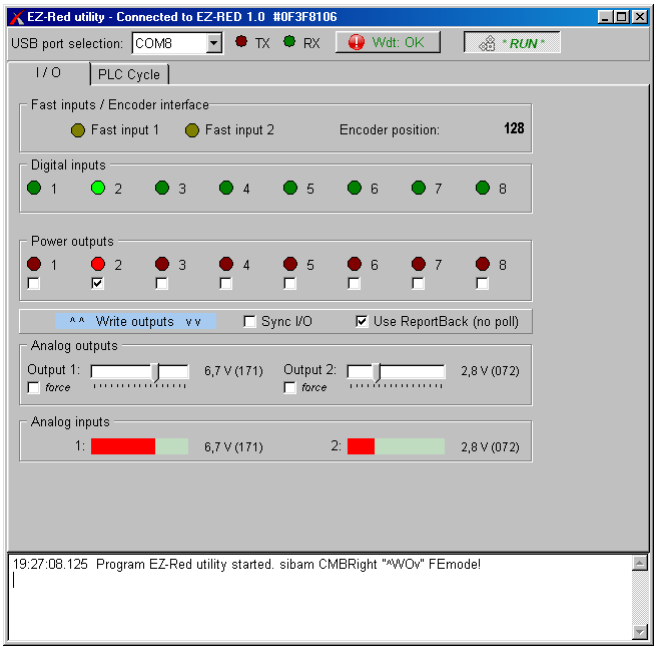
EZ-Red - Modulo I/O di potenza

Manuale del programma di supporto TSmon

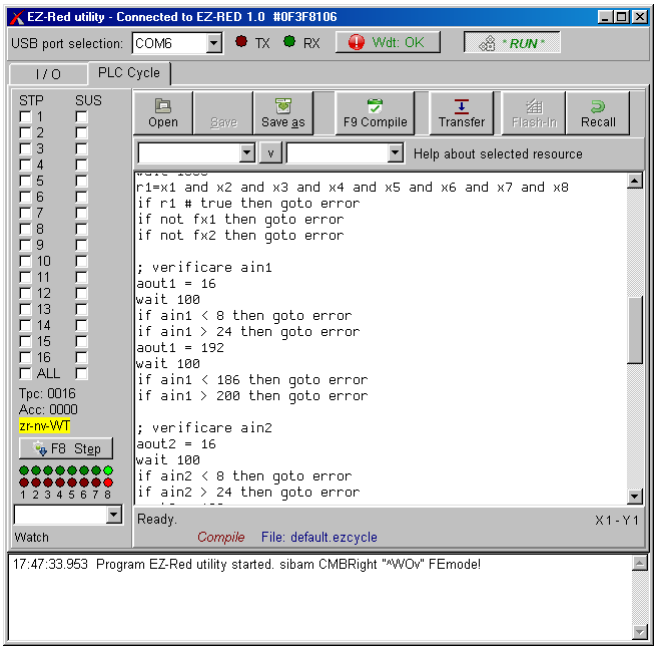
Indice

Introduzione

Il programma TSmon serve per controllare e sperimentare il modulo EZ-Red, e per compilare i cicli PLC e memorizzarli nel modulo. E' composto da una finestra principale che contiene due pagine; la prima pagina:



permette di vedere lo stato degli ingressi e d'impostare le uscite. La seconda pagina:



permette di scrivere e verificare un ciclo PLC, inviarlo al modulo e memorizzarlo in modo permanente.

Per funzionare, TSmon ha bisogno della libreria *DLL* **ezreddll.dll**, fornita a corredo. Redazione e compilazione di cicli PLC sono possibili anche senza collegare un modulo EZ-Red; per tutte le altre funzioni è, naturalmente, necessario che il modulo sia collegato.

Installazione

Driver FTDI

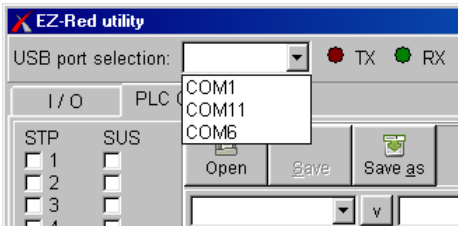
Per prima cosa occorre installare i driver FTDI, reperibili sul sito del produttore o dal sito di XON Electronics nella pagina del prodotto EZ-Red. Riferirsi al manuale d'uso per maggiori dettagli.

Programma TSmon

Copiare i file TSMON.EXE ed EZREDDLL.DLL in una cartella a scelta. Dentro questa cartella verrà creato un file TSMON.INI contenente alcune preferenze, dopo la prima esecuzione del programma.

Interfaccia USB

Per instaurare la comunicazione, occorre alimentare un EZ-Red e collegarlo tramite l'interfaccia USB al computer; poi eseguire **TSmon.exe**, infine scegliere la porta di comunicazione tra quelle disponibili:

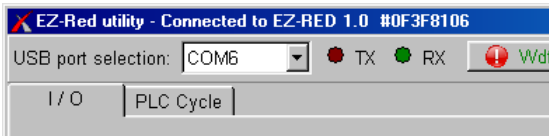


La casella a scomparsa di "USB port selection" elenca le porte seriali disponibili. Questa lista viene aggiornata automaticamente al momento dell'apertura (clic su triangolo): se il modulo EZ-Red viene collegato (o alimentato) quando il programma è già avviato, chiudere e riaprire la casella a scomparsa.

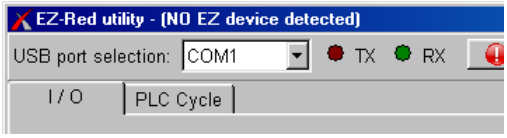
Se EZ-Red è alimentato e collegato, e la porta USB è configurata correttamente (riferirsi al manuale di EZ-Red), le due spie colorate TX ed RX dell'immagine qui sopra cominciano a lampeggiare. Se questo non succede, controllare:

1. EZ-Red è alimentato e collegato a una porta USB funzionante?
2. I driver FTDI sono stati installati?
3. Il sistema operativo rileva il nuovo hardware?
4. Le opzioni di FTDI sono impostate correttamente?

E' anche possibile provare una per una le porte COM elencate nella casella; dopo aver scelto una porta il programma cerca di instaurare la comunicazione; se il dispositivo viene rilevato, la finestra mostra nel titolo l'avvenuta connessione, e versione e numero di serie del modulo collegato:

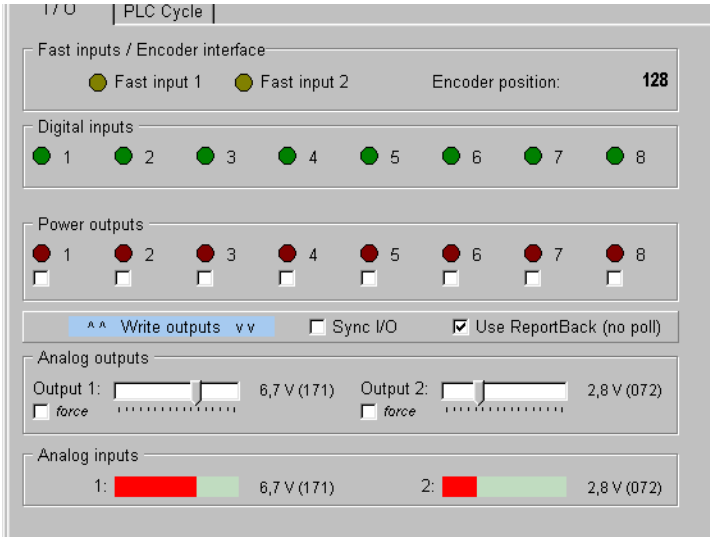


Diversamente, il messaggio “NO EZ device detected” viene mostrato:



Finestra di I/O

La finestra di I/O mostra lo stato di ingressi e uscite:



Vi sono alcuni riquadri:

- Fast inputs / Encoder interface.
  Mostra lo stato dei due ingressi veloci, e la posizione dell'encoder ad essi eventualmente collegato.
  Se c'è tensione su un ingresso veloce, il colore della spia passa da giallo spento a brillante.
- Digital inputs.
  Mostra lo stato degli otto ingressi. Il colore è verde scuro in assenza di tensione, e verde brillante quando l'ingresso è "ON" (a uno logico).
- Power outputs.
  Mostra, con rosso scuro e rosso brillante, lo stato delle uscite. Sotto ogni spia c'è una casella che si può spuntare per attivare l'uscita corrispondente, o deselezionare per spegnerla.
- Analog outputs.
  Per ognuna delle due uscite analogiche il cursore mostra il valore di tensione in modo visuale, oltre che numerico e in volt effettivi. Nell'illustrazione precedente, l'uscita 1 ha il valore 175, corrispondente a una tensione di 6,8 volt. Il cursore può essere spostato per impostare il livello d'uscita desiderato; in tal caso, la casella "*force*" ("forzare") si marca automaticamente.
- Analog inputs.
  I due ingressi analogici si comportano in modo identico alle uscite analogiche; vengono mostrate con una barra invece che con un cursore, perché non è possibile manipolarle con il computer.

La casella "Use ReportBack (no poll)" indica il tipo di dialogo fra computer ed EZ-Red. Quando non è marcata, TSmon interroga EZ-Red ciclicamente ogni 50 millisecondi circa; variazioni di stato che accadono tra una interrogazione e l'altra non vengono rilevati, e quindi in alcuni casi questo tipo di interrogazione (polling) non è ottimale. Quando la casella è spuntata, EZ-Red “avverte” il programma TSmon che c'è stata una variazione di ingressi o uscite, e il programma mostra le variazioni immediatamente.

La casella "Sync I/O" permette d'impostare il modo di aggiornamento degli I/O (riferirsi al manuale di programmazione); modificare questa impostazione da computer può interferire con il ciclo PLC.

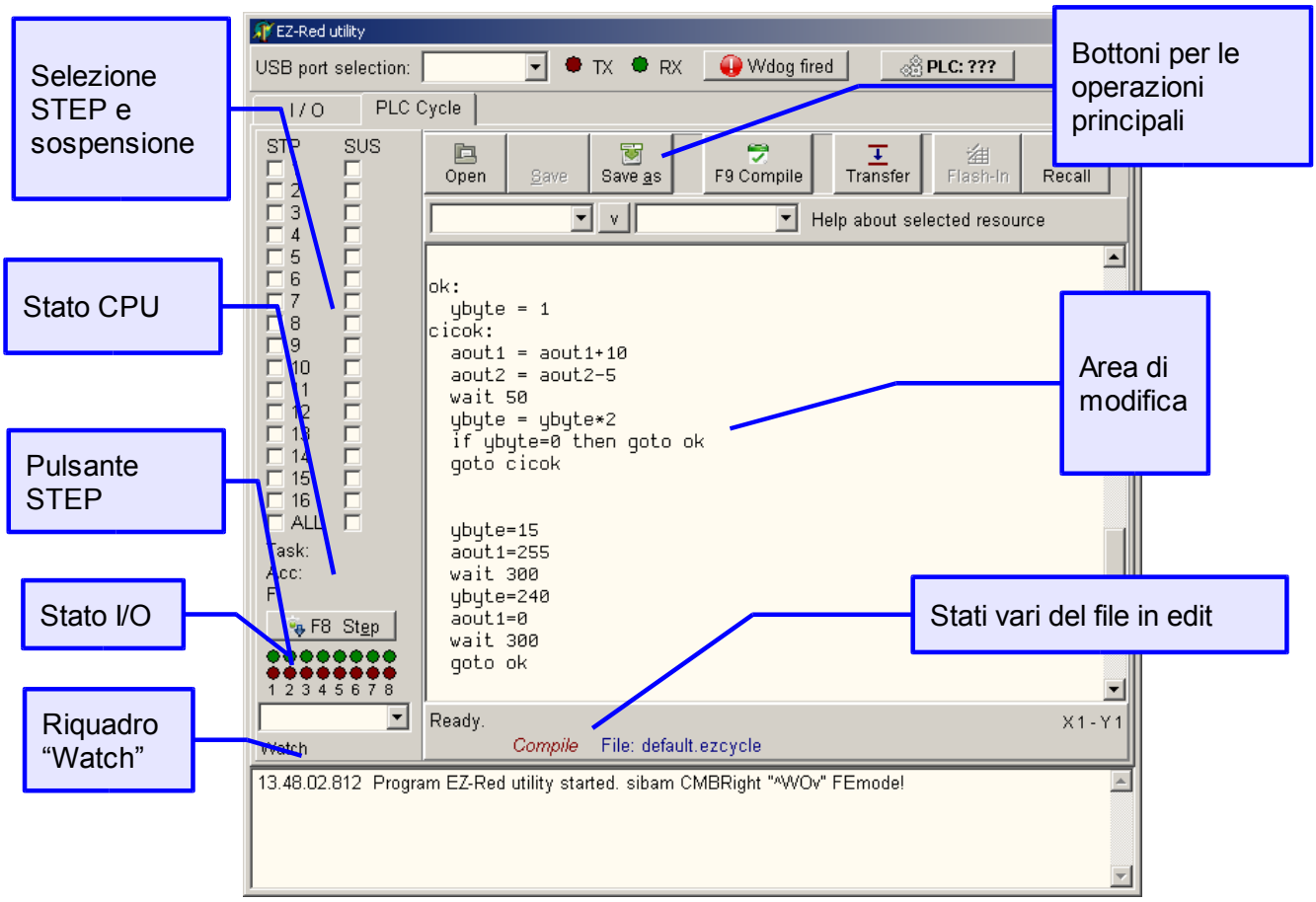
Tutti i controlli presenti su questa scheda possono essere modificati dall'eventuale ciclo PLC in esecuzione sul modulo; quando questo accade, è normale che si verifichino interferenze con l'uso interattivo tramite computer.

Finestra del Ciclo PLC

Introduzione

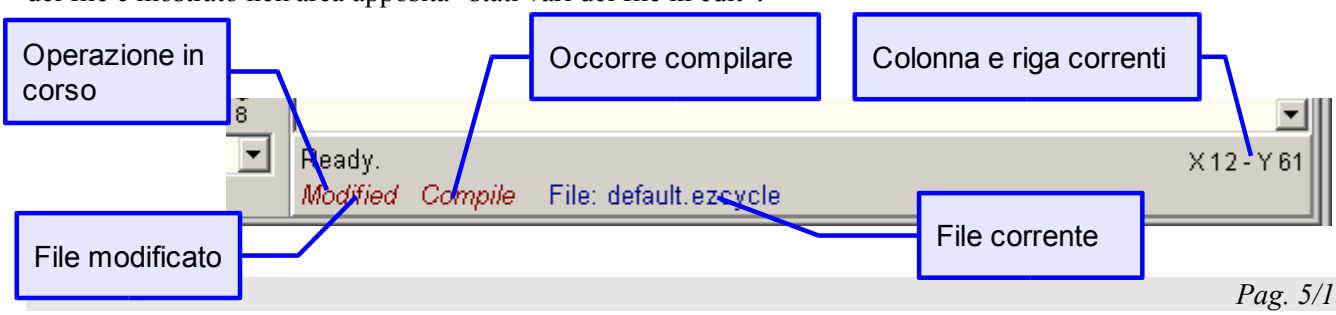
In questa finestra si trova quanto occorre per scrivere un ciclo PLC, trasferirlo al modulo, eseguirlo per cercare gli errori, e memorizzarlo in modo permanente nella memoria non volatile (*flash*) di EZ-Red.

Questo manuale non documenta il linguaggio di programmazione – occorre fare riferimento al Manuale di programmazione di EZ-Red.

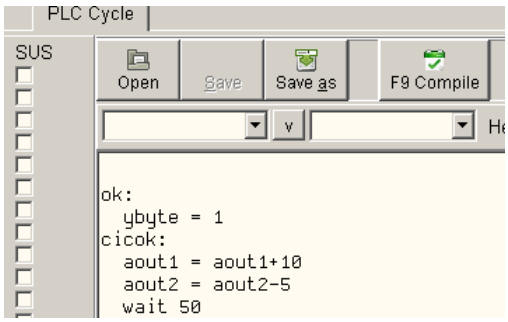


Archivi su disco

I programmi (cicli PLC) possono essere caricati da disco e salvati sullo stesso. Questi file sono file di testo semplice, con estensione **.ezCycle**; all'avviamento di TSmon, se presente, il file "default.ezCycle" viene automaticamente caricato. Tutte le operazioni vengono effettuate sul file mostrato nell'area di modifica, e lo stato del file è mostrato nell'area apposita "stati vari del file in edit":



La barra dei bottoni ne contiene tre per caricare (aprire) e salvare il file in edit:



Il bottone “Open” serve per caricare un file da disco. Se il file correntemente mostrato nella finestra è stato modificato, viene richiesta conferma per annullare le modifiche.

Il bottone “Save” permette di salvare il file su disco, con il suo nome attuale; è attivo solo se il file corrente è stato modificato.

Il bottone “Save as” permette di salvare il file corrente con un nome diverso.

Area di modifica

L'area di modifica è la finestra bianca dove si scrive il testo del programma. Le operazioni di editing sono quelle normali di qualunque editor di testo.

Cliccando con il tasto destro compare un menù con tre titoli:

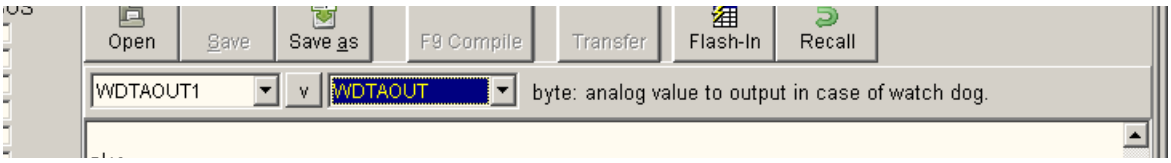
- Change font...
  Consente di scegliere un tipo di carattere da usare per scrivere il testo. La scelta fatta viene ricordata anche per le sessioni future.
- Show ASM window
  Mostra la finestra del codice oggetto (istruzioni base del PLC), che può essere utile per ispezionare meglio il risultato della compilazione di un programma, e può aiutare durante il debug.
  Se necessario, il programma viene compilato prima di mostrare la finestra.
- Show hint popups
  Consente d'inibire la comparsa dei messaggini d'aiuto, comodi solo per le prime sessioni. L'impostazione viene ricordata per le sessioni future.

Compilazione

Quando si apre un file, o si modifica quello corrente, occorre eseguire una compilazione prima di poter eseguire l'invio al modulo. Naturalmente, è possibile compilare al solo scopo di controllare la validità sintattica.

Per compilare il programma, cliccare il bottone apposito o usare il tasto F9. Se il compilatore riscontra un errore, compare un messaggio nello spazio dell'operazione in corso, e l'area di modifica mostra il punto dove è avvenuto l'errore.

Aiuti per la stesura del programma



Le due caselle a scomparsa sotto i bottoni principali possono essere aperte per vedere l'elenco di tutti gli identificatori. La casella di sinistra contiene l'elenco completo (X1, X2, X3...), mentre quella di destra solo il nome di base di ogni risorsa. Selezionando un elemento da una delle due caselle, quella opposta si posiziona in accordo. Il bottone con il simbolo “V” al centro delle due caselle inserisce nel testo del programma l'identificatore attualmente selezionato.

Trasferimento del programma

Dopo la compilazione è possibile trasferire il programma al modulo per eseguirlo o memorizzarlo in modo permanente. Cliccando il pulsante Transfer inizia il trasferimento, che può impiegare alcuni secondi.

Se all'inizio del trasferimento il programma risulta ancora da compilare, viene prima eseguita la compilazione.

La trasmissione del programma comporta l'arresto del ciclo corrente, dato che la memoria del PLC risulta instabile durante l'operazione.

Cancellazione del programma del modulo

Se si desidera cancellare il ciclo attivo, in modo che all'avviamento di EZ-Red nessun ciclo venga eseguito, occorre svuotare l'area di modifica, cioé cancellare tutto il testo all'interno, e poi cliccare con il tasto destro il pulsante Transfer. Questa è un'operazione speciale: l'effetto normale del clic destro sul pulsante è la compilazione e successiva trasmissione del programma; se però il testo del ciclo è manca, il risultato è l'azzeramento del ciclo nel modulo - sia nella memoria di transito (RAM), sia nella memoria non volatile (flash). Prima di eseguire l'operazione viene richiesta una conferma.

Flash-In e Recall

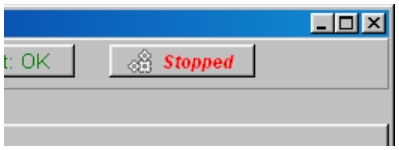
Il bottone Flash-In registra il ciclo PLC sulla memoria interna di EZ-Red, di modo che a ogni avvio (accensione) del modulo il ciclo parta automaticamente.

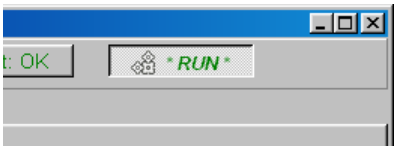
Il pulsante Recall richiama il ciclo dalla memoria non volatile; può servire per ripristinare il programma precedente, dopo averne trasferito un altro da computer. Naturalmente, l'operazione non ha alcun effetto se dopo l'ultima trasmissione il ciclo è stato scritto in modo permanente con Flash-In.

ATTENZIONE: le memorie di tipo flash non possono essere scritte, cancellate e riscritte per un numero infinito di volte; sebbene la quantità di riscritture sia molto alta, è buona norma non abusare di queste operazioni.

Esecuzione e arresto del ciclo

L'esecuzione del ciclo PLC è indipendente dalle operazioni di compilazione e trasferimento, anche se di solito avvio e arresto del programma si usano durante la fase di stesura e controllo del ciclo. Il pulsante in alto a destra della finestra di TSmon, recante la scritta "Stopped" o "* RUN *", segnala lo stato corrente e permette di avviare o arrestare il ciclo:





Quando si comanda l'avvio dell'esecuzione, tutti i parametri interni (flag di opzioni, soglie per gli ingressi, impostazioni del watch-dog) vengono portate al valore di default.

Se, mentre il ciclo è in esecuzione, si vuole apportare modifiche al testo del programma, è opportuno prima arrestare il PLC, perché quando esso è attivo il cursore della finestra di testo si sposta per mostrare le istruzioni in esecuzione.

Debug

Per analizzare il comportamento del ciclo, e scoprire eventuali errori di programmazione, si possono usare diverse tecniche: la più comune è quella di eseguire il programma in modo passo-passo, così da poter vedere chiaramente ogni singola fase. A ogni passo si osservano ingressi e uscite, e l'istruzione successiva (ancora da eseguire), per capire se l'istruzione è scritta in modo corretto oppure no.

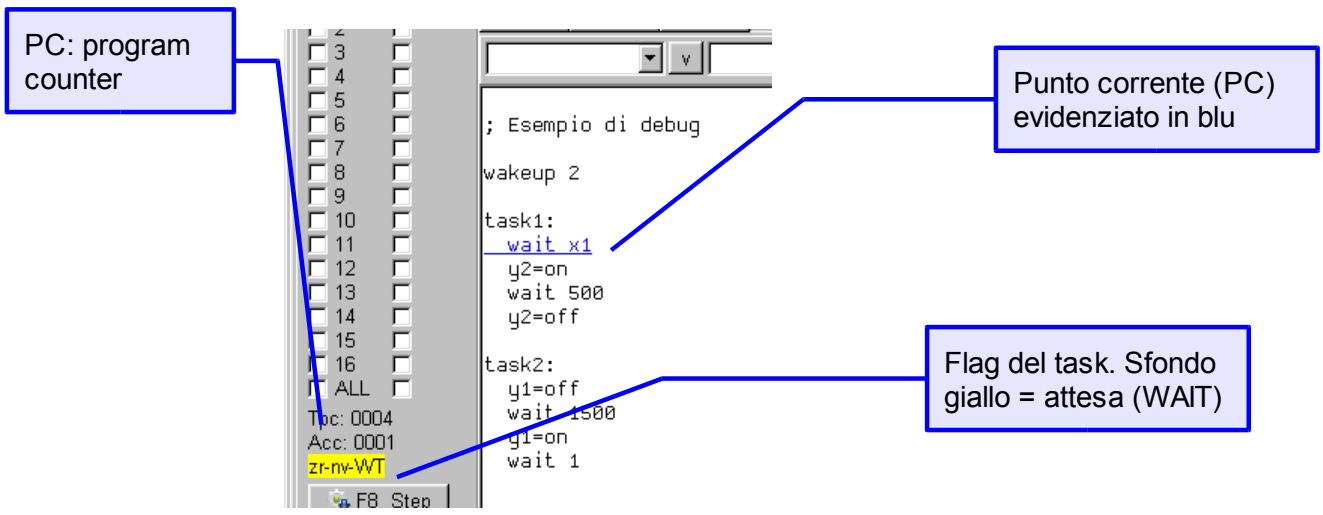
Purtroppo l'ambito applicativo di EZ-Red può rendere difficile il procedimento, sia perché EZ-Red può eseguire diversi Task concorrenti, sia perché un processo di controllo industriale può variare il suo comportamento se non viene eseguito alla velocità normale. Il sistema di debug di EZ-Red permette di sospendere o porre in modo passo-passo uno o più task, in modo da facilitare le operazioni anche nel caso di cicli complessi.

Come esempio pratico, si analizzi il seguente programma:

```
; Esempio di debug

wakeup 2

task1:
  wait x1
  y2=on
  wait 500
  y2=off

task2:
  y1=off
  wait 1500
  y1=on
  wait 1
```

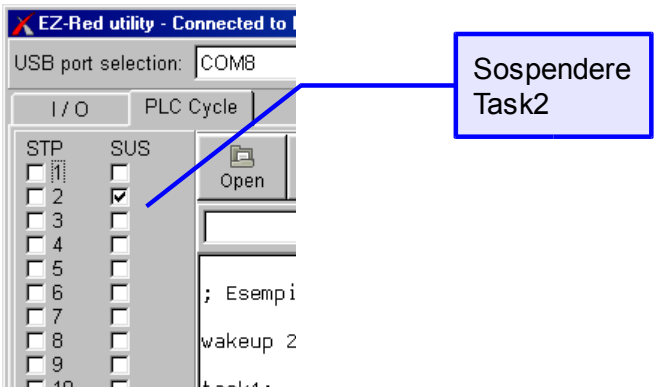
Per fare le prove pratiche si deve collegare l'uscita Y1 con l'ingresso X1.

Il programma contiene due task: il task 1 attende che l'ingresso 1 si alzi, poi genera su Y2 un impulso fi mezzo secondo e ricomincia. Il task 2 genera brevi impulsi su Y1, separati da un secondo e mezzo. Per verificare che sia così, basta scrivere il programma, e cliccare con il destro sul pulsante Transfer, così da compilarlo, trasferirlo a EZ-Red e metterlo in esecuzione. L'area di modifica si anima, mostrando il punto corrente d'esecuzione:



Il punto d'esecuzione dovrebbe trovarsi per la maggior parte del tempo sull'istruzione "wait x1", ma occasionalmente saltare a qualche altro punto.

Visto che il task 1 attende un impulso che viene generato dal task 2, si può provare a sospendere Task2 cliccando nella casella relativa:



Dato che il task 2 non genera più impulsi, il task 1 si ferma. Cliccando nuovamente sulla casella SUS di task 2, Task1 riprende; similmente si può sospendere Task1, e notare che Task2 continua a girare, ma l'uscita Y2 contrallata dal task 1 si ferma.

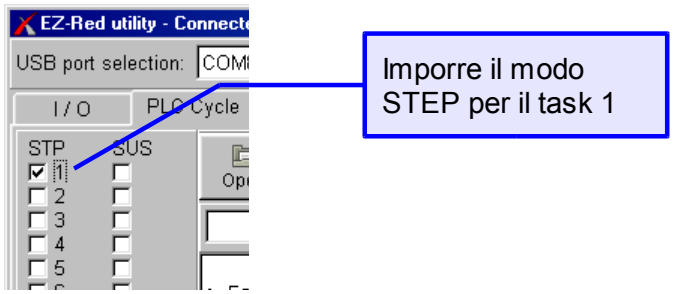
Sospensione di un task

Tramite le caselle della colonna SUS è possibile sospendere uno o più task. La casella più in basso (ALL) agisce su tutti i task insieme. Un task può essere rimesso in esecuzione togliendo la spunta, ma anche il ciclo stesso può farlo, risvegliando programmaticamente un task sospeso con l'istruzione WAKEUP.

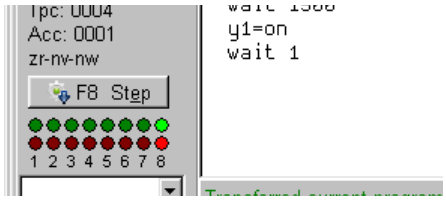
Sospendere un task può servire per non avere interferenze durante l'analisi di altre parti del ciclo.

Modo passo passo

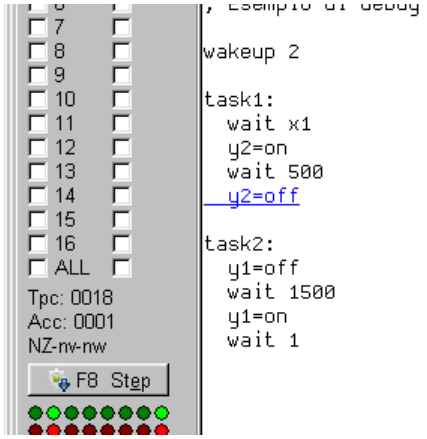
Il modo passo passo è una evoluzione del modo di sospensione. Quando un task è in modo passo-passo (STEP), effettivamente si ferma; è possibile comandare una ripartenza che dura per un'istruzione, e poi il task si ferma di nuovo. Così facendo, è possibile analizzarne il comportamento "al rallentatore". Nel programma di prova, mettere la spunta alla casella STP (step) del primo task:



Dopo aver marcato la casella, il task 1 si ferma. A questo punto premere F8 o cliccare sul pulsante "F8 Step" per eseguire una istruzione per volta:

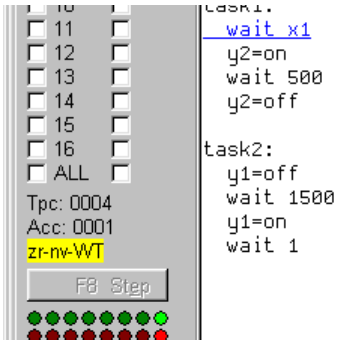


Ogni volta che si esegue step, l'istruzione evidenziata in blu viene eseguita:



Nell'immagine qui sopra, l'istruzione Y2=OFF è in attesa di essere eseguita; premendo F8 si attiverà l'esecuzione. Si noti che, mentre Task1 è fermo in attesa del comando di step, Task2 continua a funzionare normalmente, generando impulsi.

Quando l'istruzione da eseguire è una WAIT, il pulsante Step si disabilita per la durata della wait stessa, e i flag con sfondo giallo confermano che il task è in attesa e non può proseguire:



Al termine della WAIT il punto di esecuzione si sposta automaticamente all'istruzione successiva ("y2=on" in questo caso), e il task si arresta nuovamente. Può succedere che la WAIT duri per sempre, se non si verificano le condizioni per terminare l'attesa: in questo caso occorre agire in modo tale da sbloccare il task, per esempio attivando task sospesi (o ponendoli in modo step), o forzando la circuiteria esterna, oppure modificando le variabili interne di ciclo (risorse "R").

Avvertenze sul modo step

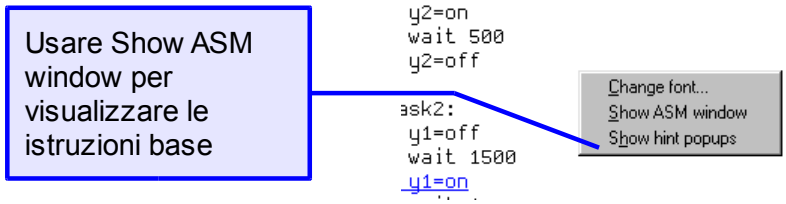
1. Alla fine di ogni task c'è un'istruzione implicita di ripartenza del task stesso. Anche se non visibile, l'istruzione esiste e deve essere eseguita. Quando, durante il modo step, si arriva su questa istruzione, nessuna linea del testo viene evidenziata in blu, però la situazione è normale: occorre comandare un ulteriore step per tornare all'inizio del task.

2. Moltre istruzioni singole, a livello di testo, sono costituite in realtà da una sequenza più lunga di istruzioni base del PLC. Per esempio l'istruzione Y1=ON, che sembra singola, è in realtà costituita da due istruzioni base. Si potrebbe pensare quindi che occorrano due passi (F8 o clic) per eseguirla interamente, e in effetti è così; però il programma TSmon, quando possibile, cerca di facilitare le cose ed esegue automaticamente questi passi aggiuntivi.

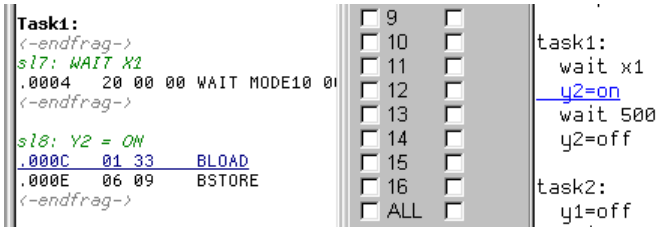
Passo-passo di più task insieme

Per quanto esposto prima, se si mettono in modo step più task insieme, il punto d'esecuzione salta continuamente da un task all'altro. Questo è corretto, perché è esattamente il modo di lavorare del PLC su più task contemporaneamente.

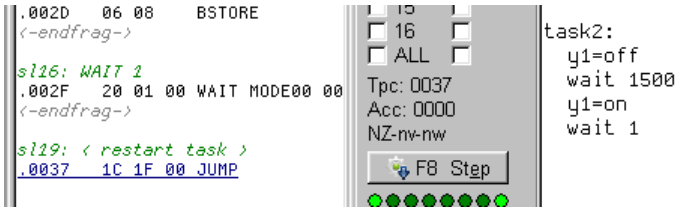
Inoltre, si ha l'impressione che la stessa istruzione di un task venga eseguita più volte. Non è così - succede che c'è un cambio di task continuo, che avviene anche nel mezzo di istruzioni che sembrano semplici ma sono invece costituite da una sequenza. Si può verificare meglio questo comportamento usando la vista di basso livello:



Aprendo la finestra ASM si può verificare esattamente il comportamento del PLC:



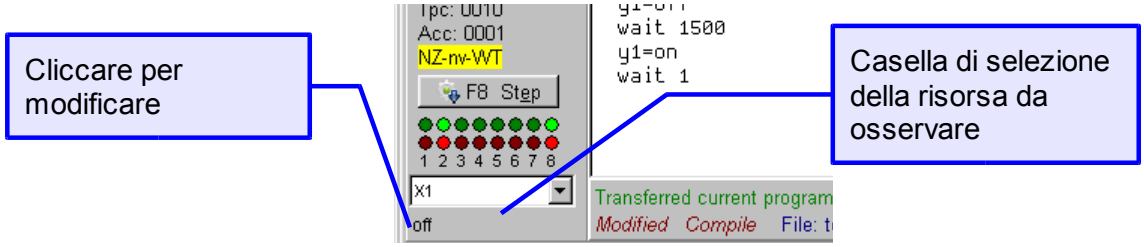
nella parte destra l'istruzione corrente è Y2=ON e, nella parte sinistra, si vede come essa sia formata da due istruzioni base.



Sempre con l'aiuto della vista ASM, si può vedere come dopo il "WAIT 1" del task 2 ci sia un'istruzione invisibile a destra, ma segnata a sinistra con il commento "< restart task >".

Ispezione e modifica delle risorse

Gli elementi sotto il pulsante "F8 Step" permettono di visualizzare e modificare le risorse (ingressi, uscite e variabili interne) del PLC:

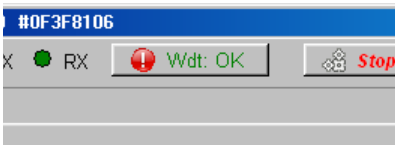


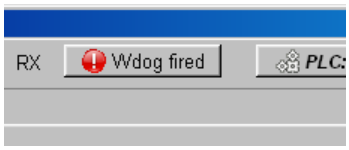
Le sedici spie colorate corrispondono agli ingressi e alle uscite normali; cliccandole si inverte il loro stato. La

casella a scomparsa subito sotto permette di selezionare una risorsa per visualizzarne continuamente il valore corrente. Cliccando sul valore della risorsa selezionata è possibile modificarlo. Un dato di tipo bit viene invertito, mentre gli altri tipi di dato presentano una richiesta interattiva per inserire il valore desiderato.

Occorre notare che è possibile modificare i bit degli ingressi (X1..X8), sia cliccando sulle piccole spie verdi, sia cliccando il valore sotto la casella di selezione per l'osservazione. Dato però che un ingresso, teoricamente, non sarebbe scrivibile, l'effetto che si ottiene è solo temporaneo: la modifica di un ingresso X prende la forma di un breve impulso simulato. La funzionalità è fornita soprattutto per interrompere le attese WAIT in casi particolari. Nel programma d'esempio di queste pagine, si può per esempio sospendere il task 2 e far girare ugualmente il Task1 cliccando l'ingresso 1 quando c'è il WAIT X1.

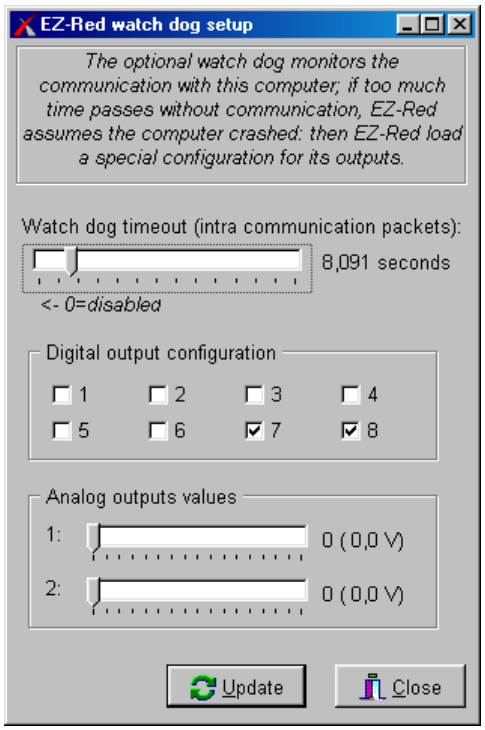
Watch dog





Il pulsante "Wdt: OK" / "Wdog fired" serve per controllare lo stato del watch-dog e impostarne il funzionamento. Esso consente tre operazioni:

1. Quando è rilasciato (Wdt: OK), se viene premuto ferma il ciclo e impone lo stato di allarme (intervento del watch-dog). Si noti che TSmon esegue le due operazioni separatamente, a scopo di "arresto d'emergenza", perché non è sicuro che il watch-dog fermi il ciclo, che anzi può rilevare l'anomalia ed eseguire operazioni aggiuntive.
2. Quando è premuto (Wdog fired), se cliccato cancella il watch-dog, ripristinando così l'operatività normale. Questo non significa che il ciclo, se fermo, riparte; semplicemente, viene tolto l'allarme.
3. Se cliccato con il tasto destro, apre una finestra di dialogo che permette di vedere le impostazioni del watch-dog e modificarle:



Con la finestra visibile, cliccare Update per aggiornare le opzioni nel modulo, oppure Close per non toccare le impostazioni.

Prova del watch-dog

Con il programma d'esempio di queste pagine, porlo in esecuzione normalmente e uscire dal programma TSmon. Il modulo EZ-Red continua a funzionare in modo normale perché il watch-dog è disabilitato.

Poi, riavviare TSmon, collegarsi al modulo tramite la selezione della porta USB corretta, cliccare con il tasto

destro sul pulsante del watch-dog, e impostare il time-out ad alcuni secondi, per esempio 10. Infine, cliccare Update per aggiornare e chiudere. Il ciclo continua a girare normalmente. A questo punto, chiudere TSmon. Dopo 10 secondi il watch-dog interviene, e la spia d'allarme lampeggia: questo accade perché chiudendo TSmon s'interrompe la comunicazione tra PC ed EZ-Red, e correttamente il watch-dog fa il suo lavoro.

E' possibile fare sì che il ciclo rilevi l'intervento del watch-dog per intraprendere azioni ulteriori. Per fare questo, occorre disabilitare il bit WDTSTOPSCYCLE, altrimenti il ciclo si ferma; inoltre occorre inserire alcune istruzioni per gestire l'anomalia. Il modo più semplice è usare un task apposito per rilevare il watch-dog; questo task rimane in attesa (con WAIT) e, terminata la WAIT, intraprende le azioni scelte.

Il listato completo del programma è il seguente:

```
; Esempio di debug

WDTSTOPSCYCLE=false
wakeup 2
wakeup 3

task1:
  wait x1
  y2=on
  wait 500
  y2=off

task2:
  y1=off
  wait 1500
  y1=on
  wait 1

task3:
  ; sorveglianza watch-dog
  wait WDTFIRED
  suspend 1
  suspend 2
  ybyte=0
 cycle:
  ybyte=255
  wait 100
  ybyte=0
  wait 100
  goto cycle
```

In testa al programma ci sono due istruzioni in più: WDTSTOPSCYCLE=false e "wakeup 3". La prima istruzione è indispensabile: se assente, l'intervento del watch-dog ferma anche il PLC, cosicché non sono possibili ulteriori azioni. La seconda, "wakeup 3", fa partire il task il cui unico compito è attendere il watch-dog. Vi possono essere altri modi, comunque: lo stato del watch dog potrebbe essere controllato dal ciclo principale, con un semplice test (IF wdtfired THEN ...). Il task 3, appena parte, si pone in attesa. Quando il watch-dog interviene, task 3 sospende gli altri due task e poi entra in un ciclo di lampeggio di tutte le uscite.