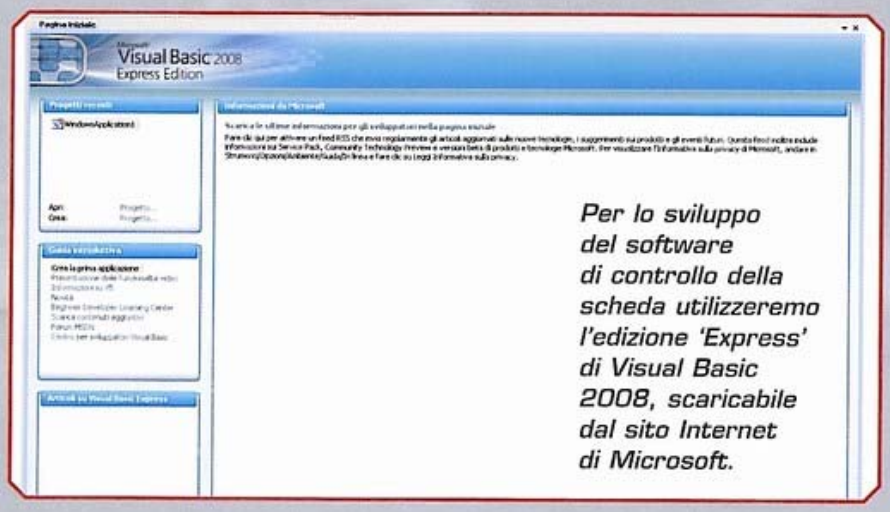
CREIAMO UN'INTERFACCIA DI CONTROLLO

In questo Workshop vediamo come realizzare un'interfaccia grafica di controllo da utilizzare assieme alla mini scheda per servocomandi che abbiamo progettato nei fascicoli precedenti.

Finora abbiamo 'comunicato' con i PIC ricorrendo a emulatori di terminale come Hyperterminal o PuTty. Sebbene tali software siano realmente molto versatili, è fuori discussione che la loro struttura non li renda particolarmente 'semplici' o 'comodi' da utilizzare, specialmente se li confrontiamo con altri tipi di applicazioni caratterizzate da interfacce grafiche più intuitive ed evolute.



Per lo sviluppo del software di controllo della scheda utilizzeremo l'edizione 'Express' di Visual Basic 2008, scaricabile dal sito Internet di Microsoft.

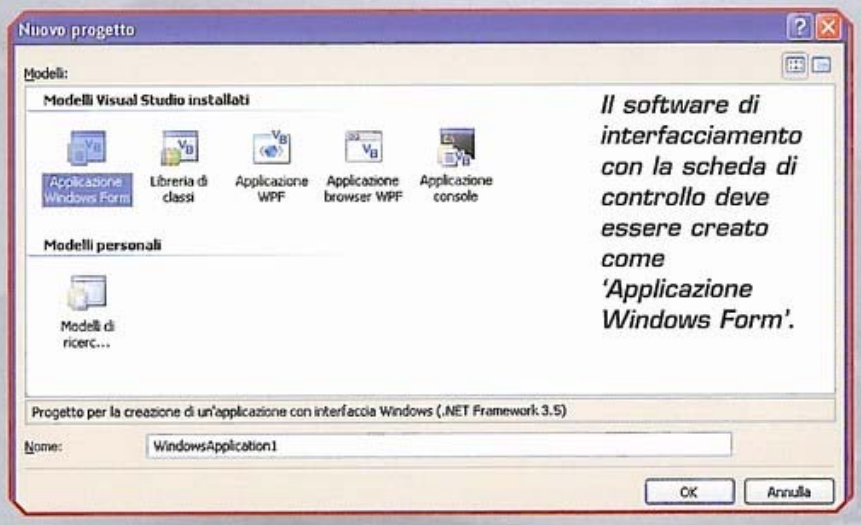
RISERVATO AI PIÙ ESPERTI ›››

In queste pagine viene proposto un mini progetto software che richiede la conoscenza di base dell'ambiente di sviluppo Visual Basic di Microsoft. Poiché tale linguaggio non viene trattato a livello didattico in quest'opera, per proseguire è necessario avere un minimo di dimestichezza con tale sistema e possedere basi di programmazione a oggetti e di utilizzo degli ambienti RAD.

Vediamo, allora, come realizzare un semplice software che implementa una comoda interfaccia grafica per controllare la nostra scheda.

PREPARIAMO IL PROGETTO ›››

Per realizzare il software di interfacciamento con la mini scheda di controllo per servocomandi utilizzeremo la versione 'Express' di Visual Basic 2008 (vedi immagine sopra) scaricabile dal sito web di Microsoft. Poiché l'insegnamento del linguaggio Visual Basic non è parte di quest'opera, ci limiteremo a presentare il codice sorgente e a illustrare la struttura del progetto, affidando il resto delle problematiche alle conoscenze dei lettori. La versione 'eseguibile' dell'applicazione sarà comunque scaricabile dal sito web dell'opera, in modo da renderla utilizzabile per tutti i lettori. Per cominciare, crea un nuovo progetto utilizzando come modello di base il 'Windows Form' (vedi immagine in alto a pagina 2) e imposta le caratteristiche specifiche (nome, percorso di salvataggio ecc.) a tuo piacimento.



Il software di interfacciamento con la scheda di controllo deve essere creato come 'Applicazione Windows Form'.

Nell'immagine vedi mostrata l'interfaccia grafica del software di controllo che stiamo realizzando. Per ogni controllo è mostrato il nome assegnato e la classe dell'oggetto.

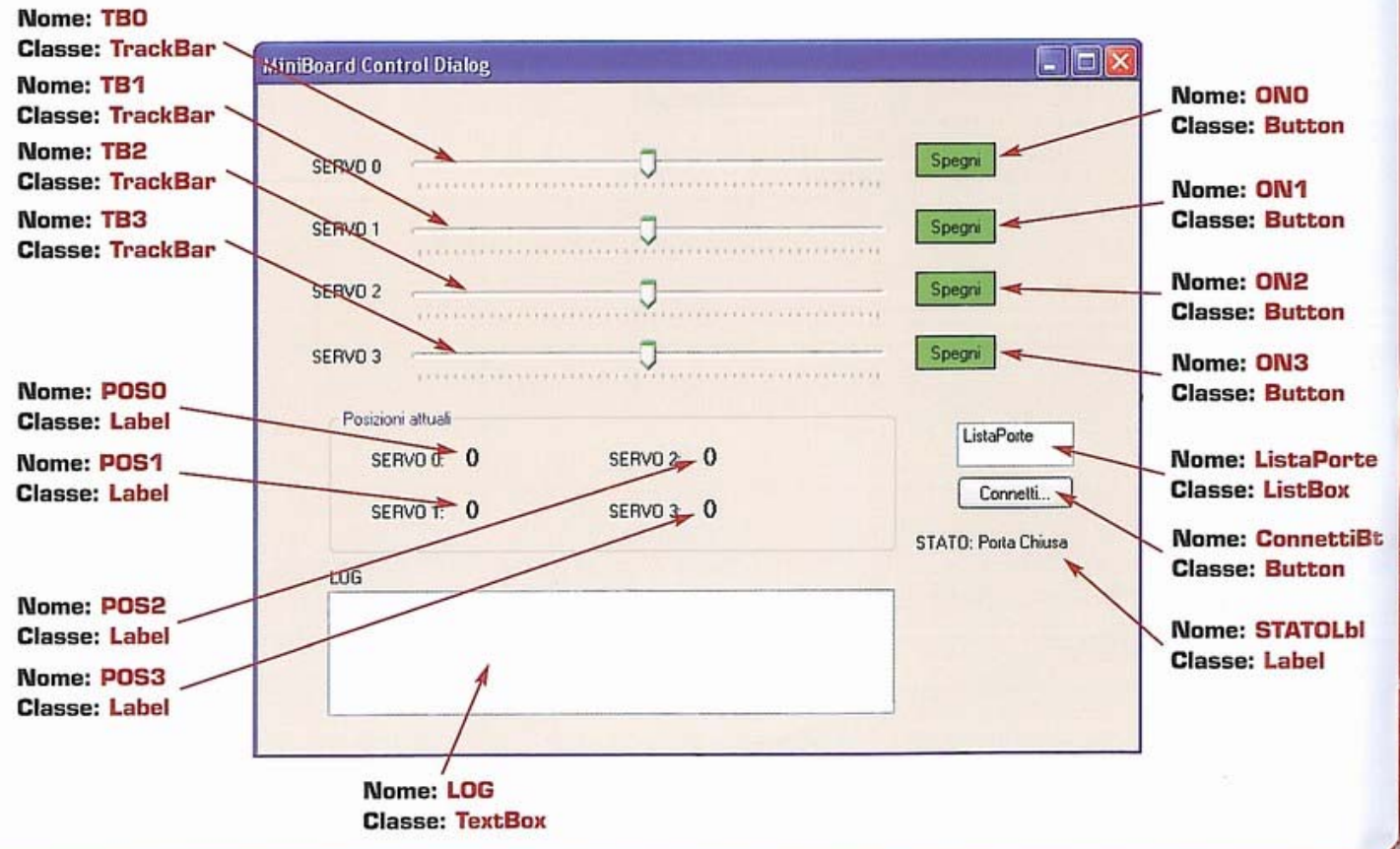
L'INTERFACCIA GRAFICA

Iniziamo, prima di ogni altra cosa, con il 'disegno' dell'interfaccia grafica. Nell'immagine sottostante ti mostriamo la struttura dell'interfaccia grafica che dovrai realizzare (nella figura sono indicate le proprietà 'Name' dei singoli controlli e la loro 'classe', in modo da poter riprodurre una finestra di dialogo 'compatibile' con il codice mostrato nelle pagine successive). Ovviamente, la disposizione dei controlli e le loro proprietà 'visive' possono essere liberamente modificate. Per tutti i valori di proprietà che richiedono valori ben precisi, invece, fai riferimento ai paragrafi successivi.

LE IMPOSTAZIONI DI BASE DEI CONTROLLI

Vediamo come impostare i singoli controlli. Partiamo dalle quattro barre di scorrimento (controlli di classe

L'INTERFACCIA GRAFICA



LA CONFIGURAZIONE DELLE TRACKBAR

PROPRIETÀ	VALORE
Name	TB0., TB1, TB2, TB3 (in base al controllo,come mostrato nell'interfaccia)
Enabled	False
LargeChange	1
Maximum	255
Minimum	1
TickFrequency	5
Value	128

LA CONFIGURAZIONE DEI PULSANTI DI ACCENSIONE

PROPRIETÀ	VALORE
Name	ON0, ON1, ON2, ON3 (in base al controllo, come mostrato nell'interfaccia)
BackColor	Lime
FlatStyle	Flat
Text	Spegni

System.Windows.Forms. .TrackBar istanziati con nomi TB0, TB1, TB2 e TB3) che ci permetteranno di impostare la posizione dei servocomandi. Le proprietà da configurare per questi controlli sono mostrate nella prima tabella in alto. L'impostazione dell'attributo Enabled con il valore False ci permette di rendere inattive le barre di scorrimento all'avvio dell'applicazione. Gli altri valori numerici stabiliscono, invece, il range numerico di scorrimento delle barre, che corrisponde a quello di movimento dei servo.

I BOTTONI DI ACCENSIONE DEI SERVO ›››

Sulla destra delle barre di scorrimento sono collocati quattro bottoni (controlli di classe System.Windows.Forms.Button chiamati ON0, ON1, ON2 e ON3), i quali consentono di controllare l'accensione e lo spegnimento dei servocomandi. Anche per questi componenti mostriamo i valori delle proprietà da configurare, mostrati nella seconda tabella dall'alto.

LE PORTE SERIALI ›››

Il controllo ListaPorte contiene i nomi delle porte COM rilevate all'avvio del programma. È un controllo di classe System. .Windows.Forms.ListBox con nome ListaPorte. Al di sotto di essa vi è il pulsante per l'avvio dell'apertura della porta seriale selezionata. Come per i controlli di accensione dei servo, anche questo pulsante è di tipo System.Windows.Forms.Button. Va creato con nome 'ConnettiBt' e proprietà 'Text' uguale a 'Connetti'. Tutte le operazioni di gestione della connessione seriale avverranno via software e saranno esplicitamente implementate in Visual Basic. Sotto il bottone trovi, infine, il controllo con nome 'StatoLbl' (classe System.Windows.Form.Label) che abbiamo inizializzato con la proprietà Text="STATO: Porta Chiusa" e con la proprietà di allineamento 'TextAlign' di valore 'MiddleRight'. In questa 'etichetta' testuale è visualizzato lo stato della porta seriale.

LA VISUALIZZAZIONE DEGLI STATI ›››

La nostra applicazione contiene anche alcuni controlli di visualizzazione dello stato del sistema e dei dati ricevuti dalla porta seriale. La posizione dei motori è mostrata in tempo reale tramite i controlli di classe System.Windows.Form.Label chiamati Pos0, Pos1, Pos2 e Pos3, dei quali è impiegata la proprietà 'Text' per visualizzare numericamente la posizione delle trackbar. Il controllo di log seriale (nome LOG e classe System.Windows.Forms. .TextBox) è un normale campo di testo editabile impostato in modo da visualizzare testi in modalità 'multilinea' (proprietà 'Multiline' uguale a True). Tramite questo controllo potremo visualizzare costantemente i dati ricevuti dalla scheda. Ovviamente, dovendo utilizzare le porte COM, è necessario aggiungere anche un componente SerialPort, a cui daremo nome 'Serial'.

Il sorgente Visual Basic

```
Public Class Form1
    Private Delegate Sub AggiornaLOGDelegate(ByVal controllo As Control, ByVal Testo as String)
    Private statoporta As Integer  'stato della porta seriale
    Private statoS0 As Boolean     'stato del Servo0 (True: acceso, False: spento)
    Private statoS1 As Boolean     'stato del Servo1 (True: acceso, False: spento)
    Private statoS2 As Boolean     'stato del Servo2 (True: acceso, False: spento)
    Private statoS3 As Boolean     'stato del Servo3 (True: acceso, False: spento)

    Private Sub AggiornaLOG(ByVal controllo As Control, ByVal Testo as String)
        If Me.InvokedRequired Then
         Me.Invoke(New AggiornaLOGDelegate(AddressOf AggiornaLOG),
                   New Object(){controllo,Testo})
         Return
        End If
        controllo.Text = controllo.Text & Testo
    End Sub

    'inizializzazione della finestra che risponde all'evento di sistema Form Load.
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
            MyBase.Load
        Dim ElencoPorte As String()  'vettore di stringhe che contiene le porte rilevate
        ElencoPorte = Serial.GetPortNames()      'copia nel vettore ElencoPorte l'elenco delle
            'seriali rilevate, acquisite chiamando il metodo GetPortNames dell'oggetto Serial.

        Dim i As Integer                         'variabile i di tipo integer
        Dim lunghezza = ElencoPorte.Length       'numero di porte rilevate
        'Inseriamo nella ListBox ListaPorte i nomi delle porte seriali rilevate
        While i < lunghezza
            ListaPorte.Items.Insert(i, ElencoPorte(i))
            i = i + 1
        End While
        'inizializziamo a 128 (valore di centratura) le trackbar
        TB0.Value = 128
        TB1.Value = 128
        TB2.Value = 128
        TB3.Value = 128
         'inizializziamo gli stati della porta e dei servocomandi
        statoporta = 0
        statoS0 = True
        statoS1 = True
        statoS2 = True
        statoS3 = True
         'copiamo nella label di visualizzazione delle posizioni i valori delle trackbar
        Pos0.Text = TB0.Value
        Pos1.Text = TB1.Value
        Pos2.Text = TB2.Value
        Pos3.Text = TB3.Value
    End Sub

        'Risposta all'evento 'Click' sul pulsante ConnettiBt
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ConnettiBt.Click

     'avviamo l'intercettazione delle eccezioni
     Try
            'se la porta è chiusa allora procediamo alla sua apertura e cambiamo il testo
            'della label di stato e del bottone
           If (statoporta = 0) Then
               Serial.PortName = ListaPorte.Text
               Serial.Open()
               ConnettiBt.Text = "Disconnetti..."
               statoporta = 1
               STATOLbl.Text = "STATO: Porta Aperta"
               'abilitiamo le TrackBar
               TB0.Enabled = True
               TB1.Enabled = True
               TB2.Enabled = True
               TB3.Enabled = True
           Else 'se la porta è aperta procediamo alla sua chiusura e cambiamo il testo
                'della label di stato e del bottone
               Serial.Close()
               statoporta = 0
               STATOLbl.Text = "STATO: Porta Chiusa"
               ConnettiBt.Text = "Connetti..."

               'disabilitiamo le TrackBar
               TB0.Enabled = False
               TB1.Enabled = False
               TB2.Enabled = False
               TB3.Enabled = False
```

```
            End If

        'Nel caso in cui venisse generata un'eccezione visualizziamo il messaggio
        'di errore in una messagebox
        Catch ex As Exception
            MsgBox("Attenzione, devi selezionare una porta seriale.")
        End Try
    End Sub

    'risposta al click sul pulsante ON0
    Private Sub ON0_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ON0.Click
        If statoS0 Then  'Se il servo 0 è acceso invia il comando seriale di spegnimento
                         ''o0' e modifica il pulsante
            ON0.Text = "Accendi"
            Serial.Write("o0" & vbCr)
            statoS0 = False
            ON0.BackColor = Color.Red
        Else   'altrimenti, se è spento  invia il comando seriale di
               'accensione 'O0' e modifica il pulsante
            ON0.Text = "Spegni"
            Serial.Write("O0" & vbCr)
            statoS0 = True
            ON0.BackColor = Color.Lime
        End If
    End Sub

    'risposta al click sul pulsante ON1
    Private Sub ON1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ON1.Click
        If statoS1 Then 'Se il servo 1 è acceso invia il comando seriale di spegnimento
                        ''o1' e modifica il pulsante
            ON1.Text = "Accendi"
            Serial.Write("o1" & vbCr)
            statoS1 = False
            ON1.BackColor = Color.Red
        Else  'altrimenti, se è spento  invia il comando seriale di accensione 'O1'
              'e modifica il pulsante
            ON1.Text = "Spegni"
            Serial.Write("O1" & vbCr)
            statoS1 = True
            ON1.BackColor = Color.Lime
        End If
    End Sub

    'risposta al click sul pulsante ON2
    Private Sub ON2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ON2.Click
        If statoS2 Then 'Se il servo 2 è acceso invia il comando seriale di spegnimento
                        ''o2' e modifica il pulsante
            ON2.Text = "Accendi"
            Serial.Write("o2" & vbCr)
            statoS2 = False
            ON2.BackColor = Color.Red
        Else  'altrimenti, se è spento  invia il comando seriale di accensione
              ''O2' e modifica il pulsante
            ON2.Text = "Spegni"
            Serial.Write("O2" & vbCr)
            statoS2 = True
            ON2.BackColor = Color.Lime
        End If
    End Sub

    'risposta al click sul pulsante ON3
    Private Sub ON3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ON3.Click
        If statoS3 Then 'Se il servo 3 è acceso invia il comando seriale di spegnimento
                        ''o3' e modifica il pulsante
            ON3.Text = "Accendi"
            Serial.Write("o3" & vbCr)
            statoS3 = False
            ON3.BackColor = Color.Red
        Else  'altrimenti, se è spento  invia il comando seriale di accensione
              ''O3' e modifica il pulsante
            ON3.Text = "Spegni"
            Serial.Write("O3" & vbCr)
            statoS3 = True
            ON3.BackColor = Color.Lime
        End If
    End Sub


    'risposta all'evento di rilascio del pulsante del mouse sulla trackbar TB0
    Private Sub TB0_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles TB0.MouseUp
        Dim comando As String
```

```
        comando = "M0"     'radice del comando da inviare
        If TB0.Value < 10 Then  'se il valore indicato sulla trackbar è inferiore a 10
            comando = comando & "00" & TB0.Value & vbCr  'concatena al comando 2 zeri e il
                                                         'valore della trackbar
        ElseIf TB0.Value < 100 Then  'altrimenti se il valore indicato sulla trackbar
                                     'è inferiore a 100
            comando = comando & "0" & TB0.Value & vbCr   'concatena al comando uno zero
                                                         'e il valore della trackbar
        Else 'altrimenti
            comando = comando & TB0.Value & vbCr  'concatena al comando il valore
                                                  'della trackbar
        End If
        Serial.Write(comando)    'invia il comando appena composto tramite la porta seriale
        Pos0.Text = TB0.Value    'aggiorna la label di posizionamento del servo 0
    End Sub

     'risposta all'evento di rilascio del pulsante del mouse sulla trackbar TB1
    Private Sub TB1_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles TB1.MouseUp
        Dim comando As String
        comando = "M1"   'radice del comando da inviare
        If TB0.Value < 10 Then  'se il valore indicato sulla trackbar è inferiore a 10
            comando = comando & "00" & TB1.Value & vbCr  'concatena al comando 2 zeri e il
                                                         'valore della trackbar
        ElseIf TB0.Value < 100 Then  'altrimenti se il valore indicato sulla trackbar è
                                     'inferiore a 100
            comando = comando & "0" & TB1.Value & vbCr  'concatena al comando 1 zero
                                                        'e il valore della trackbar
        Else 'altrimenti
            comando = comando & TB1.Value & vbCr 'concatena al comando il valore
                                                 'della trackbar
        End If
        Serial.Write(comando) 'invia il comando appena composto tramite la porta seriale
        Pos1.Text = TB1.Value  'aggiorna la label di posizionamento del servo 1
    End Sub

    'risposta all'evento di rilascio del pulsante del mouse sulla trackbar TB2
    Private Sub TB2_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles TB2.MouseUp
        Dim comando As String
        comando = "M2"    'radice del comando da inviare
        If TB0.Value < 10 Then  'se il valore indicato sulla trackbar è inferiore a 10
            comando = comando & "00" & TB2.Value & vbCr  'concatena al comando 2 zeri
                                                         'e il valore della trackbar
        ElseIf TB0.Value < 100 Then  'altrimenti se il valore indicato sulla trackbar
                                     'è inferiore a 100
            comando = comando & "0" & TB2.Value & vbCr 'concatena al comando 1 zero
                                                       'e il valore della trackbar
        Else 'altrimenti
            comando = comando & TB2.Value & vbCr 'concatena al comando il valore
                                                 'della trackbar
        End If
        Serial.Write(comando)  'invia il comando appena composto tramite la porta seriale
        Pos2.Text = TB2.Value  'aggiorna la label di posizionamento del servo 2
    End Sub

    'risposta all'evento di rilascio del pulsante del mouse sulla trackbar TB3
    Private Sub TB3_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles TB3.MouseUp
        Dim comando As String
        comando = "M3"    'radice del comando da inviare
        If TB0.Value < 10 Then  'se il valore indicato sulla trackbar è inferiore a 10
            comando = comando & "00" & TB3.Value & vbCr  'concatena al comando 2 zeri
                                                         'e il valore della trackbar
        ElseIf TB0.Value < 100 Then    'altrimenti se il valore indicato sulla
                                       'trackbar è inferiore a 100
            comando = comando & "0" & TB3.Value & vbCr 'concatena al comando 1 zero e il
                                                       'valore della trackbar
        Else  'altrimenti
            comando = comando & TB3.Value & vbCr  'concatena al comando il valore
                                                  'della trackbar
        End If
        Serial.Write(comando)  'invia il comando appena composto tramite la porta seriale
        Pos3.Text = TB3.Value  'aggiorna la label di posizionamento del servo 3
    End Sub

    'risposta all'evento di ricezione dati sulla porta seriale
    Private Sub Serial_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles Serial.DataReceived
        'finchè sono presenti dati nel buffer seriale li concatena al testo del controllo LOG
        While Serial.BytesToRead
            AggiornaLOG(LOG, "" & Chr(Serial.ReadChar()))
        End While
    End Sub

End Class
```