

```
' (C) 2007 Thomas Tingsted Mathiesen (www.tma.dk/gps)
' This code is freeware.
' Compile, use at your own risk - NO WARRENTY OF ANY KIND!
' Support: tma@tma.dk, subject="gps api"
' Requires Microsoft .NET Framework 2.0
' Developed in Visual Studio Pro 2005

Public Class frmMain
    Private WithEvents gps As New tmagpsapi.NMEA
    Private WithEvents sp As New tmagpsapi.tmaSerialport
    Private Delegate Sub dlgData(ByVal strData As String)
    Private Delegate Sub dlgGPS(ByVal strData As String)
    Private Compas As Bitmap = Nothing
    Private Arrow As Bitmap = Nothing
    Private LastVTG As New tmagpsapi.NMEA_GPVTG

    Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.
    FormClosingEventArgs) Handles Me.FormClosing
        CloseComport()
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        ShowComportSpeeds()
        ShowComports()
        Dim tmpDbl As Double = 55.67582
        tarLat.Text = tmpDbl.ToString
        tmpDbl = 12.57008
        tarLon.Text = tmpDbl.ToString
        Compas = pb1.Image
        Arrow = pb2.Image
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnGPGGA.Click
        gps.NMEA_POS.GPGGA = tGPGGA.Text
    End Sub

    Private Sub gps_ErrorDecodingGPGGA(ByVal ErrorMessage As String) Handles gps.
    ErrorDecodingGPGGA
        Log("Err:" & ErrorMessage)
    End Sub

    Private Sub gps_GeneralError(ByVal Func As String, ByVal Ex As System.Exception) Handles
    gps.GeneralError
        MsgBox("General error." & vbCrLf & "Module:" & Func & vbCrLf & "Error:" & Ex.Message)
    End Sub

    Private Sub ShowComports()
        cboCom.Items.Clear()
        For Each sp As String In My.Computer.Ports.SerialPortNames
            cboCom.Items.Add(Replace(sp, "COM", ""))
        Next
        If cboCom.Items.Count > 0 Then
            cboCom.SelectedIndex = cboCom.Items.Count - 1
            cboSpeed.SelectedIndex = 2
        End If
    End Sub

    Private Sub ShowComportSpeeds()
        cboSpeed.Items.Clear()
        cboSpeed.Items.Add("1200")
        cboSpeed.Items.Add("2400")
        cboSpeed.Items.Add("4800")
        cboSpeed.Items.Add("9600")
        cboSpeed.Items.Add("14400")
        cboSpeed.Items.Add("19200")
        cboSpeed.Items.Add("38400")
        cboSpeed.Items.Add("57600")
    End Sub

    Private Sub gps_SuccessfulFix(ByVal Position As tmagpsapi.NMEA_Position) Handles gps.
    SuccessfulFix
        With Position
            tFixStatus.Text = .FixStatus
            lfixstatus.Text = .FixStatus.ToString
            tLatitude.Text = .Latitude
        End With
    End Sub
End Class
```

```

tLongitude.Text = .Longitude
tLatDec.Text = .LatitudeDecimal
tLonDec.Text = .LongitudeDecimal
tLatNaval.Text = .toNavalLatitude
tLonNaval.Text = .toNavalLongitude
tlatHemis.Text = .LatitudeHemisphere.ToString
tlonHemis.Text = .LongitudeHemisphere.ToString
tSats.Text = .NumberOfSats
tElevation.Text = .Altitude.ToString
tGoogle.Text = .toGoogleMapURL
If chkTrack.Checked = True Then
    Dim Transform As New tmagpsapi.NMEA_Transform
    Dim oTarget As tmagpsapi.NMEA_Transform.structTarget = Nothing
    oTarget = Transform.GetCourseAndDistance(.LatitudeDecimal, .LongitudeDecimal, .
Altitude, CType(tarLat.Text, Double), CType(tarLon.Text, Double), .Altitude)
    tarDist.Text = Math.Round(oTarget.Distance, 2).ToString & " Meters"
    If oTarget.Distance > 1000 Then
        lblTgtRange.Text = Math.Round(oTarget.Distance / 1000, 2).ToString & "KM"
    Else
        lblTgtRange.Text = Math.Round(oTarget.Distance, 2).ToString & "M"
    End If
    tarDir.Text = Math.Round(oTarget.Course, 2).ToString & "°"
    DrawDirection(oTarget.Course)
Else
    DrawDirection(0)
End If
End With
End Sub

Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    If Not sp Is Nothing Then
        If sp.IsComportOpen Then
            Log("Comport already open")
            Exit Sub
        End If
    End If
    sp = New tmagpsapi.tmaSerialport
    sp.Openport(CInt(cboCom.Text), IO.Ports.Parity.None, tmagpsapi.tmaSerialport.
enumDatabits.Bit8, IO.Ports.StopBits.One, CInt(cboSpeed.Text))
End Sub

Private Sub sp_ComPortClosed() Handles sp.ComPortClosed
    Log("Comport closed")
End Sub

Private Sub Log(ByVal Message As String)
    lstLog.Items.Insert(0, Message)
End Sub

Private Sub sp_ComPortError(ByVal ex As System.Exception, ByVal Message As String)
Handles sp.ComPortError
    Log("ComportError: (" & Message & ") " & ex.Message)
End Sub

Private Sub sp_ComPortOpen() Handles sp.ComPortOpen
    Log("Comport open")
End Sub

Private Sub sp_LineRecieved(ByVal Data As String) Handles sp.LineRecieved
    Dim tmparr() As String
    tmparr = Split(Data, ",")
    Select Case tmparr(0)
        Case "$GPGGA"
            ProcessGPGGA(Data)
        Case "$GPVTG"
            ProcessVTG(Data)
    End Select
    GpsData(Data)
End Sub

Private Sub ProcessVTG(ByVal Data As String)
    If InvokeRequired Then
        Try
            Dim dlg As New dlgData(AddressOf ProcessVTG)
            Dim Arguments() As Object = {Data}

```

```

        Invoke(dlg, Arguments)
    Catch
        'Ignore errors
    End Try
Else
    gps.NMEA_Direction.VTG = Data
End If

End Sub

Private Sub ProcessGPGGA(ByVal Data As String)
    If InvokeRequired Then
        Try
            Dim dlg As New dlgData(AddressOf ProcessGPGGA)
            Dim Arguments() As Object = {Data}
            Invoke(dlg, Arguments)
        Catch
            'Ignore errors
        End Try
    Else
        gps.NMEA_POS.GPGGA = Data
    End If
End Sub

Private Sub GpsData(ByVal Data As String)
    If Me.InvokeRequired Then
        Dim dlg As New dlgGPS(AddressOf GpsData)
        Dim Arguments() As Object = {Data}
        Invoke(dlg, Arguments)
    Else
        If lstGPS.Items.Count > 5 Then
            lstGPS.Items.RemoveAt(5)
        End If
        Data = Replace(Data, Chr(10), "")
        Data = Replace(Data, Chr(13), "")
        lstGPS.Items.Insert(0, Data)
    End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    CloseComport()
End Sub

Private Sub CloseComport()
    If sp Is Nothing Then Exit Sub
    If sp.IsComportOpen Then
        sp.Close()
        sp = Nothing
    Else
        Log("Comport is not open")
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    If tGoogle.Text <> "" Then
        System.Diagnostics.Process.Start(tGoogle.Text)
    End If
End Sub

Private Sub DrawDirection(ByVal Angle As Double)
    'Rotate the arrow towards to target position - relative to your current heading
    If chkTrack.Checked Then
        If Angle > LastVTG.TrueTrack Then
            Angle = Angle - LastVTG.TrueTrack
        Else
            Dim tmpDbl As Double = 360 - LastVTG.TrueTrack
            Angle = Angle + tmpDbl
            If Angle > 360 Then Angle = Angle - 360
        End If
        pb2.Image = Arrow
        pb2.Refresh()
        RotateImage(pb2, Angle)
    End If

```

```
'Rotate the compas according to your current heading
If LastVTG.TrueTrack > 0 Then
    RotateImage(pb1, LastVTG.TrueTrack)
End If
'Draw the little red line on top of the compas
Dim gr As Graphics = Panell.CreateGraphics
gr.DrawLine(Pens.Red, CInt(Panell.Width / 2), 1, CInt(Panell.Width / 2), 11)
End Sub

Private Sub gps_SucessfulSpeedDirection(ByVal VTG As tmagpsapi.NMEA_GPVGTG) Handles gps.
SucessfulSpeedDirection
    LastVTG = VTG
    lblKmt.Text = VTG.SpeedKMT.ToString & " km/t"
    lblKnt.Text = VTG.SpeedKnots.ToString & " Knots"
    lblHdg.Text = VTG.TrueTrack.ToString & "°"
    lblSpd.Text = Math.Round(VTG.SpeedKMT, 2).ToString & " km/t"
    proSpd.Value = CInt(VTG.SpeedKMT)
End Sub

Public Sub RotateImage(ByVal PictureBox As PictureBox, ByVal Degrees As Double)
    Dim g As Graphics = PictureBox.CreateGraphics
    Dim CurBitmap = PictureBox.Image
    Dim x As New System.Drawing.Drawing2D.Matrix
    x.RotateAt(Degrees, New PointF(CurBitmap.Width / 2, CurBitmap.Height / 2))
    g.Transform = x
    Dim MaxBredde As Double = 0
    MaxBredde = (CurBitmap.Height ^ 2) + (CurBitmap.Width ^ 2)
    MaxBredde = Math.Sqrt(MaxBredde)
    g.DrawImage(CurBitmap, New Rectangle(0, 0, CurBitmap.Width, CurBitmap.Height), 0, 0,
CurBitmap.Width, CurBitmap.Height, GraphicsUnit.Pixel)
End Sub

Private Sub lstLog_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System
.EventArgs) Handles lstLog.SelectedIndexChanged
    MsgBox(lstLog.SelectedItem)
End Sub
End Class
```