

# ***Explorer v3***

Manuale programmatore.

*by ROSA Davide*

## **Indice:**

Introduzione	Pagina 2
Listato del form di avvio.	Pagina 3
Commento form di avvio	Pagina 5
Listato form principale	Pagina 6
Commento form principale	Pagina 18
Listato form mappa	Pagina 19
Commento form mappa	Pagina 20
Listato modulo	Pagina 20
Commento modulo	Pagina 20

## **Introduzione:**

Per l'utilizzo del programma durante l'esecuzione consultare il manuale utente fornito con il programma.

In questo manuale viene spiegato il funzionamento del codice stesso del programma.

Il progetto del programma Explorer è composto da:

- Form di avvio
- Form del programma principale
- Form del mappa.
- Un modulo contenenti le variabile utilizzate in comune dai form.

## Listato form di avvio:

```
Dim dirlbl As Boolean
Dim XA, YA, VA, M1A, M2A, M3A, M4A As Integer
Dim XB, YB, VB, M1B, M2B, M3B, M4B As Integer
Dim XC, YC, VC, M1C, M2C, M3C, M4C As Integer
Dim XD, YD, VD, M1D, M2D, M3D, M4D As Integer
Dim XMAX, YMAX, XMIN, YMIN As Integer

Private Sub Form_Activate()
    LoopAnimazione
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    frmMain.Show
    frmAvvio.Hide
End Sub

Private Sub Form_Load()
    dirlbl = True
    XMAX = frmAvvio.Width
    XMIN = 0
    YMAX = frmAvvio.Height
    YMIN = 0
    M1A = 1
    M2A = 1
    M3A = 0
    M4A = 0
    M1B = 1
    M2B = 0
    M3B = 0
    M4B = 1
    M1C = 0
    M2C = 1
    M3C = 1
    M4C = 0
    M1D = 0
    M2D = 0
    M3D = 1
    M4D = 1
    Randomize
    XA = Int(XMAX * Rnd)
    YA = Int(YMAX * Rnd)
    XB = Int(XMAX * Rnd)
    YB = Int(YMAX * Rnd)
    XC = Int(XMAX * Rnd)
    YC = Int(YMAX * Rnd)
    XD = Int(XMAX * Rnd)
    YD = Int(YMAX * Rnd)
    VA = 20
    VB = 20
    VC = 20
    VD = 20
    frmAvvio.lblExplorev3.Left = (frmAvvio.Width / 2) - (frmAvvio.lblExplorev3.Width / 2)
    frmAvvio.lblProgetto.Left = (frmAvvio.Width / 2) - (frmAvvio.lblProgetto.Width / 2)
    frmAvvio.lblRealizzato.Left = (frmAvvio.Width / 2) - (frmAvvio.lblRealizzato.Width / 2)
    frmAvvio.lblRosaDavide.Left = (frmAvvio.Width / 2) - (frmAvvio.lblRosaDavide.Width / 2)
    frmAvvio.Linea(0).Visible = True
    frmAvvio.Linea(0).X1 = XA
    frmAvvio.Linea(0).Y1 = YA
```

```

frmAvvio.Linea(0).X2 = XB
frmAvvio.Linea(0).Y2 = YB
frmAvvio.Linea(1).Visible = True
frmAvvio.Linea(1).X1 = XB
frmAvvio.Linea(1).Y1 = YB
frmAvvio.Linea(1).X2 = XC
frmAvvio.Linea(1).Y2 = YC
frmAvvio.Linea(2).Visible = True
frmAvvio.Linea(2).X1 = XC
frmAvvio.Linea(2).Y1 = YC
frmAvvio.Linea(2).X2 = XD
frmAvvio.Linea(2).Y2 = YD
frmAvvio.Linea(3).Visible = True
frmAvvio.Linea(3).X1 = XD
frmAvvio.Linea(3).Y1 = YD
frmAvvio.Linea(3).X2 = XA
frmAvvio.Linea(3).Y2 = YA
End Sub

```

```
Private Sub LoopAnimazione()
```

```
LoopAni:
```

```
DoEvents
```

```
If (frmAvvio.lblPremiUnTasto.Left < 0) Or (frmAvvio.lblPremiUnTasto.Left > (frmAvvio.Width -
frmAvvio.lblPremiUnTasto.Width)) Then dirlbl = Not dirlbl
```

```
If dirlbl = True Then
```

```
frmAvvio.lblPremiUnTasto.Left = frmAvvio.lblPremiUnTasto.Left + 50
```

```
Else
```

```
frmAvvio.lblPremiUnTasto.Left = frmAvvio.lblPremiUnTasto.Left - 50
```

```
End If
```

```
frmAvvio.Linea(0).X1 = XA
```

```
frmAvvio.Linea(0).Y1 = YA
```

```
frmAvvio.Linea(0).X2 = XB
```

```
frmAvvio.Linea(0).Y2 = YB
```

```
frmAvvio.Linea(1).X1 = XB
```

```
frmAvvio.Linea(1).Y1 = YB
```

```
frmAvvio.Linea(1).X2 = XC
```

```
frmAvvio.Linea(1).Y2 = YC
```

```
frmAvvio.Linea(2).X1 = XC
```

```
frmAvvio.Linea(2).Y1 = YC
```

```
frmAvvio.Linea(2).X2 = XD
```

```
frmAvvio.Linea(2).Y2 = YD
```

```
frmAvvio.Linea(3).X1 = XD
```

```
frmAvvio.Linea(3).Y1 = YD
```

```
frmAvvio.Linea(3).X2 = XA
```

```
frmAvvio.Linea(3).Y2 = YA
```

```
If XA < XMIN + VA Then M1A = 1: M3A = 0
```

```
If YA > YMAX - VA Then M2A = 1: M4A = 0
```

```
If XA > XMAX - VA Then M3A = 1: M1A = 0
```

```
If YA < YMIN + VA Then M4A = 1: M2A = 0
```

```
If M1A = 1 And M2A = 1 Then XA = XA + VA: YA = YA - VA
```

```
If M1A = 1 And M4A = 1 Then XA = XA + VA: YA = YA + VA
```

```
If M3A = 1 And M2A = 1 Then XA = XA - VA: YA = YA - VA
```

```
If M3A = 1 And M4A = 1 Then XA = XA - VA: YA = YA + VA
```

```
If XB < XMIN + VB Then M1B = 1: M3B = 0
```

```
If YB > YMAX - VB Then M2B = 1: M4B = 0
```

```
If XB > XMAX - VB Then M3B = 1: M1B = 0
```

```
If YB < YMIN + VB Then M4B = 1: M2B = 0
```

```
If M1B = 1 And M2B = 1 Then XB = XB + VB: YB = YB - VB
```

```
If M1B = 1 And M4B = 1 Then XB = XB + VB: YB = YB + VB
```

```
If M3B = 1 And M2B = 1 Then XB = XB - VB: YB = YB - VB
If M3B = 1 And M4B = 1 Then XB = XB - VB: YB = YB + VB
```

```
If XC < XMIN + VC Then M1C = 1: M3C = 0
If YC > YMAX - VC Then M2C = 1: M4C = 0
If XC > XMAX - VC Then M3C = 1: M1C = 0
If YC < YMIN + VC Then M4C = 1: M2C = 0
If M1C = 1 And M2C = 1 Then XC = XC + VC: YC = YC - VC
If M1C = 1 And M4C = 1 Then XC = XC + VC: YC = YC + VC
If M3C = 1 And M2C = 1 Then XC = XC - VC: YC = YC - VC
If M3C = 1 And M4C = 1 Then XC = XC - VC: YC = YC + VC
```

```
If XD < XMIN + VD Then M1D = 1: M3D = 0
If YD > YMAX - VD Then M2D = 1: M4D = 0
If XD > XMAX - VD Then M3D = 1: M1D = 0
If YD < YMIN + VD Then M4D = 1: M2D = 0
If M1D = 1 And M2D = 1 Then XD = XD + VD: YD = YD - VD
If M1D = 1 And M4D = 1 Then XD = XD + VD: YD = YD + VD
If M3D = 1 And M2D = 1 Then XD = XD - VD: YD = YD - VD
If M3D = 1 And M4D = 1 Then XD = XD - VD: YD = YD + VD
```

```
GoTo LoopAni
```

```
End Sub
```

### *Commento form di avvio:*

Nel form di avvio viene presentato il programma, nel codice si nota che quando il form parte imposta delle variabili addette all'animazione del poligono del della scritta "Premere un tasto per continuare...". Dopo di che entra in un ciclo infinito finché non viene premuto un tasto qualsiasi, avvenuto ciò mostra il form principale e nasconde quello di avvio.

## Listato del form principale:

```
'Variabili globali del trogramma principale
Dim StepMotore1, StepMotore2, bib As Double
Dim StepPer1cm, StepPerLunghezza As Double
Dim VecStepMotore1, VecStepMotore2 As Double
Dim NumStepMotore1, NumStepMotore2 As Double
Dim StepMotore1Corrente, StepMotore2Corrente As Double
Dim StepMotoriTotale, NumStepRotazione90 As Double
Dim NSAvantiMotore1, NSIndietroMotore1 As Double
Dim NSAvantiMotore2, NSIndietroMotore2, IndexOstacoli As Double
Dim TasAvanti, TasIndietro, TasSinistra, TasDestra, TasGuidaAI As Boolean
Dim No, Ns, StopC As Boolean
Dim IndPortal, IndPortaO, VettoreRiconoscimentoVeicolo As String
'Funzioni per i/o sulla porta
Private Declare Function Inp Lib "InOut32.Dll" (ByVal IOPort As Integer) As Byte
Private Declare Sub Out Lib "InOut32.Dll" (ByVal IOPort As Integer, ByVal Data As Byte)
'Funzione per convertire da decimale a binario
Private Function DecBin(ByVal bib As Integer) As String
    Dim abin, db1, db2 As Integer
    Dim b1, b2 As String
    b1 = ""
    b2 = ""
    For abin = 1 To 8
        DoEvents
        db1 = bib Mod 2
        db2 = Int(bib / 2)
        bib = db2
        b1 = Str$(db1)
        b2 = Trim(b2) & Trim(b1)
    Next abin
    DecBin = b2
End Function
'Funzione per convertire da binario a decimale
Private Function BinDec(ByVal bib As String) As Integer
    Dim x As Integer
    x = 0
    If Mid(bib, 1, 1) = "1" Then x = x + 1
    If Mid(bib, 2, 1) = "1" Then x = x + 2
    If Mid(bib, 3, 1) = "1" Then x = x + 4
    If Mid(bib, 4, 1) = "1" Then x = x + 8
    If Mid(bib, 5, 1) = "1" Then x = x + 16
    If Mid(bib, 6, 1) = "1" Then x = x + 32
    If Mid(bib, 7, 1) = "1" Then x = x + 64
    If Mid(bib, 8, 1) = "1" Then x = x + 128
    BinDec = x
End Function

Private Sub CtrlMtr(NMotore As Integer, Direzione As Integer)
    If NMotore = 1 Then
        If Direzione = 1 Then
            StepMotore1 = StepMotore1 + 1
            If StepMotore1 > 4 Then StepMotore1 = 1
        Else
            StepMotore1 = StepMotore1 - 1
            If StepMotore1 < 1 Then StepMotore1 = 4
        End If
    Else
        If Direzione = 1 Then
            StepMotore2 = StepMotore2 + 1
            If StepMotore2 > 4 Then StepMotore2 = 1
        Else
            StepMotore2 = StepMotore2 - 1
            If StepMotore2 < 1 Then StepMotore2 = 4
        End If
    End Sub
```

```

Else
    StepMotore2 = StepMotore2 - 1
    If StepMotore2 < 1 Then StepMotore2 = 4
End If
End If
End Sub

```

```

Private Sub chkPorta278_Click()
    IndPortaO = "&H278"
    IndPortal = "&H279"
End Sub

```

```

Private Sub chkPorta378_Click()
    IndPortaO = "&H378"
    IndPortal = "&H379"
End Sub

```

```

Private Sub chkVisualizzaMappa_Click()
    If frmMain.chkVisualizzaMappa.Value = 1 Then
        frmMappa.Show
    Else
        frmMappa.Hide
    End If
End Sub

```

```

Private Sub cmdAbilitaDisabilitaGiodaAI_Click()
    TasGuidaAI = Not TasGuidaAI
    If TasGuidaAI = True Then
        frmMain.lstAzzioniAI.Clear
        frmMain.txtAzioneCorrente.Text = "Attivata guida AI"
        frmMain.lstAzzioniAI.AddItem frmMain.txtAzioneCorrente.Text
    Else
        frmMain.txtAzioneCorrente.Text = "Disattivata guida AI"
        frmMain.lstAzzioniAI.AddItem frmMain.txtAzioneCorrente.Text
    End If
End Sub

```

```

Private Sub cmdComandoAvantiDiNStep_Click()
    Dim x As Integer
    StopC = False
    For x = 1 To Val(frmMain.txtNumeroStep.Text)
        DoEvents
        If StopC = True Then Exit For
        CtrlMtr 1, 1
        CtrlMtr 2, 1
        OutDati
    Next x
End Sub

```

```

Private Sub cmdComandoAvantiLoop_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasAvanti = True
End Sub

```

```

Private Sub cmdComandoAvantiLoop_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasAvanti = False
End Sub

```

```

Private Sub cmdComandoIndietroDiNStep_Click()
    Dim x As Integer
    StopC = False

```

```

For x = 1 To Val(frmMain.txtNumeroStep.Text)
    DoEvents
    If StopC = True Then Exit For
    CtrlMtr 1, 2
    CtrlMtr 2, 2
    OutDati
Next x
End Sub

Private Sub cmdComandoIndietroLoop_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasIndietro = True
End Sub

Private Sub cmdComandoIndietroLoop_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasIndietro = False
End Sub

Private Sub cmdDestra_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasDestra = True
End Sub

Private Sub cmdDestra_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasDestra = False
End Sub

Private Sub cmdGiraADestraDiN_Click()
    Dim x As Integer
    StopC = False
    For x = 1 To ((NumStepRotazione90 * 4) * Val(frmMain.txtGiraADestraDiN.Text)) / 360
        DoEvents
        If StopC = True Then Exit For
        CtrlMtr 1, 1
        CtrlMtr 2, 2
        OutDati
    Next x
End Sub

Private Sub cmdGiraASinistraDiN_Click()
    Dim x As Integer
    StopC = False
    For x = 1 To ((NumStepRotazione90 * 4) * Val(frmMain.txtGiraASinistraDiN.Text)) / 360
        DoEvents
        If StopC = True Then Exit For
        CtrlMtr 1, 2
        CtrlMtr 2, 1
        OutDati
    Next x
End Sub

Private Sub cmdMotore1Avanti_Click()
    CtrlMtr 1, 1
    OutDati
End Sub

Private Sub cmdMotore1Indietro_Click()
    CtrlMtr 1, 2
    OutDati
End Sub

Private Sub cmdMotore2Avanti_Click()

```

```

    CtrlMtr 2, 1
    OutDati
End Sub

```

```

Private Sub cmdMotore2Indietro_Click()
    CtrlMtr 2, 2
    OutDati
End Sub

```

```

Private Sub AnalisiSensori()
'Letture dei dati nella porta input
    bib = Inp(Val(IndPortal) And &HFFFF)
    frmMain.StatusBar1.Panels(2).Text = DecBin(bib) & " = " & bib
'Controllo sensore
    If Mid(DecBin(bib), 4, 1) = "1" Then
        frmMain.txtSensore1.Text = "ATTIVO"
        If frmMain.chkAttivaAllarmiSensori.Value = 1 Then
            MsgBox "Sensore attivo..Motori bloccati!", vbCritical, "Attenzione"
        End If
    Else
        frmMain.txtSensore1.Text = "disattivo"
    End If
End Sub

```

```

Private Sub cmdNumeroDiStepPerRotazioneDi90_Click()
'Change del numero di step per una rotazione di 90°
    NumStepRotazione90 = Val(frmMain.txtNumeroDiStepPerRotazioneDi90.Text)
End Sub

```

```

Private Sub cmdOKLunghezzaPer1cm_Click()
    StepPer1cm = Val(frmMain.txtLunghezzaPer1cm.Text)
End Sub

```

```

Private Sub cmdOkStepMotori_Click()
'Change del numero degli step totali
    StepMotoriTotale = Val(frmMain.txtNumeroStepMotori.Text)
End Sub

```

```

Private Sub cmdSalvaOpzioni_Click()
'Salvataggio impostazioni
    Open "opzioni.cfg" For Output As #1
        Print #1, frmMain.txtRitordoOutput.Text
        Print #1, frmMain.chkPorta378.Value
        Print #1, frmMain.chkPorta278.Value
        Print #1, frmMain.chkVisualizzaMappa.Value
        Print #1, frmMain.chkAttivaSensoreOstacoli.Value
        Print #1, frmMain.txtNumeroStepMotori.Text
        Print #1, frmMain.txtNumeroDiStepPerRotazioneDi90.Text
        Print #1, frmMain.chkAttivaAllarmiSensori.Value
        Print #1, frmMain.chkCodificaPerDecoder.Value
        Print #1, frmMain.txtZoom.Text
        Print #1, frmMain.txtLunghezzaPer1cm.Text
        Print #1, frmMain.txtVettoreRiconoscimentoVeicolo.Text
    Close #1
    cmdOkStepMotori_Click
    If Zoom <> Val(frmMain.txtZoom.Text) Then cmdZoom_Click
End Sub

```

```

Private Sub cmdSinistra_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasSinistra = True
End Sub

```

```

Private Sub cmdSinistra_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    TasSinistra = False
End Sub

Private Sub cmdStop_Click()
    StopC = True
End Sub

Private Sub cmdZoom_Click()
Dim x As Integer
    x = MsgBox("Per cambiare lo zoom devo cancellare la mappa.. Continuare?", vbYesNo, "Attenzione")
    If x = 7 Then
        frmMain.txtZoom.Text = Zoom
        Exit Sub
    End If
    On Error GoTo Continua
    For x = 1 To IndexOstacoli
        DoEvents
        Unload frmMappa.linOstacolo(x)
    Next x
Continua:
    Zoom = Abs(Val(frmMain.txtZoom.Text))
    frmMain.txtZoom.Text = Zoom
    OutDati
End Sub

Private Sub Form_Activate()
    LoopControllo
End Sub

Private Sub Form_Load()
    Dim x As Variant
'Posizionamento form principale sullo schermo
    frmMain.Top = 0
    frmMain.Left = (Screen.Width / 2) - (frmMain.Width / 2)
    Randomize Timer
    TasAvanti = False
    TasIndietro = False
    TasGuidaAI = False
    On Error GoTo Continua
'Caricamento opzioni salvate
    Open "opzioni.cfg" For Input As #1
        Input #1, x
        frmMain.txtRitordoOutput.Text = x
        Input #1, x
        frmMain.chkPorta378.Value = x
        Input #1, x
        frmMain.chkPorta278.Value = x
        Input #1, x
        frmMain.chkVisualizzaMappa.Value = x
        Input #1, x
        frmMain.chkAttivaSensoreOstacoli.Value = x
        Input #1, x
        frmMain.txtNumeroStepMotori.Text = x
        Input #1, x
        frmMain.txtNumeroDiStepPerRotazioneDi90.Text = x
        Input #1, x
        frmMain.chkAttivaAllarmiSensori.Value = x
        Input #1, x
        frmMain.chkCodificaPerDecoder.Value = x
        Input #1, x
        frmMain.txtZoom.Text = x

```

```

    Input #1, x
    frmMain.txtLunghezzaPer1cm.Text = x
    Input #1, x
    frmMain.txtVettoreRiconoscimentoVeicolo.Text = x
Continua:
    Close #1
'Caricamento valori di default
    If Err.Number = 53 Then
        frmMain.txtRitordoOutput.Text = 0
        frmMain.chkPorta378.Value = True
        frmMain.chkPorta278.Value = False
        frmMain.chkVisualizzaMappa.Value = 0
        frmMain.chkAttivaSensoreOstacoli.Value = 0
        frmMain.txtNumeroStepMotori.Text = 219
        frmMain.txtNumeroDiStepPerRotazioneDi90.Text = 400
        frmMain.chkAttivaAllarmiSensori.Value = 0
        frmMain.chkCodificaPerDecoder.Value = 0
        frmMain.txtZoom.Text = 1
        frmMain.txtLunghezzaPer1cm.Text = 219
        frmMain.txtVettoreRiconoscimentoVeicolo.Text = "1110"
    End If
    StepMotoriTotale = Val(frmMain.txtNumeroStepMotori.Text)
    NumStepRotazione90 = Val(frmMain.txtNumeroDiStepPerRotazioneDi90.Text)
    VettoreRiconoscimentoVeicolo = frmMain.txtVettoreRiconoscimentoVeicolo.Text
    StepMotore1 = 1
    StepMotore2 = 1
    StepPer1cm = frmMain.txtLunghezzaPer1cm.Text
    StepPerLunghezza = 0
    Zoom = Val(frmMain.txtZoom.Text)
    VecStepMotore1 = StepMotore1
    VecStepMotore2 = StepMotore2
    NumStepMotore1 = 0
    StepMotore1Corrente = 0
    NumStepMotore2 = 0
    StepMotore2Corrente = 0
    NSAvantiMotore1 = 0
    NSIndietroMotore1 = 0
    NSAvantiMotore2 = 0
    NSIndietroMotore2 = 0
    IndexOstacoli = 0
    frmMain.lstAzzioniAI.Clear
    frmMain.txtAzioneCorrente.Text = ""
    Ns = True
    If frmMain.chkPorta378.Value = True Then
        IndPortaO = "&H378"
        IndPortal = "&H379"
    Else
        IndPortaO = "&H278"
        IndPortal = "&H279"
    End If
End Sub

Private Sub OutDati()
    Dim db1 As Integer
    Dim TimeRitardo As Double
'Visualizza veicolo nella mappa
    ControlVeicolo
'Salvataggio step 1° motore
    VecStepMotore1 = Val(frmMain.txtStepMotore1.Text)
'Visualizza step 2° motore
    VecStepMotore2 = Val(frmMain.txtStepMotore2.Text)
'Visualizza step 1° motore

```

```

frmMain.txtStepMotore1.Text = StepMotore1
'Visualizza step 2° motore
frmMain.txtStepMotore2.Text = StepMotore2
'Conteggio statistiche 1° motore
If VecStepMotore1 <> StepMotore1 Then
    NumStepMotore1 = NumStepMotore1 + 1
    If (VecStepMotore1 = 1 And StepMotore1 = 2) Or (VecStepMotore1 = 2 And StepMotore1 = 3) Or
(VecStepMotore1 = 3 And StepMotore1 = 4) Or (VecStepMotore1 = 4 And StepMotore1 = 1) Then
        StepMotore1Corrente = StepMotore1Corrente + 1
        NSAvantiMotore1 = NSAvantiMotore1 + 1
        frmMain.txtDiagnosiMotore1.Text = "Avanti"
    Elseif (VecStepMotore1 = 1 And StepMotore1 = 4) Or (VecStepMotore1 = 4 And StepMotore1 = 3) Or
(VecStepMotore1 = 3 And StepMotore1 = 2) Or (VecStepMotore1 = 2 And StepMotore1 = 1) Then
        StepMotore1Corrente = StepMotore1Corrente - 1
        NSIndietroMotore1 = NSIndietroMotore1 + 1
        frmMain.txtDiagnosiMotore1.Text = "Indietro"
    End If
End If
'Conteggio statistiche 2° motore
If VecStepMotore2 <> StepMotore2 Then
    NumStepMotore2 = NumStepMotore2 + 1
    If (VecStepMotore2 = 1 And StepMotore2 = 2) Or (VecStepMotore2 = 2 And StepMotore2 = 3) Or
(VecStepMotore2 = 3 And StepMotore2 = 4) Or (VecStepMotore2 = 4 And StepMotore2 = 1) Then
        StepMotore2Corrente = StepMotore2Corrente + 1
        NSAvantiMotore2 = NSAvantiMotore2 + 1
        frmMain.txtDiagnosiMotore2.Text = "Avanti"
    Elseif (VecStepMotore2 = 1 And StepMotore2 = 4) Or (VecStepMotore2 = 4 And StepMotore2 = 3) Or
(VecStepMotore2 = 3 And StepMotore2 = 2) Or (VecStepMotore2 = 2 And StepMotore2 = 1) Then
        StepMotore2Corrente = StepMotore2Corrente - 1
        NSIndietroMotore2 = NSIndietroMotore2 + 1
        frmMain.txtDiagnosiMotore2.Text = "Indietro"
    End If
End If
'Visualizzazione lunghezza pecorsa
If (frmMain.txtDiagnosiMotore1.Text = "Avanti" And frmMain.txtDiagnosiMotore2.Text = "Avanti") Or
(frmMain.txtDiagnosiMotore1.Text = "Indietro" And frmMain.txtDiagnosiMotore2.Text = "Indietro") Then
    StepPerLunghezza = StepPerLunghezza + 1
End If
frmMappa.StatusBar1.Panels(4).Text = Format(StepPerLunghezza / StepPer1cm, "#####0.0")
'Visualizza statistiche 1° motore
frmMain.txtNumStepMotore1.Text = NumStepMotore1
frmMain.txtNSAvantiMotore1.Text = NSAvantiMotore1
frmMain.txtNSIndietroMotore1.Text = NSIndietroMotore1
frmMain.txtNumGiriMotore1.Text = Format(NumStepMotore1 / StepMotoriTotale, "#####0.0")
'Visualizza statistiche 2° motore
frmMain.txtNumStepMotore2.Text = NumStepMotore2
frmMain.txtNSAvantiMotore2.Text = NSAvantiMotore2
frmMain.txtNSIndietroMotore2.Text = NSIndietroMotore2
frmMain.txtNumGiriMotore2.Text = Format(NumStepMotore2 / StepMotoriTotale, "#####0.0")
'Calcolo gradi del 1° motore
Y = (360 * (NSAvantiMotore1 - NSIndietroMotore1)) / StepMotoriTotale
If Y >= 360 Then
    For x = 1 To Int(Y / 360)
        Y = Y - 360
    Next x
End If
frmMain.txtGradiMotore1.Text = Format(Y, "#####0.0")
'Calcolo gradi del 2° motore
Y = (360 * (NSAvantiMotore2 - NSIndietroMotore2)) / StepMotoriTotale
If Y >= 360 Then
    For x = 1 To Int(Y / 360)
        Y = Y - 360
    Next x
End If

```

```

    Next x
End If
frmMain.txtGradiMotore2.Text = Format(Y, "#####0.0")
'Modifica linea di indicazione del 1° motore
frmMain.linMotore1.X1 = 2190 + (Cos((6.28318530717959 / StepMotoriTotale) * StepMotore1Corrente) *
570)
frmMain.linMotore1.Y1 = 840 + (Sin((6.28318530717959 / StepMotoriTotale) * StepMotore1Corrente) *
570)
'Modifica linea di indicazione del 2° motore
frmMain.linMotore2.X1 = 2190 + (Cos((6.28318530717959 / StepMotoriTotale) * StepMotore2Corrente) *
570)
frmMain.linMotore2.Y1 = 840 + (Sin((6.28318530717959 / StepMotoriTotale) * StepMotore2Corrente) *
570)
'Preparazione bit da trasmettere
If frmMain.chkCodificaPerDecoder.Value = 1 Then
'Codifica bit per decoder
    If StepMotore1 = 1 Then
        db1 = 0
    ElseIf StepMotore1 = 2 Then
        db1 = 1
    ElseIf StepMotore1 = 3 Then
        db1 = 2
    ElseIf StepMotore1 = 4 Then
        db1 = 3
    End If
    If StepMotore2 = 1 Then
        db1 = db1 + 0
    ElseIf StepMotore2 = 2 Then
        db1 = db1 + 4
    ElseIf StepMotore2 = 3 Then
        db1 = db1 + 8
    ElseIf StepMotore2 = 4 Then
        db1 = db1 + 12
    End If
Else
'Codifica bit senza decoder
    If StepMotore1 = 1 Then
        db1 = 1
    ElseIf StepMotore1 = 2 Then
        db1 = 2
    ElseIf StepMotore1 = 3 Then
        db1 = 4
    ElseIf StepMotore1 = 4 Then
        db1 = 8
    End If
    If StepMotore2 = 1 Then
        db1 = db1 + 16
    ElseIf StepMotore2 = 2 Then
        db1 = db1 + 32
    ElseIf StepMotore2 = 3 Then
        db1 = db1 + 64
    ElseIf StepMotore2 = 4 Then
        db1 = db1 + 128
    End If
End If
'Ciclo di ritardo
TimeRitardo = Timer + (Val(frmMain.txtRitardoOutput.Text) / 1000)
Do
    DoEvents
    AnalisiSensori
    If Mid(DecBin(bib), 5, 4) <> VettoreRiconoscimentoVeicolo Then

```

```

        If MsgBox("Attenzione!" & Chr$(13) & "Veicolo non connesso o non alimentato." & Chr$(13) &
        "Impossibile continuare,controllare il sistema." & Chr$(13) & "Continuare l'esecuzione del programma?",
        vbYesNo, "Errore!") = vbNo Then End
        End If
        Loop While TimeRitardo > Timer
'Trasmissione bit
        Out Val(IndPortaO), db1
        frmMain.StatusBar1.Panels(4).Text = DecBin(db1) & " = " & db1
End Sub

Private Sub StatusBar1_PanelClick(ByVal Panel As MSCComctlLib.Panel)
Dim x As Integer
    If Panel = "Esci" Then
        'Resettamente porta in uscita
        Out Val(IndPortaO), 0
        End
    End If
    If Panel = "Reset" Then
        'Cancellazione ostacoli sulla mappa
        For x = 1 To IndexOstacoli
            DoEvents
            Unload frmMappa.linOstacolo(x)
        Next
        'Ricarica opzioni
        Form_Load
    End If
End Sub

Private Sub Tabbella_Click()
    If frmMain.Tabbella.SelectedItem.Index = 1 Then
        frmMain.frmComandi.ZOrder
    ElseIf frmMain.Tabbella.SelectedItem.Index = 2 Then
        frmMain.frmGuidaAI.ZOrder
    ElseIf frmMain.Tabbella.SelectedItem.Index = 3 Then
        frmMain.frmMotori.ZOrder
    ElseIf frmMain.Tabbella.SelectedItem.Index = 4 Then
        frmMain.frmDiagnosi.ZOrder
    ElseIf frmMain.Tabbella.SelectedItem.Index = 5 Then
        frmMain.frmSensori.ZOrder
    ElseIf frmMain.Tabbella.SelectedItem.Index = 6 Then
        frmMain.frmOpzioni.ZOrder
    Else
        frmMain.frmInformazioni.ZOrder
    End If
End Sub

Private Sub LoopControllo()
LoopCtrl:
    DoEvents
'Controllo per il loop del tasto avanti della sezione comandi
    If TasAvanti = True Then
        CtrlMtr 1, 1
        CtrlMtr 2, 1
    End If
'Controllo per il loop del tasto indietro della sezione comandi
    If TasIndietro = True Then
        CtrlMtr 1, 2
        CtrlMtr 2, 2
    End If
'Controllo per il loop del tasto sinistra della sezione comandi
    If TasSinistra = True Then
        CtrlMtr 1, 2

```

```

    CtrlMtr 2, 1
End If
'Controllo per il loop del tasto destra della sezione comandi
If TasDestra = True Then
    CtrlMtr 1, 1
    CtrlMtr 2, 2
End If
'Controllo per la guida AI
If TasGuidaAI = True Then
    frmMain.cmdComandoAvantiDiNStep.Enabled = False
    frmMain.cmdComandoAvantiLoop.Enabled = False
    frmMain.cmdComandoIndietroDiNStep.Enabled = False
    frmMain.cmdComandoIndietroLoop.Enabled = False
    frmMain.cmdDestra.Enabled = False
    frmMain.cmdGiraADestraDiN.Enabled = False
    frmMain.cmdGiraASinistraDiN.Enabled = False
    frmMain.cmdMotore1Avanti.Enabled = False
    frmMain.cmdMotore1Indietro.Enabled = False
    frmMain.cmdMotore2Avanti.Enabled = False
    frmMain.cmdMotore2Indietro.Enabled = False
    frmMain.cmdSinistra.Enabled = False
    frmMain.cmdStop.Enabled = False
    GuidaAI
    frmMain.cmdComandoAvantiDiNStep.Enabled = True
    frmMain.cmdComandoAvantiLoop.Enabled = True
    frmMain.cmdComandoIndietroDiNStep.Enabled = True
    frmMain.cmdComandoIndietroLoop.Enabled = True
    frmMain.cmdDestra.Enabled = True
    frmMain.cmdGiraADestraDiN.Enabled = True
    frmMain.cmdGiraASinistraDiN.Enabled = True
    frmMain.cmdMotore1Avanti.Enabled = True
    frmMain.cmdMotore1Indietro.Enabled = True
    frmMain.cmdMotore2Avanti.Enabled = True
    frmMain.cmdMotore2Indietro.Enabled = True
    frmMain.cmdSinistra.Enabled = True
    frmMain.cmdStop.Enabled = True
End If
'Controllo di fermo del 1° motore
If VecStepMotore1 = StepMotore1 Then
    frmMain.txtDiagnosiMotore1.Text = "Fermo"
End If
'Controllo di fermo del 2° motore
If VecStepMotore2 = StepMotore2 Then
    frmMain.txtDiagnosiMotore2.Text = "Fermo"
End If
'Richiamo funzione di uscita dati
OutDati
GoTo LoopCtrl
End Sub

Private Sub GuidaAI()
Dim x, kr, Perc As Integer
'Guida AI
Perc = StepPerLunghezza
frmMain.lstAzzioniAI.AddItem " "
Do
    If Mid(DecBin(bib), 4, 1) = "0" Then
        CtrlMtr 1, 1
        CtrlMtr 2, 1
        If frmMain.lstAzzioniAI.List(frmMain.lstAzzioniAI.ListCount - 1) <> "Avanti" Then
            frmMain.lstAzzioniAI.RemoveItem (frmMain.lstAzzioniAI.ListCount - 1)

```

```

        frmMain.txtAzioneCorrente.Text = "Avanti di: " & Format((StepPerLunghezza - Perc) /
StepPer1cm, "#####0.0") & " cm"
        frmMain.lstAzzioniAI.AddItem frmMain.txtAzioneCorrente.Text
    End If
    OutDati
Else
'Ciclo di indietro finche non si disattiva il sensore
    Perc = StepPerLunghezza
    frmMain.lstAzzioniAI.AddItem "Indietro"
    Do While Mid(DecBin(bib), 4, 1) = "1" And TasGuidaAI = True
        frmMain.lstAzzioniAI.RemoveItem (frmMain.lstAzzioniAI.ListCount - 1)
        frmMain.txtAzioneCorrente.Text = "Indietro di: " & Format((StepPerLunghezza - Perc) /
StepPer1cm, "#####0.0") & " cm"
        frmMain.lstAzzioniAI.AddItem frmMain.txtAzioneCorrente.Text
        DoEvents
        CtrlMtr 1, 2
        CtrlMtr 2, 2
        OutDati
    Loop
    Ns = False
'Ciclo di indietro standard
    For x = 1 To NumStepRotazione90 / 2
        If TasGuidaAI = False Then Exit For
        frmMain.lstAzzioniAI.RemoveItem (frmMain.lstAzzioniAI.ListCount - 1)
        frmMain.txtAzioneCorrente.Text = "Indietro di: " & Format((StepPerLunghezza - Perc) /
StepPer1cm, "#####0.0") & " cm"
        frmMain.lstAzzioniAI.AddItem frmMain.txtAzioneCorrente.Text
        DoEvents
        CtrlMtr 1, 2
        CtrlMtr 2, 2
        OutDati
    Next x
'Scelta a random della direzione di svolta
    kr = Int((Rnd * 100) + 1)
    If TasGuidaAI = True Then
        If kr > 50 Then
            kr = 45 + Int((Rnd * 90) + 1)
            frmMain.txtAzioneCorrente.Text = "Gira di: " & kr & "° a destra"
            frmMain.lstAzzioniAI.AddItem frmMain.txtAzioneCorrente.Text
        End If
    End If
'Ciclo di svolta a destra
    For x = 1 To ((NumStepRotazione90 * kr) / 90)
        If TasGuidaAI = False Then Exit For
        DoEvents
        CtrlMtr 1, 1
        CtrlMtr 2, 2
        OutDati
    Next x
Else
    kr = 45 + Int((Rnd * 90) + 1)
    frmMain.txtAzioneCorrente.Text = "Gira di: " & kr & "° a sinistra"
    frmMain.lstAzzioniAI.AddItem frmMain.txtAzioneCorrente.Text
'Ciclo di svolta a sinistra
    For x = 1 To ((NumStepRotazione90 * kr) / 90)
        If TasGuidaAI = False Then Exit For
        DoEvents
        CtrlMtr 1, 2
        CtrlMtr 2, 1
        OutDati
    Next x
End If
End If
Ns = True

```

```

        Perc = StepPerLunghezza
        frmMain.lstAzzioniAI.AddItem " "
    End If
    Loop While TasGuidaAI = True
End Sub

Private Sub ControlVeicolo()
Dim r As Double
Dim d, x As Integer
    DoEvents
'Visualizzazione del numero di ostacoli sulla mappa
    frmMappa.StatusBar1.Panels(2).Text = frmMappa.linOstacolo.Count - 1
'Calcolo dell'angolo di rotazione del veicolo
    r = -(((NSAvantiMotore2 + NSIndietroMotore1) - (NSAvantiMotore1 + NSIndietroMotore2)) * 360) /
(NumStepRotazione90 * 8)
'Visualizzazione veicolo sulla mappa
    d = 300 * Zoom
    frmMappa.linVeicolo(0).X1 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r - 90 - 45)) * d))
    frmMappa.linVeicolo(0).Y1 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r - 90 - 45)) * d))
    frmMappa.linVeicolo(0).X2 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r - 45)) * d))
    frmMappa.linVeicolo(0).Y2 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r - 45)) * d))
    frmMappa.linVeicolo(1).X1 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r - 90 - 45)) * d))
    frmMappa.linVeicolo(1).Y1 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r - 90 - 45)) * d))
    frmMappa.linVeicolo(1).X2 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r + 90)) * d))
    frmMappa.linVeicolo(1).Y2 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r + 90)) * d))
    frmMappa.linVeicolo(2).X1 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r + 90)) * d))
    frmMappa.linVeicolo(2).Y1 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r + 90)) * d))
    frmMappa.linVeicolo(2).X2 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r - 45)) * d))
    frmMappa.linVeicolo(2).Y2 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r - 45)) * d))
'Creazione nuovo ostacolo sulla mappa
    d = 400 * Zoom
    If Mid(DecBin(bib), 4, 1) = "0" Then No = True
    If Mid(DecBin(bib), 4, 1) = "1" And No = True And Ns = True And frmMain.chkAttivaSensoreOstacoli.Value
= 1 Then
        No = False
        IndexOstacoli = IndexOstacoli + 1
        Load frmMappa.linOstacolo(IndexOstacoli)
        frmMappa.linOstacolo(IndexOstacoli).X1 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r + 90 + 45))
* d))
        frmMappa.linOstacolo(IndexOstacoli).Y1 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r + 90 + 45)) *
d))
        frmMappa.linOstacolo(IndexOstacoli).X2 = (xVeicolo + (Cos((6.28318530717959 / 360) * (r + 45)) * d))
        frmMappa.linOstacolo(IndexOstacoli).Y2 = (yVeicolo + (Sin((6.28318530717959 / 360) * (r + 45)) * d))
        frmMappa.linOstacolo(IndexOstacoli).Visible = True
    End If
'Spostamento ostacoli sulla mappa
    d = 1 * Zoom
    If frmMain.txtDiagnosiMotore1.Text = "Avanti" And frmMain.txtDiagnosiMotore2.Text = "Avanti" Then
        For x = 1 To IndexOstacoli
            DoEvents
            frmMappa.linOstacolo(x).X1 = (frmMappa.linOstacolo(x).X1 + (Cos((6.28318530717959 / 360) * (r -
90)) * d))
            frmMappa.linOstacolo(x).Y1 = (frmMappa.linOstacolo(x).Y1 + (Sin((6.28318530717959 / 360) * (r -
90)) * d))
            frmMappa.linOstacolo(x).X2 = (frmMappa.linOstacolo(x).X2 + (Cos((6.28318530717959 / 360) * (r -
90)) * d))
            frmMappa.linOstacolo(x).Y2 = (frmMappa.linOstacolo(x).Y2 + (Sin((6.28318530717959 / 360) * (r -
90)) * d))
        Next x
        ElseIf frmMain.txtDiagnosiMotore1.Text = "Indietro" And frmMain.txtDiagnosiMotore2.Text = "Indietro"
Then
            For x = 1 To IndexOstacoli

```

```

DoEvents
frmMappa.linOstacolo(x).X1 = (frmMappa.linOstacolo(x).X1 + (Cos((6.28318530717959 / 360) * (r +
90)) * d))
frmMappa.linOstacolo(x).Y1 = (frmMappa.linOstacolo(x).Y1 + (Sin((6.28318530717959 / 360) * (r +
90)) * d))
frmMappa.linOstacolo(x).X2 = (frmMappa.linOstacolo(x).X2 + (Cos((6.28318530717959 / 360) * (r +
90)) * d))
frmMappa.linOstacolo(x).Y2 = (frmMappa.linOstacolo(x).Y2 + (Sin((6.28318530717959 / 360) * (r +
90)) * d))
Next x
End If
End Sub

Private Sub uedGiraADestra_Change()
frmMain.txtGiraADestraDiN.Text = frmMain.uedGiraADestra.Value
End Sub

Private Sub uedGiraASinistra_Change()
frmMain.txtGiraASinistraDiN.Text = frmMain.uedGiraASinistra.Value
End Sub

Private Sub uedNumStep_Change()
frmMain.txtNumeroStep.Text = frmMain.uedNumStep.Value
End Sub

```

### *Commento form principale:*

Il form principale si occupa della gestione di tutto il progetto e delle richieste dell'utente.  
Le funzioni:

- Imp: questa funzione è contenuta in una dll esterna al programma, fornita da terzi, serve per l'input dalle periferiche di sistema avente l'indirizzo della periferica.
- Out: questa funzione è contenuta in una dll esterna al programma, fornita da terzi, serve per l'output sulle periferiche di sistema avente l'indirizzo della periferica.
- DecBin: converte un numero decimale in una stringa lunga 8 caratteri contenente il numero dato in forma binaria.
- BinDec: converte una stringa di 8 caratteri ,contenenti un numero in forma binaria, in un numero decimale.
- CtrlMtr: questa funzione si occupa del controllo delle bobine dei motori, questa funzione si avvale di 2 parametri, il primo assume 2 valori che corrispondono al numero del motore, mentre il secondo parametri assume due valori che corrisponde alla direzione in cui la bobina deve scorrere.
- AnalisiSensori: si occupa della lettura sulla periferica e nel caso del visualizzare un messaggio.
- Form\_Load: nella fase di caricamento del form vengono aggiornate tutte le variabili, caricando anche quelle salvate precedentemente.
- OutDati: si occupa della visualizzazione dei dati sul form e dell'output dei dati sulla periferica.
- LoopCtrl: questa funzione viene richiamata all'avvio del form, crea un loop infinito nel quale vengono richiamate le varie funzioni richieste dall'utente.
- GuidaAI: questa funzione si occupa della guida automatica del programma.
- ControlVeicolo: questa funzione si occupa della visualizzazione del veicolo e degli ostacoli sul form mappa, movendoli anche quando occorre.

## Listato form mappa:

```
Private Sub ChiudiMappa_Click()
    Unload Me
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then Unload Me
End Sub

Private Sub Form_Load()
    Dim r As Double
    'Posiziona mappa sullo schermo
    frmMappa.Top = frmMain.Height
    frmMappa.Left = 0
    frmMappa.Width = Screen.Width
    frmMappa.Height = Screen.Height - frmMain.Height - 400
    frmMappa.shpMargineMappa.Top = 0
    frmMappa.shpMargineMappa.Left = 0
    frmMappa.shpMargineMappa.Width = frmMappa.Width - 140
    frmMappa.shpMargineMappa.Height = frmMappa.Height - 900
    'Rilevamento coordinate centro mappa
    xVeicolo = (frmMappa.Width - 140) / 2
    yVeicolo = (frmMappa.Height - 900) / 2
    'Visualizzazione veicolo sulla mappa
    d = 300 * Zoom
    frmMappa.linVeicolo(0).X1 = (xVeicolo + (Cos((6.28318530717959 / 360) * (-90 - 45)) * d))
    frmMappa.linVeicolo(0).Y1 = (yVeicolo + (Sin((6.28318530717959 / 360) * (-90 - 45)) * d))
    frmMappa.linVeicolo(0).X2 = (xVeicolo + (Cos((6.28318530717959 / 360) * (-45)) * d))
    frmMappa.linVeicolo(0).Y2 = (yVeicolo + (Sin((6.28318530717959 / 360) * (-45)) * d))
    frmMappa.linVeicolo(1).X1 = (xVeicolo + (Cos((6.28318530717959 / 360) * (-90 - 45)) * d))
    frmMappa.linVeicolo(1).Y1 = (yVeicolo + (Sin((6.28318530717959 / 360) * (-90 - 45)) * d))
    frmMappa.linVeicolo(1).X2 = (xVeicolo + (Cos((6.28318530717959 / 360) * (90)) * d))
    frmMappa.linVeicolo(1).Y2 = (yVeicolo + (Sin((6.28318530717959 / 360) * (90)) * d))
    frmMappa.linVeicolo(2).X1 = (xVeicolo + (Cos((6.28318530717959 / 360) * (90)) * d))
    frmMappa.linVeicolo(2).Y1 = (yVeicolo + (Sin((6.28318530717959 / 360) * (90)) * d))
    frmMappa.linVeicolo(2).X2 = (xVeicolo + (Cos((6.28318530717959 / 360) * (-45)) * d))
    frmMappa.linVeicolo(2).Y2 = (yVeicolo + (Sin((6.28318530717959 / 360) * (-45)) * d))
    No = True
End Sub

Private Sub Form_Resize()
    Dim x, xv, yv As Integer
    xv = xVeicolo
    yv = yVeicolo
    'Calcolo centro mappa
    xVeicolo = (frmMappa.Width - 140) / 2
    yVeicolo = (frmMappa.Height - 900) / 2
    'Sposta gli oggetti sulla mappa
    For x = 1 To frmMappa.linOstacolo.Count - 1
        DoEvents
        frmMappa.linOstacolo(x).X1 = frmMappa.linOstacolo(x).X1 + xVeicolo - xv
        frmMappa.linOstacolo(x).Y1 = frmMappa.linOstacolo(x).Y1 + yVeicolo - yv
        frmMappa.linOstacolo(x).X2 = frmMappa.linOstacolo(x).X2 + xVeicolo - xv
        frmMappa.linOstacolo(x).Y2 = frmMappa.linOstacolo(x).Y2 + yVeicolo - yv
    Next x
    'Posizionamento margini mappa
    frmMappa.shpMargineMappa.Width = frmMappa.Width - 140
    frmMappa.shpMargineMappa.Height = frmMappa.Height - 900
End Sub
```

```

Private Sub Form_Unload(Cancel As Integer)
    frmMain.chkVisualizzaMappa.Value = 0
End Sub

Private Sub StampaMappa_Click()
    On Error GoTo Errore
    frmMappa.PrintForm
    Exit Sub
Errore:
    MsgBox "Impossibile stampare!", vbCritical, "ERRORE!"
    Printer.KillDoc
End Sub

```

### Commento form mappa:

Il form mappa si occupa solo di visualizzare gli oggetti che in form principale crea. In oltre questo form sposta gli oggetti al suo interno in modo che essi siano sempre centrati nel form.

### Listato modulo:

```

'Variabili globali
Public xVeicolo, yVeicolo, Zoom As Double

```

### Commento modulo:

Il modulo contiene solamente 3 variabili che in form principale ed il form mappa utilizzano contemporaneamente.