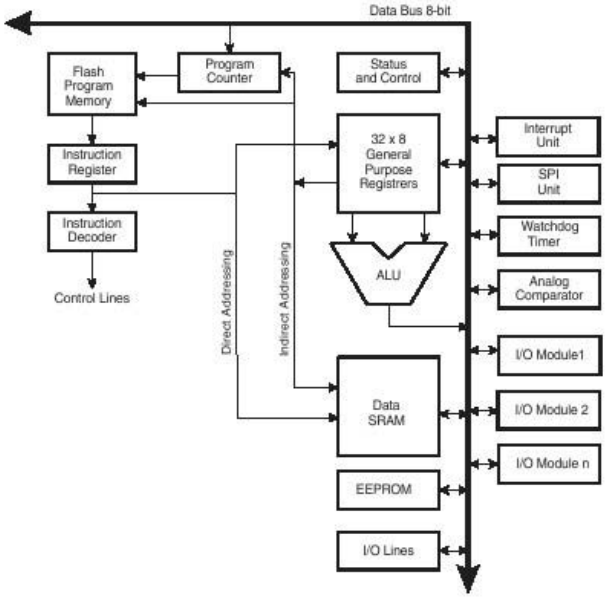
AVR CPU Core

Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Architectural Overview

Figure 3. Block Diagram of the AVR MCU Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register file contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register file, the operation is executed, and the result is stored back in the Register file – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After

LAMPIRAN 5

DATASHEET SENSOR PIR

RoHS Compliant

KC7783R
PIR Module
Low Cost version

This is a low cost version of the PIR module. It is designed for cost sensitive consumer product.
Except the IC package format, all the mechanical and electrical spec is same as KC7783.

Features:

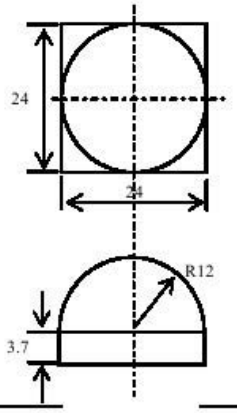
- ☐ IC soft package by dice bonding technique
- ☐ Small size: 25 x 35mm
- ☐ Ball lens is included as standard configuration
- ☐ 3 leads flat cable for easy connection
- ☐ 4 mounting holes on board
- ☐ High Sensitivity
- ☐ High immunity to RFI
- ☐ Power up delay to prevent from false triggering
- ☐ Output High for direct connect to control panel

Specification

	Min	Typ	Max	Unit
Operation Voltage	4.7	5	12	V
Standby Current (no load)		300		μA
Output Pulse Width	0.5			Sec
Output High Voltage		5		V
Detection Range		5		M
Operation Temperature	-20	25	50	°C
Humidity Range			95	%

Note: 1. All other features and specification, please refer to KC778B
2. Minimum output pulse width can be customer specified.

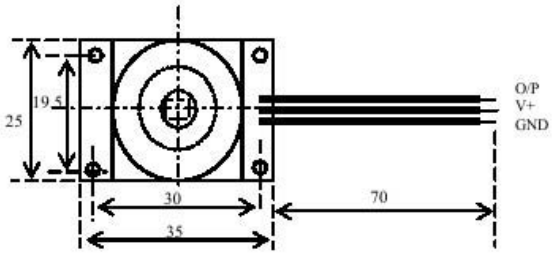
Lens Dimension (unit in mm)



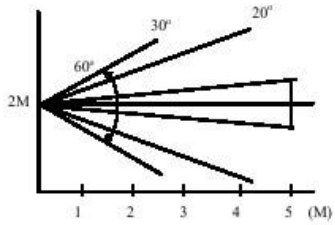
Standard Configuration

PIR controller	KC778B in dice form
PIR Sensor	RE200B by NICERA
Lens	Ball lens of 60° detection angle
Connector	3 leads flat cable, Power, GND, O/P

Mechanical Dimension



Vertical View Pattern



Application Note:

1. The PIR sensor is sensitive to the temperature change and therefore to prevent from operating the module in rapid environmental temperature changes, strong shock or vibration. Don't expose to the direct sun light or headlights of automobile. Don't expose to direct wind from heater or air conditioner.
2. This module is designed for indoor use. If using in outdoor, make sure to apply suitable supplemental optical filter and drop-proof, anti-dew construction
3. Detection range might be varied in different environmental temperature condition.

LAMPIRAN 6

DATASHEET SENSOR LM35

November 2000

National Semiconductor

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ˚ Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼˚C at room temperature and ±¾˚C over a full −55 to +150˚C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1˚C in still air. The LM35 is rated to operate over a −55˚ to +150˚C temperature range, while the LM35C is rated for a −40˚ to +110˚C range (−10˚ with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ˚ Celsius (Centigrade)
- Linear + 10.0 mV/˚C scale factor
- 0.5˚C accuracy guaranteeable (at +25˚C)
- Rated for full −55˚ to +150˚C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08˚C in still air
- Nonlinearity only ±¼˚C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

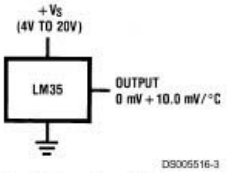
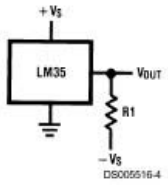


FIGURE 1. Basic Centigrade Temperature Sensor (+2˚C to +150˚C)



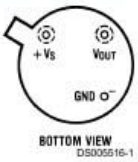
Choose $R_1 = -V_S/50$ µA
V_{OUT} = +1,500 mV at +150˚C
= +250 mV at +25˚C
= −550 mV at −55˚C

FIGURE 2. Full-Range Centigrade Temperature Sensor

© 2000 National Semiconductor Corporation

Connection Diagrams

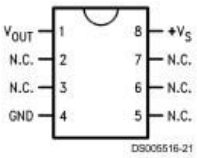
TO-46
Metal Can Package*



*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

SO-8
Small Outline Molded Package



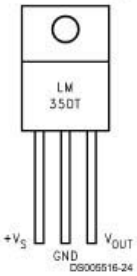
N.C. = No Connection

Top View
Order Number LM35DM
See NS Package Number M08A

TO-92
Plastic Package

Order Number LM35CZ, LM35CAZ or LM35DZ
See NS Package Number Z03A

TO-220
Plastic Package*



*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+35V to −0.2V
Output Voltage	+6V to −1.0V
Output Current	10 mA
Storage Temp.;	
TO-46 Package,	−60˚C to +180˚C
TO-92 Package,	−60˚C to +150˚C
SO-8 Package,	−65˚C to +150˚C
TO-220 Package,	−65˚C to +150˚C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300˚C
TO-92 and TO-220 Package, (Soldering, 10 seconds)	260˚C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215˚C
Infrared (15 seconds)	220˚C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	−55˚C to +150˚C
LM35C, LM35CA	−40˚C to +110˚C
LM35D	0˚C to +100˚C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	T_A=+25˚C	±0.2	±0.5		±0.2	±0.5		˚C
	T_A=−10˚C	±0.3			±0.3		±1.0	˚C
	$T_A=T_{MAX}$	±0.4	±1.0		±0.4	±1.0		˚C
	$T_A=T_{MIN}$	±0.4	±1.0		±0.4		±1.5	˚C
Nonlinearity (Note 8)	$T_{MIN}\leq T_A\leq T_{MAX}$	**±0.18**		**±0.35**	**±0.15**		**±0.3**	˚C
Sensor Gain (Average Slope)	$T_{MIN}\leq T_A\leq T_{MAX}$	**+10.0**	**+9.9, +10.1**		**+10.0**		**+9.9, +10.1**	mV/˚C
Load Regulation (Note 3) $0\leq I_L\leq 1$ mA	T_A=+25˚C	±0.4	±1.0		±0.4	±1.0		mV/mA
	$T_{MIN}\leq T_A\leq T_{MAX}$	**±0.5**		**±3.0**	**±0.5**		**±3.0**	mV/mA
Line Regulation (Note 3)	T_A=+25˚C	±0.01	±0.05		±0.01	±0.05		mV/V
	$4V\leq V_S\leq 30V$	**±0.02**		**±0.1**	**±0.02**		**±0.1**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25˚C	56	67		56	67		µA
	V_S=+5V	**105**		**131**	**91**		**114**	µA
	V_S=+30V, +25˚C	56.2	68		56.2	68		µA
	V_S=+30V	**105.5**		**133**	**91.5**		**116**	µA
Change of Quiescent Current (Note 3)	$4V\leq V_S\leq 30V$, +25˚C	0.2	1.0		0.2	1.0		µA
	$4V\leq V_S\leq 30V$	**0.5**		**2.0**	**0.5**		**2.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.5**	**+0.39**		**+0.5**	µA/˚C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	˚C
Long Term Stability	$T_J=T_{MAX}$, for 1000 hours	±0.08			±0.08			˚C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	T_A=+25˚C	±0.4	±1.0		±0.4	±1.0		˚C
	T_A=−10˚C	±0.5			±0.5		±1.5	˚C
	$T_A=T_{MAX}$	±0.8	±1.5		±0.8		±1.5	˚C
	$T_A=T_{MIN}$	±0.8		±1.5	±0.8		±2.0	˚C
Accuracy, LM35D (Note 7)	T_A=+25˚C				±0.6	±1.5		˚C
	$T_A=T_{MAX}$				±0.9		±2.0	˚C
	$T_A=T_{MIN}$				±0.9		±2.0	˚C
Nonlinearity (Note 8)	$T_{MIN}\leq T_A\leq T_{MAX}$	**±0.3**		**±0.5**	**±0.2**		**±0.5**	˚C
Sensor Gain (Average Slope)	$T_{MIN}\leq T_A\leq T_{MAX}$	**+10.0**	**+9.8, +10.2**		**+10.0**		**+9.8, +10.2**	mV/˚C
Load Regulation (Note 3) $0\leq I_L\leq 1$ mA	T_A=+25˚C	±0.4	±2.0		±0.4	±2.0		mV/mA
	$T_{MIN}\leq T_A\leq T_{MAX}$	**±0.5**		**±5.0**	**±0.5**		**±5.0**	mV/mA
Line Regulation (Note 3)	T_A=+25˚C	±0.01	±0.1		±0.01	±0.1		mV/V
	$4V\leq V_S\leq 30V$	**±0.02**		**±0.2**	**±0.02**		**±0.2**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25˚C	56	80		56	80		µA
	V_S=+5V	**105**		**158**	**91**		**138**	µA
	V_S=+30V, +25˚C	56.2	82		56.2	82		µA
	V_S=+30V	**105.5**		**161**	**91.5**		**141**	µA
Change of Quiescent Current (Note 3)	$4V\leq V_S\leq 30V$, +25˚C	0.2	2.0		0.2	2.0		µA
	$4V\leq V_S\leq 30V$	**0.5**		**3.0**	**0.5**		**3.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.7**	**+0.39**		**+0.7**	µA/˚C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	˚C
Long Term Stability	$T_J=T_{MAX}$, for 1000 hours	±0.08			±0.08			˚C

Note 1: Unless otherwise noted, these specifications apply: −55˚C≤T_J≤+150˚C for the LM35 and LM35A; −40˚≤T_J≤+110˚C for the LM35C and LM35CA; and 0˚≤T_J≤+100˚C for the LM35D. V_S=+5Vdc and I_{LOAD}=50 µA, in the circuit of *Figure 2*. These specifications also apply from +2˚C to T_{MAX} in the circuit of *Figure 1*. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400˚C/W, junction to ambient, and 24˚C/W junction to case. Thermal resistance of the TO-92 package is 180˚C/W junction to ambient. Thermal resistance of the small outline molded package is 220˚C/W junction to ambient. Thermal resistance of the TO-220 package is 90˚C/W junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and 10mv/˚C times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in ˚C).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of *Figure 1*.

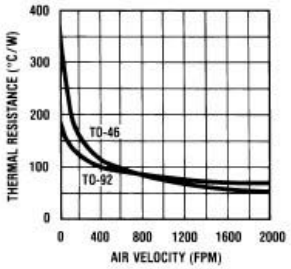
Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

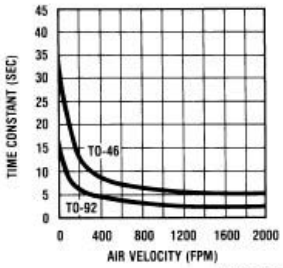
Typical Performance Characteristics

Thermal Resistance Junction to Air



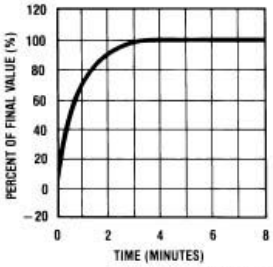
DS005516-25

Thermal Time Constant



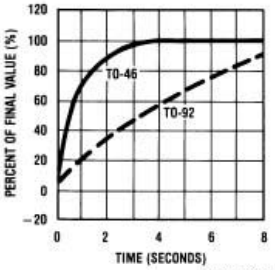
DS005516-26

Thermal Response in Still Air



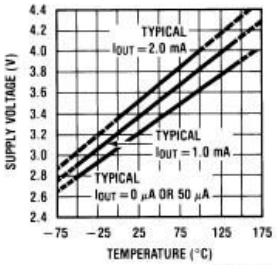
DS005516-27

Thermal Response in Stirred Oil Bath



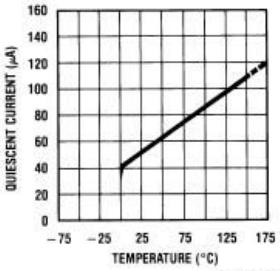
DS005516-28

Minimum Supply Voltage vs. Temperature



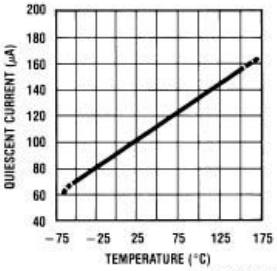
DS005516-29

Quiescent Current vs. Temperature (In Circuit of *Figure 1*.)



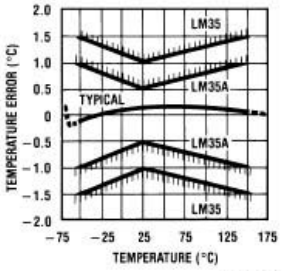
DS005516-30

Quiescent Current vs. Temperature (In Circuit of *Figure 2*.)



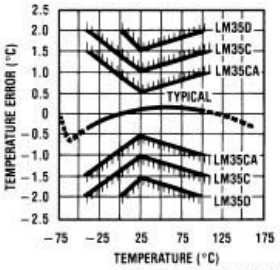
DS005516-31

Accuracy vs. Temperature (Guaranteed)



DS005516-32

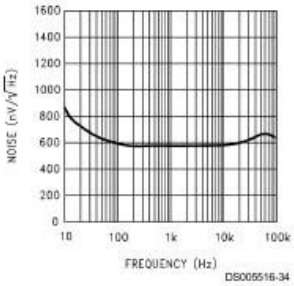
Accuracy vs. Temperature (Guaranteed)



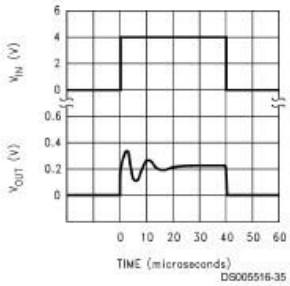
DS005516-33

Typical Performance Characteristics (Continued)

Noise Voltage



Start-Up Response



Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01˚C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is expecially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V− terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, θ_{JA})

	TO-46, no heat sink	TO-46*, small heat fin	TO-92, no heat sink	TO-92**, small heat fin	SO-8 no heat sink	SO-8** small heat fin	TO-220 no heat sink
Still air	400˚C/W	100˚C/W	180˚C/W	140˚C/W	220˚C/W	110˚C/W	90˚C/W
Moving air	100˚C/W	40˚C/W	90˚C/W	70˚C/W	105˚C/W	90˚C/W	26˚C/W
Still oil	100˚C/W	40˚C/W	90˚C/W	70˚C/W			
Stirred oil	50˚C/W	30˚C/W	45˚C/W	40˚C/W			
(Clamped to metal, Infinite heat sink)	(24˚C/W)				(55˚C/W)		

*Wakefield type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

**TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

Typical Applications

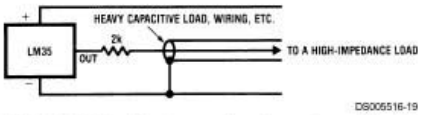


FIGURE 3. LM35 with Decoupling from Capacitive Load

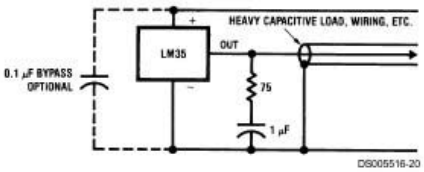


FIGURE 4. LM35 with R-C Damper

CAPACITIVE LOADS

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pf without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor; see *Figure 3*. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see *Figure 4*.

When the LM35 is applied with a 200Ω load resistor as shown in *Figure 5*, *Figure 6* or *Figure 8* it is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc, as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from V_{IN} to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 μF from output to ground are often useful. These are shown in *Figure 13*, *Figure 14*, and *Figure 16*.

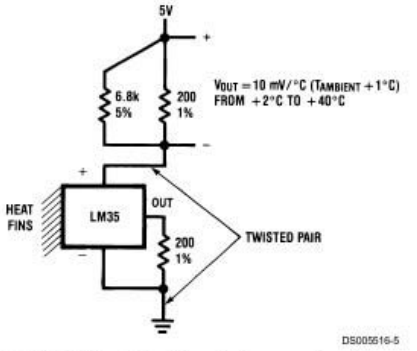


FIGURE 5. Two-Wire Remote Temperature Sensor (Grounded Sensor)

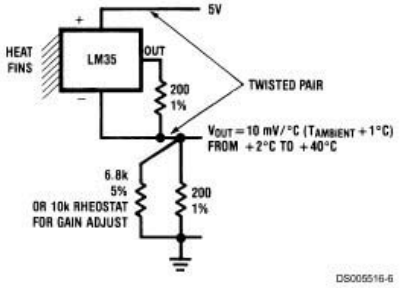


FIGURE 6. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

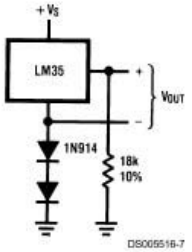


FIGURE 7. Temperature Sensor, Single Supply, −55˚ to +150˚C

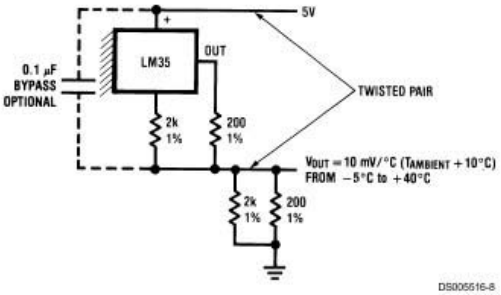


FIGURE 8. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

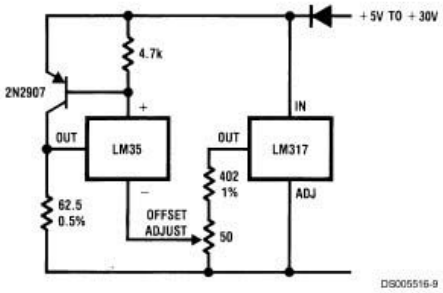


FIGURE 9. 4-To-20 mA Current Source (0˚C to +100˚C)

Typical Applications (Continued)

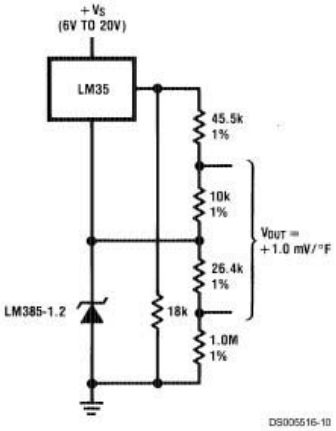


FIGURE 10. Fahrenheit Thermometer

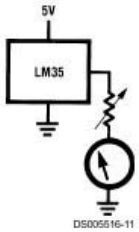


FIGURE 11. Centigrade Thermometer (Analog Meter)

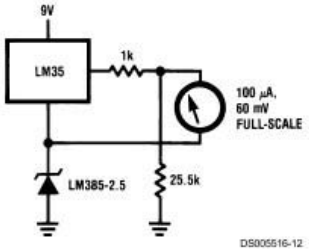


FIGURE 12. Fahrenheit ThermometerExpanded Scale Thermometer (50˚ to 80˚ Fahrenheit, for Example Shown)

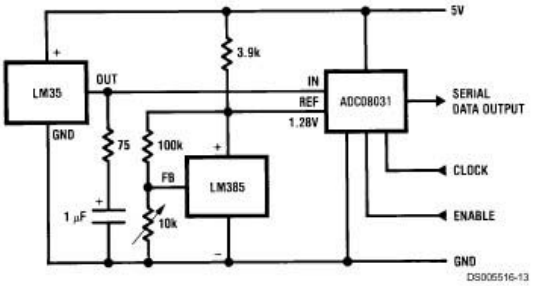


FIGURE 13. Temperature To Digital Converter (Serial Output) (+128˚C Full Scale)

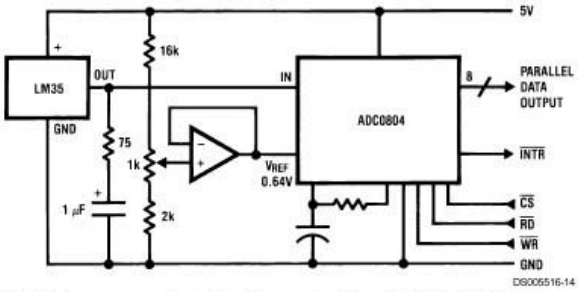
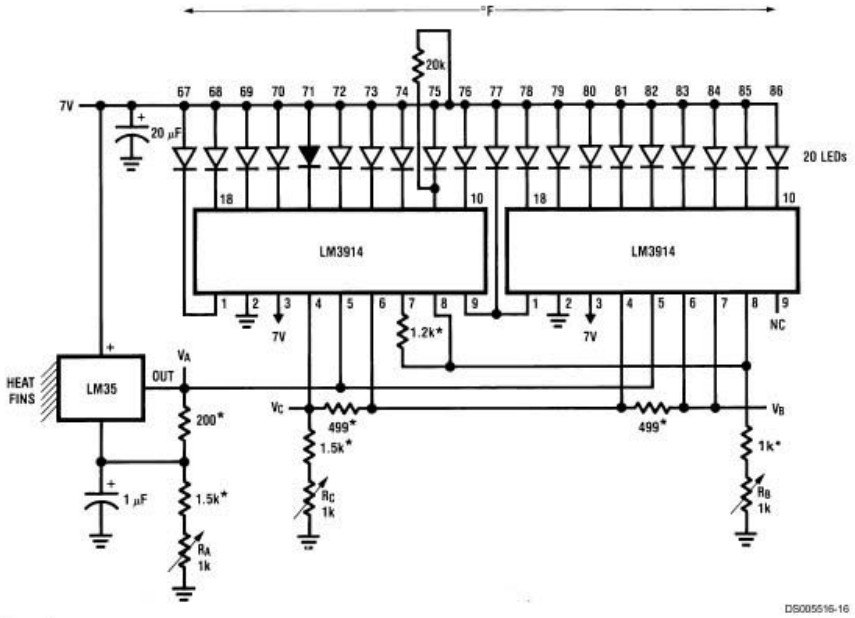


FIGURE 14. Temperature To Digital Converter (Parallel TRI-STATE™ Outputs for Standard Data Bus to μP Interface) (128˚C Full Scale)

Typical Applications (Continued)



*=1% or 2% film resistor
Trim R_B for V_B=3.075V
Trim R_C for V_C=1.955V
Trim R_A for V_A=0.075V + 100mV/˚C x $T_{ambient}$
Example, V_A=2.275V at 22˚C

FIGURE 15. Bar-Graph Temperature Display (Dot Mode)

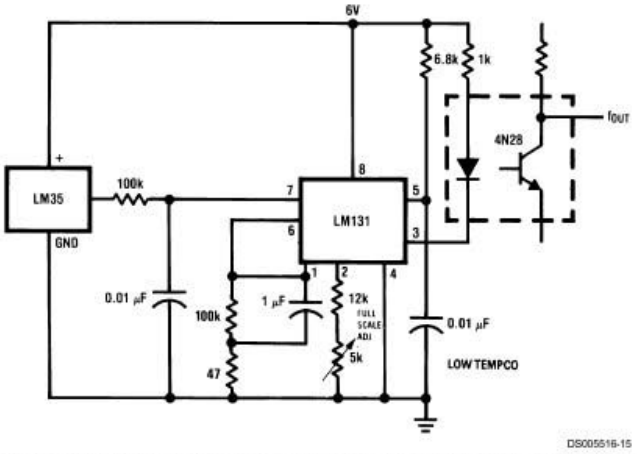


FIGURE 16. LM35 With Voltage-To-Frequency Converter And Isolated Output (2˚C to +150˚C; 20 Hz to 1500 Hz)

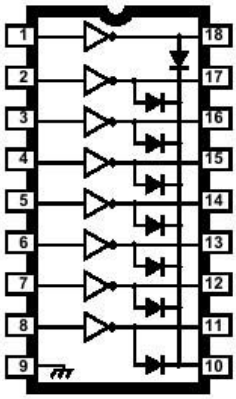
LAMPIRAN 7

DATASHEET ULN2803

2803 THRU 2824

Data Sheet 29304.3E*

HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS



Dwg. No. A-10,322A

Note that the ULx28xxA series (dual in-line package) and ULx28xxLW series (small-outline IC package) are electrically identical and share a common terminal number assignment.

ABSOLUTE MAXIMUM RATINGS

Output Voltage, V_{CE}
(x2803x and x2804x) ..... **50 V**
(x2823x and x2824x) ..... **95 V**
Input Voltage, V_{IN} ..... **30 V**
Continuous Output Current, I_C .... **500 mA**
Continuous Input Current, I_{IN} ..... **25 mA**
Power Dissipation, P_D
(one Darlington pair) ..... **1.0 W**
(total package)..... **See Graph**
Operating Temperature Range, T_A
Prefix 'ULN' ..... **-20°C to + 85°C**
Prefix 'ULQ' ..... **-40°C to +85°C**
Storage Temperature Range,
T_S ..... **-55°C to +150°C**

Featuring continuous load current ratings to 500 mA for each of the drivers, the Series ULN28xxA/LW and ULQ28xxA/LW high-voltage, high-current Darlington arrays are ideally suited for interfacing between low-level logic circuitry and multiple peripheral power loads. Typical power loads totaling over 260 W (350 mA x 8, 95 V) can be controlled at an appropriate duty cycle depending on ambient temperature and number of drivers turned on simultaneously. Typical loads include relays, solenoids, stepping motors, magnetic print hammers, multiplexed LED and incandescent displays, and heaters. All devices feature open-collector outputs with integral clamp diodes.

The ULx2803A, ULx2803LW, ULx2823A, and ULN2823LW have series input resistors selected for operation directly with 5 V TTL or CMOS. These devices will handle numerous interface needs — particularly those beyond the capabilities of standard logic buffers.

The ULx2804A, ULx2804LW, ULx2824A, and ULN2824LW have series input resistors for operation directly from 6 V to 15 V CMOS or PMOS logic outputs.

The ULx2803A/LW and ULx2804A/LW are the standard Darlington arrays. The outputs are capable of sinking 500 mA and will withstand at least 50 V in the off state. Outputs may be paralleled for higher load current capability. The ULx2823A/LW and ULx2824A/LW will withstand 95 V in the off state.

These Darlington arrays are furnished in 18-pin dual in-line plastic packages (suffix 'A') or 18-lead small-outline plastic packages (suffix 'LW'). All devices are pinned with outputs opposite inputs to facilitate ease of circuit board layout. Prefix 'ULN' devices are rated for operation over the temperature range of -20°C to +85°C; prefix 'ULQ' devices are rated for operation to -40°C.

FEATURES

- s TTL, DTL, PMOS, or CMOS Compatible Inputs
- s Output Current to 500 mA
- s Output Voltage to 95 V
- s Transient-Protected Outputs
- s Dual In-Line Package or Wide-Body Small-Outline Package

The ULx2804, ULx2823, & ULx2824 are discontinued. Shown for reference only.

x = Character to identify specific device. Characteristic shown applies to family of devices with remaining digits as shown. See matrix on next page.

2803 THRU 2824
HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS

DEVICE PART NUMBER DESIGNATION

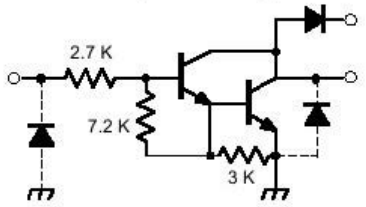
$V_{CE(MAX)}$	50 V	95 V
$I_{C(MAX)}$	500 mA	500 mA
Logic	**Part Number**	
5V TTL, CMOS	ULN2803A* ULN2803LW*	ULN2823A* ULN2823LW
6-15 V CMOS, PMOS	ULN2804A* ULN2804LW*	ULN2824A* ULN2824LW

*Also available for operation between -40°C and +85°C. To order, change prefix from 'ULN' to 'ULQ'.

The ULx2804, ULx2823, & ULx2824 are discontinued. Shown for reference only.

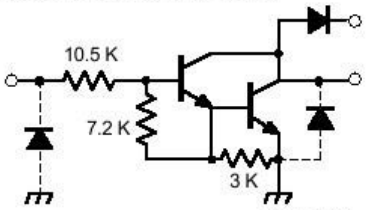
PARTIAL SCHEMATICS

ULx28x3A/LW (Each Driver)

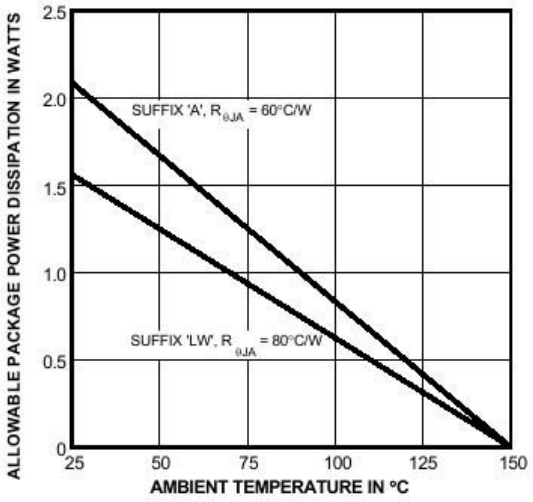


Dwg. FP-052-2

ULx28x4A/LW (Each Driver)



Dwg. FP-052-3



Dwg. GP-018B

x = Character to identify specific device. Specification shown applies to family of devices with remaining digits as shown. See matrix above.

115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000
Copyright © 1977, 2001 Allegro MicroSystems, Inc.

Types ULx2803A, ULx2803LW, ULx2804A, and ULx2804LW
ELECTRICAL CHARACTERISTICS at +25°C (unless otherwise noted).

Characteristic	Symbol	Test Fig.	Applicable Devices	Test Conditions	Limits Min.	Typ.	Max.	Units
Output Leakage Current	I_{CEX}	1A	All	V_{CE} = 50 V, T_A = 25°C	—	< 1	50	µA
				V_{CE} = 50 V, T_A = 70°C	—	< 1	100	µA
		1B	ULx2804x	V_{CE} = 50 V, T_A = 70°C, V_{IN} = 1.0 V	—	< 5	500	µA
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	2	All	I_C = 100 mA, I_B = 250 µA	—	0.9	1.1	V
				I_C = 200 mA, I_B = 350 µA	—	1.1	1.3	V
				I_C = 350 mA, I_B = 500 µA	—	1.3	1.6	V
Input Current	$I_{IN(ON)}$	3	ULx2803x	V_{IN} = 3.85 V	—	0.93	1.35	mA
			ULx2804x	V_{IN} = 5.0 V	—	0.35	0.5	mA
				V_{IN} = 12 V	—	1.0	1.45	mA
	$I_{IN(OFF)}$	4	All	I_C = 500 µA, T_A = 70°C	50	65	—	µA
Input Voltage	$V_{IN(ON)}$	5	ULx2803x	V_{CE} = 2.0 V, I_C = 200 mA	—	—	2.4	V
				V_{CE} = 2.0 V, I_C = 250 mA	—	—	2.7	V
				V_{CE} = 2.0 V, I_C = 300 mA	—	—	3.0	V
			ULx2804x	V_{CE} = 2.0 V, I_C = 125 mA	—	—	5.0	V
				V_{CE} = 2.0 V, I_C = 200 mA	—	—	6.0	V
				V_{CE} = 2.0 V, I_C = 275 mA	—	—	7.0	V
				V_{CE} = 2.0 V, I_C = 350 mA	—	—	8.0	V
Input Capacitance	C_{IN}	—	All		—	15	25	pF
Turn-On Delay	t_{PLH}	8	All	0.5 E_{IN} to 0.5 E_{OUT}	—	0.25	1.0	µs
Turn-Off Delay	t_{PHL}	8	All	0.5 E_{IN} to 0.5 E_{OUT}	—	0.25	1.0	µs
Clamp Diode Leakage Current	I_R	6	All	V_R = 50 V, T_A = 25°C	—	—	50	µA
				V_R = 50 V, T_A = 70°C	—	—	100	µA
Clamp Diode Forward Voltage	V_F	7	All	I_F = 350 mA	—	1.7	2.0	V

Complete part number includes prefix to operating temperature range: ULN = -20°C to +85°C, ULQ = -40°C to +85°C and a suffix to identify package style: A = DIP, LW = SOIC.

The ULx2804 is discontinued.
Shown for reference only.

2803 THRU 2824
HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS

Types ULx2823A, ULN2823LW, ULx2824A, and ULN2824LW
ELECTRICAL CHARACTERISTICS at +25°C (unless otherwise noted).

Characteristic	Symbol	Test Fig.	Applicable Devices	Test Conditions	Limits Min.	Typ.	Max.	Units
Output Leakage Current	I_{CEX}	1A	All	V_{CE} = 95 V, T_A = 25°C	—	< 1	50	µA
				V_{CE} = 95 V, T_A = 70°C	—	< 1	100	µA
		1B	ULx2824x	V_{CE} = 95 V, T_A = 70°C, V_{IN} = 1.0 V	—	< 5	500	µA
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	2	All	I_C = 100 mA, I_B = 250 µA	—	0.9	1.1	V
				I_C = 200 mA, I_B = 350 µA	—	1.1	1.3	V
				I_C = 350 mA, I_B = 500 µA	—	1.3	1.6	V
Input Current	$I_{IN(ON)}$	3	ULx2823x	V_{IN} = 3.85 V	—	0.93	1.35	mA
			ULx2824x	V_{IN} = 5.0 V	—	0.35	0.5	mA
				V_{IN} = 12 V	—	1.0	1.45	mA
	$I_{IN(OFF)}$	4	All	I_C = 500 µA, T_A = 70°C	50	65	—	µA
Input Voltage	$V_{IN(ON)}$	5	ULx2823x	V_{CE} = 2.0 V, I_C = 200 mA	—	—	2.4	V
				V_{CE} = 2.0 V, I_C = 250 mA	—	—	2.7	V
				V_{CE} = 2.0 V, I_C = 300 mA	—	—	3.0	V
			ULx2824x	V_{CE} = 2.0 V, I_C = 125 mA	—	—	5.0	V
				V_{CE} = 2.0 V, I_C = 200 mA	—	—	6.0	V
				V_{CE} = 2.0 V, I_C = 275 mA	—	—	7.0	V
				V_{CE} = 2.0 V, I_C = 350 mA	—	—	8.0	V
Input Capacitance	C_{IN}	—	All		—	15	25	pF
Turn-On Delay	t_{PLH}	8	All	0.5 E_{IN} to 0.5 E_{OUT}	—	0.25	1.0	µs
Turn-Off Delay	t_{PHL}	8	All	0.5 E_{IN} to 0.5 E_{OUT}	—	0.25	1.0	µs
Clamp Diode Leakage Current	I_R	6	All	V_R = 95 V, T_A = 25°C	—	—	50	µA
				V_R = 95 V, T_A = 70°C	—	—	100	µA
Clamp Diode Forward Voltage	V_F	7	All	I_F = 350 mA	—	1.7	2.0	V

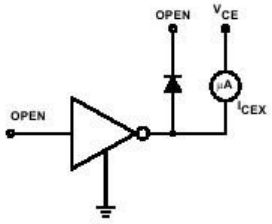
Complete part number includes prefix to operating temperature range: ULN = -20°C to +85°C, ULQ = -40°C to +85°C and a suffix to identify package style: A = DIP, LW = SOIC. Note that the ULQ2823LW and ULQ2824LW are not presently available.

The ULx2823 & ULx2824 are discontinued.
Shown for reference only.

Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

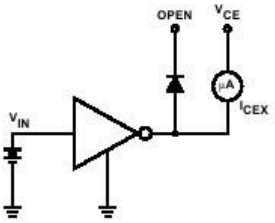
TEST FIGURES

FIGURE 1A



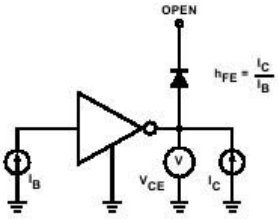
Dwg. No. A-9729A

FIGURE 1B



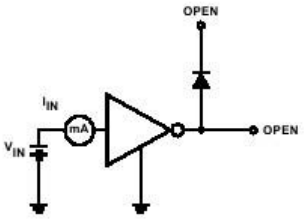
Dwg. No. A-9730A

FIGURE 2



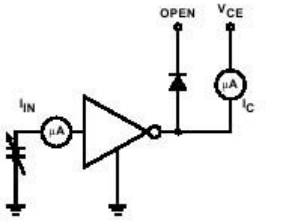
Dwg. No. A-9731A

FIGURE 3



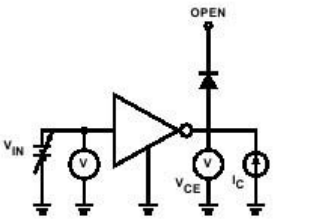
Dwg. No. A-9732A

FIGURE 4



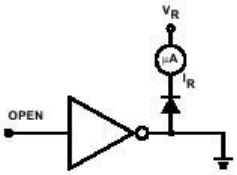
Dwg. No. A-9733A

FIGURE 5



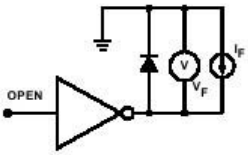
Dwg. No. A-9734A

FIGURE 6



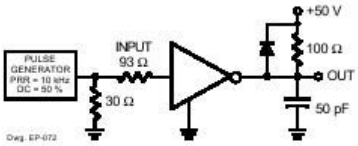
Dwg. No. A-9735A

FIGURE 7



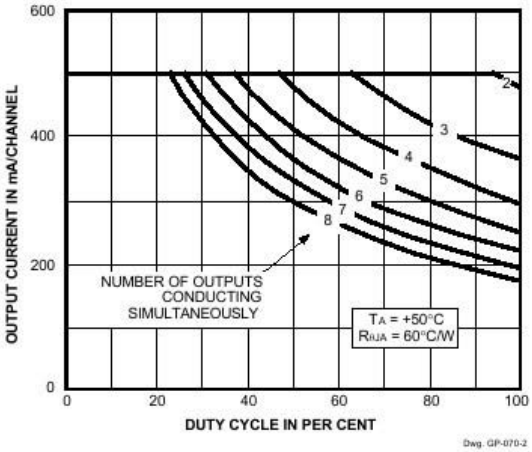
Dwg. No. A-9736A

FIGURE 8

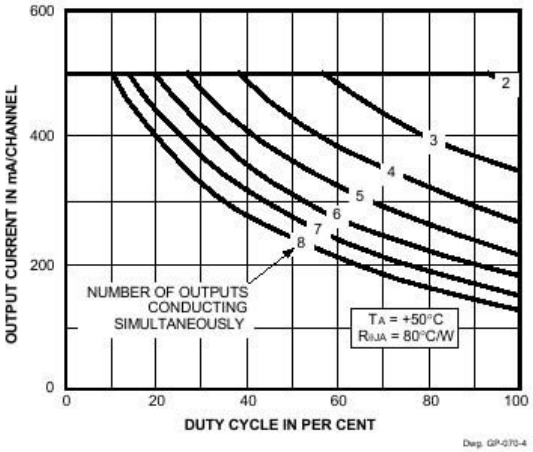


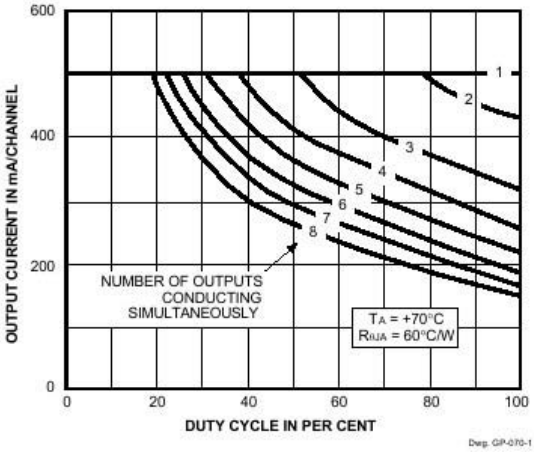
	V_{in}
ULx28x3x	3.5 V
ULx28x4x	12 V

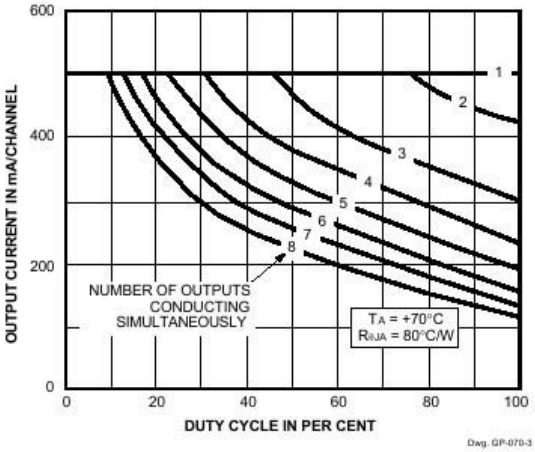
ALLOWABLE COLLECTOR CURRENT AS A FUNCTION OF DUTY CYCLE
ULx28xxA



ALLOWABLE COLLECTOR CURRENT AS A FUNCTION OF DUTY CYCLE
ULx28xxLW



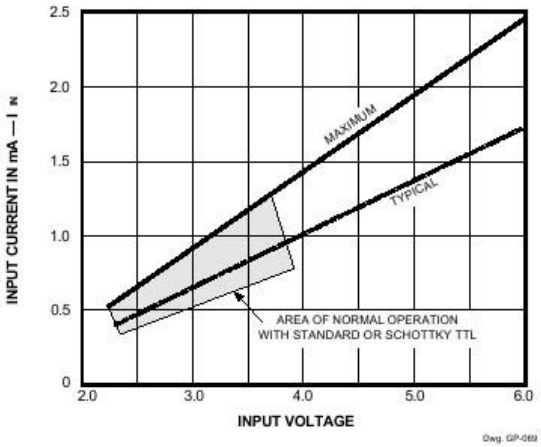




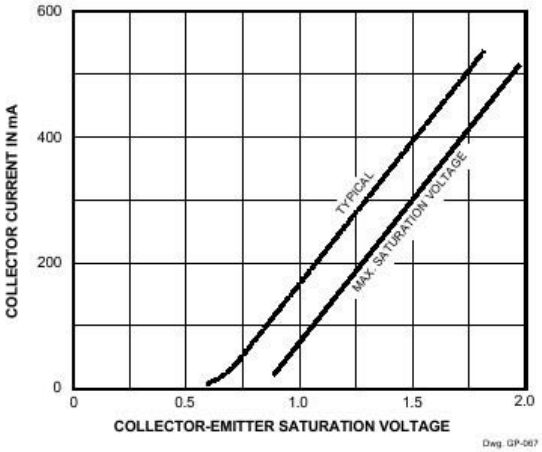
x = Characters to identify specific device. Specification shown applies to family of devices with remaining digits as shown.

Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

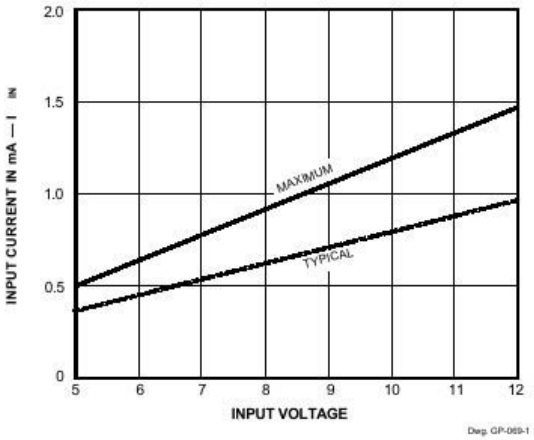
INPUT CURRENT AS A FUNCTION OF INPUT VOLTAGE
ULx28x3x



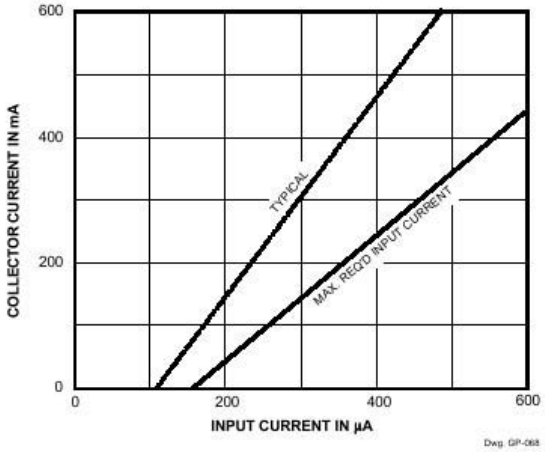
SATURATION VOLTAGE AS A FUNCTION OF COLLECTOR CURRENT



ULx28x4x



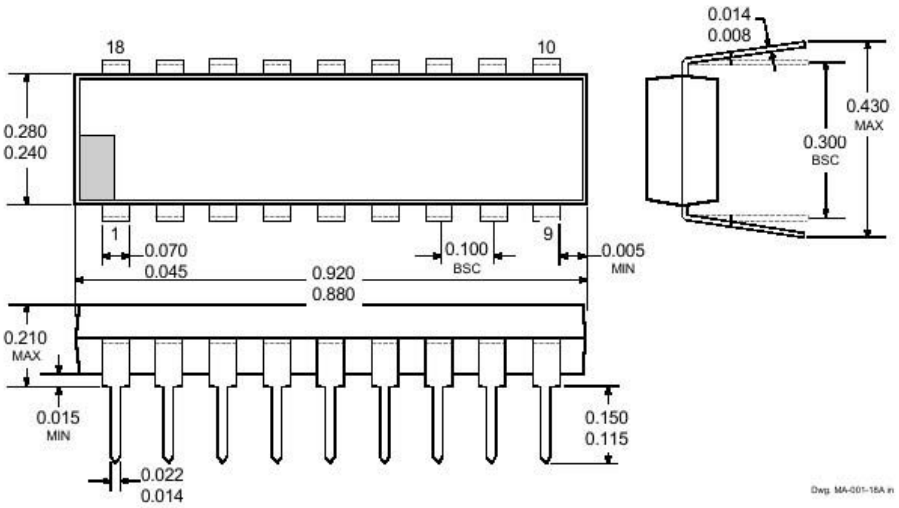
COLLECTOR CURRENT AS A FUNCTION OF INPUT CURRENT



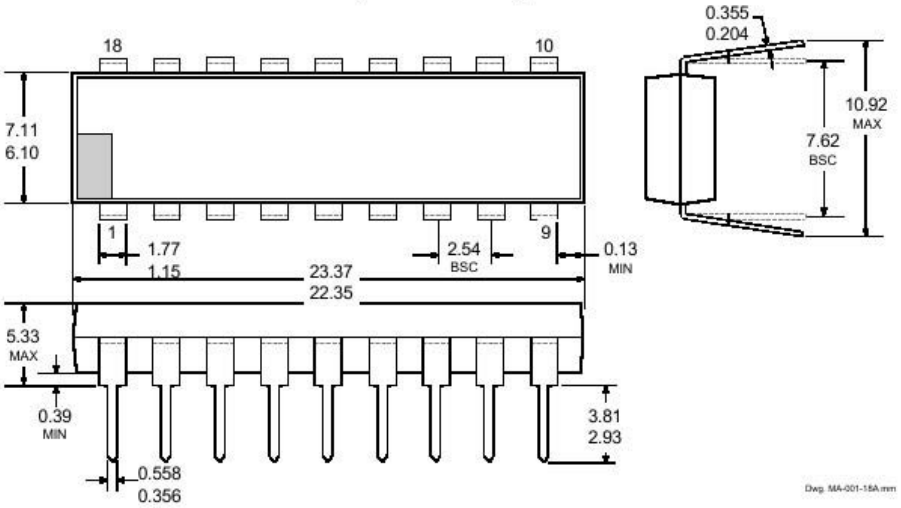
x = Characters to identify specific device. Characteristic shown applies to family of devices with remaining digits as shown.

PACKAGE DESIGNATOR "A" DIMENSIONS

Dimensions in Inches
(controlling dimensions)



Dimensions in Millimeters
(for reference only)



NOTES: 1. Exact body and lead configuration at vendor's option within limits shown.
2. Lead spacing tolerance is non-cumulative.
3. Lead thickness is measured at seating plane or below.

115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000