

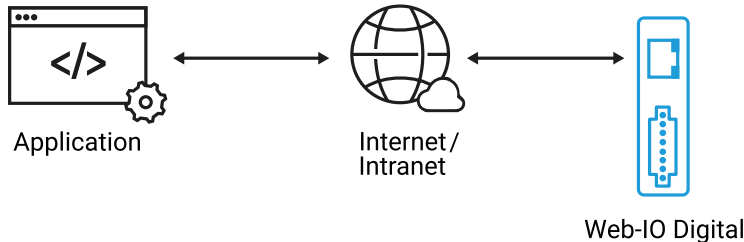
Applicazione al Web-IO digitale:

Web-IO digitale con Delphi 2005 controllo e monitoraggio

Ulteriori applicazioni

Panoramica del prodotto

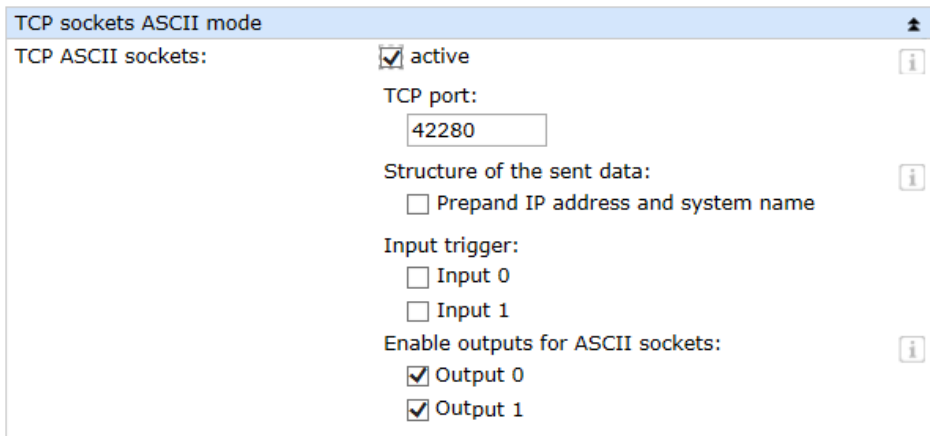
Come successore di Borland Delphi, Delphi 2005 utilizza la struttura Microsoft.net. Delphi 2005 (Delphi.net) offre tutto ciò che è necessario per la programmazione di applicazioni TCP/IP. In tal modo Delphi 2005 è uno strumento ausiliario prediletto per la creazione di applicazioni che comunicano con il [Web-IO digitale](#). Ulteriori driver o DLL non saranno necessari.



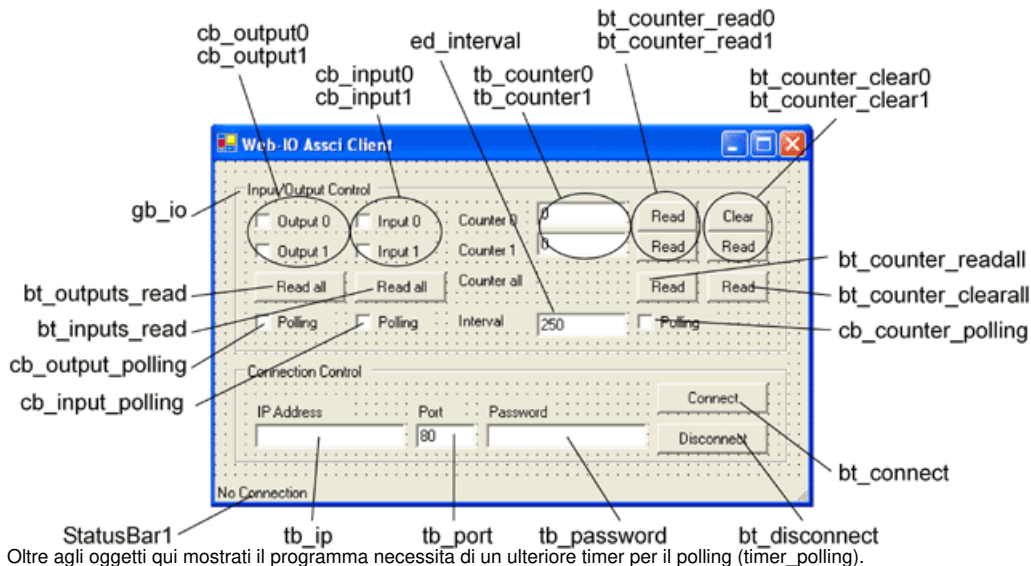
Con il seguente esempio di programma potete riprodurre il vostro Web-IO digitale con i suoi input e output in un'applicazione Windows. Inoltre potete collegare gli output del Web-IO.

Preparativi

- [Mettere sotto tensione il Web-IO e collegare gli IO](#)
- [Collegare il Web-IO alla rete](#)
- [Assegnazione di indirizzi IP](#)
- Nel Web-IO nell'area *Vie di comunicazione* >> *API socket* Attivare i *socket ASCII TCP* e autorizzare gli output per l'attivazione



Collocazione dei diversi elementi di comando e oggetti di visualizzazione nel modulo VB.net



Nella denominazione dei singoli oggetti è utile utilizzare nomi che ne riprendono il significato. In questo esempio la prima parte del nome descrive il tipo dell'oggetto e la seconda parte la funzione.

Avvio del programma

Mentre la costruzione delle interfacce utente grafiche in Delphi 2005 è ugualmente facile da realizzare come con le versioni precedenti di Delphi, la comunicazione di rete sotto Delphi.net risulta leggermente più difficile. Ciò dipende innanzitutto dal fatto che per le applicazioni framework l'elemento di controllo ClientSocket non è più disponibile. Per l'accesso alla rete deve invece essere indicato nell'area uses il namespace per le classi socket utilizzate.

```
uses
  System.Drawing, System.Collections, System.ComponentModel,
  System.Windows.Forms, System.Data, System.Net,
  System.Net.Sockets, System.IO, System.Text, System.Configuration;
```

Le procedure proprie che non vengono create da Delphi stessa devono essere dichiarate nell'intestazione del testo sorgente. (Maggiori informazioni sul contenuto di queste procedure più avanti.)

```
private
  { Private-Deklarationen }
public
  constructor Create;
  procedure callback_connect(ari : IAsyncResult);
  procedure callback_readdata(ari : IAsyncResult);
  procedure sendcommand(sendstring : String);
  procedure close();
end;
```

Inoltre deve essere creato il socket, su cui deve svolgersi la comunicazione, e deve essere definito un buffer per i dati di ingresso nonché una variabile AsyncResult.

```
var
  TCP_Client : Socket;
  ReceiveBuffer : array [0..512] of byte;
  ar : IAsyncResult;
```

Controllo del collegamento

Inizializzazione del collegamento

Immettendo l'indirizzo IP del Web-IO nel campo di testo *tb_ip* e facendo clic sul pulsante *bt_connect* viene avviata la creazione del collegamento

```
procedure TWinForm.bt_connect_Click(sender: System.Object; e: System.EventArgs);
var
  WebioEP : IPEndPoint;
begin
  if (tb_ip.Text <> '') and (tb_port.Text <> '') then
  begin
    bt_connect.Enabled := false;
    TCP_Client := Socket.Create(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.TCP);
    StatusBar1.Text := 'Try to connect to ' + tb_ip.Text;
    timer_polling.Enabled := true;
    try
      TCP_Client.BeginConnect(WebioEP, callback_connect, TCP_Client);
    except
      on ex : Exception
      do
        begin
          StatusBar1.Text := 'ERROR on connecting';
          close();
        end;
      end;
    end;
  end;
end;
```

Creazione del collegamento

Per lo svolgimento della gestione TCP/IP viene innanzitutto definito un IPEndPoint dall'indirizzo IP e dalla porta TCP e con ciò viene inizializzato il socket TCP_client. Durante la richiesta di collegamento viene creato un rimando a una procedura di callback.

Collegamento realizzato

Non appena il Web-IO accetta il collegamento, viene eseguita la procedura di callback. Nella riga di stato viene visualizzata la realizzazione del collegamento, gli elementi di comando vengono abilitati per l'utilizzo e il pulsante Disconnect risulta utilizzabile. Il pulsante Connect viene bloccato per l'uso. Inoltre viene creato un rimando a una routine di callback per la ricezione dei dati.

```
procedure TWinForm.callback_connect(ari : IAsyncResult);
begin
  gb_io.enabled := true;
  bt_disconnect.Enabled := True;
  statusbar1.Text := 'Connected to' + tb_ip.Text;
  try
    TCP_Client.EndConnect(ari);
    TCP_Client.BeginReceive(ReceiveBuffer, 0, 512, SocketFlags.None, callback_readdata, ari.AsyncState);
  except
    on ex : Exception
    do
      begin
        StatusBar1.Text := 'ERROR on connect: ' + ex.ToString;
        close();
      end;
    end;
  end;
end;
```

Disinserzione del collegamento

Il collegamento rimane fino a quando non viene terminato dall'utente facendo clic sul pulsante Disconnect oppure il Web-IO termina il collegamento.

In questo caso viene richiamata una corrispondente procedura.

```
procedure TWinForm.close();
begin
  timer_polling.Enabled := False;
  Try
    TCP_client.EndReceive(ar);
  except
  end;
  Try
    TCP_client.Shutdown(SocketShutdown.Both);
  except
  end;
  Try
    TCP_client.Close()
  except
  end;
  bt_connect.Enabled := True;
  gb_io.Enabled := False;
  bt_disconnect.Enabled := False;
  StatusBar1.Text := 'no connection';
end;

procedure TWinForm.bt_disconnect_Click(sender: System.Object; e: System.EventArgs);
begin
  close();
end;
```

Errore di collegamento

Tutte le azioni che riguardano la comunicazione TCP/IP vengono eseguite nell'ambito dell'istruzione Try. Se compaiono errori, viene richiamata anche la procedura CloseConnection.

Utilizzo e comunicazione della parte client

Non appena viene realizzato un collegamento con il Web-IO, l'utente può inviare comandi al Web-IO utilizzando i corrispondenti elementi del programma.

```
procedure TWinForm.sendcommand(sendstring: String);
var
  SendBuffer : array[0..512] of byte;
begin
  SendBuffer := System.Text.Encoding.ASCII.GetBytes(sendstring);
  try
    TCP_Client.Send(SendBuffer, 0, sendstring.Length, SocketFlags.None );
  except
    on ex : Exception
    do
      begin
        StatusBar1.Text := 'ERROR on send: ' + ex.ToString;
        close();
      end;
    end;
  end;
end;
```

Impostazione degli output

L'impostazione degli output è resa possibile all'utente da due caselle di spunta *cb_outputx*. Il programma utilizza a tale scopo l'evento MouseUP di questo oggetto. Se viene registrato un MouseUp, ossia un abbandono della casella di spunta degli output, il programma esegue la corrispondente procedura e inoltra al Web-IO, in base all'eventuale impostazione della casella di spunta, il comando adatto.

```
procedure TWinForm.cb_output0_MouseUp(sender: System.Object; e: System.Windows.Forms.MouseEventArgs);
begin
  if cb_output0.Checked Then
    sendcommand('GET /outputaccess0?PW='+ tb_password.Text + '&State=ON&')
  else
    sendcommand('GET /outputaccess0?PW='+ tb_password.Text + '&State=OFF&');
end;

procedure TWinForm.cb_output1_MouseUp(sender: System.Object; e: System.Windows.Forms.MouseEventArgs);
begin
  if cb_output1.Checked Then
    sendcommand('GET /outputaccess1?PW='+ tb_password.Text + '&State=ON&')
  else
    sendcommand('GET /outputaccess1?PW='+ tb_password.Text + '&State=OFF&');
end;
```

Interrogazione dello stato degli output/input

L'utente può richiedere lo stato degli output e degli input facendo clic sul relativo pulsante.

```

procedure TWinForm.bt_outputs_read_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /output?PW='+ tb_password.Text + '&');
end;

procedure TWinForm.bt_inputs_read_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /input?PW='+ tb_password.Text + '&');
end;

```

Interrogazione/cancellazione dei counter

È possibile interrogare o cancellare anche gli stati dei counter degli input.

```

procedure TWinForm.bt_counter_read0_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /counter0?PW='+ tb_password.Text + '&');
end;

procedure TWinForm.bt_counter_read1_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /counter0?PW='+ tb_password.Text + '&');
end;

procedure TWinForm.bt_counter_clear0_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /counterclear0?PW='+ tb_password.Text + '&');
end;

procedure TWinForm.bt_counter_clear1_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /counterclear1?PW='+ tb_password.Text + '&');
end;

```

Naturalmente è possibile leggere o cancellare contemporaneamente anche tutti i counter.

```

procedure TWinForm.bt_counter_readall_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /counter?PW='+ tb_password.Text + '&');
end;

procedure TWinForm.bt_counter_clearall_Click(sender: System.Object; e: System.EventArgs);
begin
  sendcommand('GET /counterclear?PW='+ tb_password.Text + '&');
end;

```

Ricezione dei dati dal Web-IO

Analisi e visualizzazione dei dati ricevuti

Tutti i comandi e le richieste al Web-IO vengono confermati con una stringa di risposta. Tali risposte hanno una struttura specifica in base al tipo.

- Per gli output: output;<valore binario dello stato degli output in formato esadecimale>
- Per gli input: input;<valore binario dello stato degli input in formato esadecimale>
- Per i counter: counterx;<stato del conteggio decimale>
- oppure counter;<stato del conteggio decimale 0 >; <stato del conteggio decimale 0 >; ... se tutti i counter devono essere letti in un'unica volta.
- Tutte le stringhe di risposta terminano con 0 byte.

Alla ricezione dei dati viene richiamata la corrispondente procedura di callback

```

procedure TWinForm.callback_readdata(ari : IAsyncResult);
var
  bytesread : Integer;
  receivestring : string;
begin
  try
    bytesread := TCP_Client.EndReceive(ari)
  except
  end;
  if Bytesread = 0 then
    close()
  else
    begin
      receivestring := System.Text.Encoding.ASCII.GetString(ReceiveBuffer,0,bytesread);
      try
        TCP_Client.BeginReceive(ReceiveBuffer,0, 512,SocketFlags.None, callback_readdata,ari.AsyncState);
      except
        on ex : Exception
        do
          begin
            statusbar1.Text :='ERROR on read: ' + ex.ToString;
            close();
          end;
        end;
      end;
    end;
  if receivestring[1] = 'o' then
    begin
      if(convert.ToInt16(receivestring[8]) and 1) = 1 then
        cb_output0.Checked := true
      else
        cb_output0.Checked := false;
      if(convert.ToInt16(receivestring[8]) and 2) = 2 then
        cb_output1.Checked := true
      else
        cb_output1.Checked := false;
    end;
  if receivestring[1] = 'i' then
    begin
      if(convert.ToInt16(receivestring[7]) and 1) = 1 then
        cb_input0.Checked := true
      else
        cb_input0.Checked := false;
      if(convert.ToInt16(receivestring[7]) and 2) = 2 then
        cb_input1.Checked := true
      else
        cb_input1.Checked := false;
    end;
  if receivestring[1] = 'c' then
    begin
      if copy(ReceiveString, 8, 1) = '0' then
        tb_counter0.Text := copy(ReceiveString, 10,length(ReceiveString)-10);
      if copy(ReceiveString, 8, 1) = '1' then
        tb_counter1.Text := copy(ReceiveString, 10,length(ReceiveString)-10);
      if copy(ReceiveString, 8, 1) = ';' then
        begin
          ReceiveString[8] := ' ';
          tb_counter0.Text := copy(ReceiveString, 9, pos(';',ReceiveString)-9);
          tb_counter1.Text := copy(ReceiveString,pos(';',ReceiveString)+1,length(ReceiveString)-pos(';',ReceiveString)-1);
        end;
    end;
  end;
end;
end;
end;

```

La procedura di ricezione controlla sulla base del primo carattere dei dati ricevuti se si tratta di messaggi degli input, degli output o dei counter. In base a ciò viene ad es. stabilito quale stato ha quale output. Nei counter è possibile sia interrogare i valori dei singoli contatori che leggere tutti i counter contemporaneamente. Gli stati dei singoli contatori vengono visualizzati in una stringa in decimali separati mediante punto e virgola.

Polling

Interrogazione ciclica di determinati valori

Per permettere anche un aggiornamento automatico della visualizzazione, viene utilizzato un timer.

In base alle caselle di spunta per il polling degli output, degli input e dei counter le corrispondenti informazioni vengono interrogate dal Web-IO nell'intervallo impostato.

```

procedure TWinForm.timer_polling_Tick1(sender: System.Object; e: System.EventArgs);
begin
  if TCP_Client.Connected then
    begin
      if cb_output_polling.Checked then
        sendcommand('GET/output?PW=' + tb_password.Text + '&');
      if cb_input_polling.Checked then
        sendcommand('GET/input?PW=' + tb_password.Text + '&');
      if cb_counter_polling.Checked then
        sendcommand('GET/counter?PW=' + tb_password.Text + '&');
    end;
  end;
end;

```

L'intervallo desiderato può essere immesso nel corrispondente campo di testo. In caso di una modifica l'intervallo del timer viene adattato automaticamente.

```
procedure TWinForm.tb_interval_TextChanged(sender: System.Object; e: System.EventArgs);
begin
  timer_polling.Interval := convert.ToInt16(tb_interval.Text);
end;
```

Il [programma esempio](#) supporta tutte le comuni funzioni del Web-IO nella modalità stringa di comando, ottimizzata per [iWeb-IO 2x input digitale, 2x output digitale](#). Per gli altri modelli di Web-IO devono eventualmente essere eseguiti adattamenti al programma. Ulteriori esempi di programma per la programmazione socket sono riportati nelle [pagine dei tool](#) per il Web-IO. Una descrizione dettagliata sull'interfaccia socket dei modelli Web-IO digitali è riportata nel [manuale di riferimento](#).

[↓ Download esempio di programma](#)

Prodotti



Web-IO 4.0 digitale
2x input, 2x output

All'occorrenza possibilità di
alimentazione anche tramite PoE



Web-IO 4.0 digitale
12x input, 12x output

12x ingressi,
12x uscite



Altri web-IO

Tutti i Web-IO digitali W&T da 24 V

W&T
www.WuT.de

[Saremo lieti di fornirvi una consulenza personalizzata!](#)

Wiesemann & Theis
GmbH
Porschestr. 12
42279 Wuppertal
Tel.: +49 202/2680-110 (Lun-Ven. 8-17)
Fax: +49 202/2680-265
info@wut.de

© Wiesemann & Theis GmbH, con riserva di errori e modifiche: poiché possono verificarsi errori, nessuna nostra informazione deve essere utilizzata senza essere stata verificata. Vi preghiamo di comunicarci tutti gli errori o gli equivoci che avete rilevato in modo tale che possiamo riconoscerli ed eliminarli quanto prima.

[Protezione dei dati](#)