

UNISIM: sviluppo, validazione e prototipizzazione rapida di sistemi di automazione

Gianmaria De Tommasi, Pierangelo Di Sanzo, Alfredo Pironti

UNISIM è un tool grafico che permette di sviluppare progetti d'automazione seguendo le direttive dello standard IEC 61131-3. Gli strumenti messi a disposizione da UNISIM consentono di completare il ciclo di sviluppo del software, effettuando la validazione dell'algoritmo di controllo attraverso un motore di simulazione ed, eventualmente, la sua prototipizzazione utilizzando un PC equipaggiato con schede di ingresso/uscita commerciali. Il tool è stato realizzato seguendo un approccio object-oriented ed il suo modello ad oggetti è caratterizzato da una struttura gerarchica che rispecchia il modello software dello standard IEC 61131-3. UNISIM, inoltre, utilizza il nuovo schema XML Formats for IEC 61131-3 per l'interscambio delle informazioni relative ad un progetto di automazione. Grazie a questa scelta il software sviluppato e testato con UNISIM può essere importato e messo in esecuzione su piattaforme di sviluppo di tipo commerciale. Viceversa, un progetto già sviluppato può essere validato con l'ausilio di UNISIM prima di essere avviato sull'impianto reale.

Keyword

IEC 61131-3, controllori a logica programmabile, prototipizzazione rapida

Nei moderni sistemi di automazione industriale basati su controllori a logica programmabile (PLC,[1]), il compito del progettista non è limitato alla definizione e all'implementazione del codice di controllo, ma comprende anche: la configurazione dell'architettura hardware del sistema di controllo; la definizione delle connessioni di comunicazione tra i vari dispositivi (resources) che compongono il sistema complessivo; l'assegnazione dei compiti (tasks) ad ogni dispositivo del sistema; la definizione delle strutture dati necessarie per svolgere il compito assegnato. È possibile definire un progetto d'automazione come l'insieme delle informazioni di configurazione, dei dati, e del codice di controllo, necessari per realizzare le funzionalità richieste. La norma del Comitato Elettrotecnico Internazionale IEC 61131-3 [2], oltre a definire i linguaggi di programmazione per i controllori a logica programmabile, specifica una struttura standard per i progetti d'automazione.

UNISIM [3] è un tool grafico, distribuito con licenza GPL, che permette di sviluppare software d'automazione secondo le direttive dello standard. Gli strumenti di UNISIM possono essere di ausilio all'intero ciclo di sviluppo del software, mettendo a disposizione dell'utente un simulatore interno per la validazione degli algoritmi, e supportando la prototipizzazione rapida del

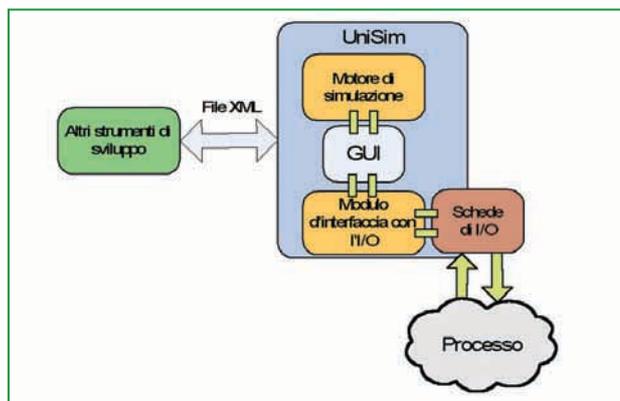


Figura 1 - Schema di principio dell'architettura di UNISIM in cui vengono messe in evidenza le interazioni con i sistemi esterni

sistema attraverso un comune Personal Computer. La versione attuale di UNISIM permette di: gestire la configurazione di una risorsa; sviluppare il codice di controllo utilizzando due dei cinque linguaggi di programmazione previsti dallo standard (il Sequential Functional Chart ed il Ladder Diagram); utilizzare le POU (Program Organization Unit) di tipo program; utilizzare il tipo di dati booleano.

I progetti sviluppati con UNISIM vengono salvati come documenti XML seguendo lo schema XML Formats for IEC 61131-3 [4] specificato dall'organizzazione PLCOpen [5]. Questo standard ha tra i suoi scopi quello di facilitare la portabilità del software tra sistemi di diverse case produttrici. I progetti d'automazione sviluppati con UNISIM, quindi, possono essere importati da tutti gli altri software di sviluppo che supportano

G. De Tommasi, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II; P. Di Sanzo, Università degli Studi di Napoli Federico II; A. Pironti, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II

tale schema XML (cfr figura 1). In ogni caso è possibile esportare il codice sviluppato con UNISIM anche verso altre piattaforme. A questo proposito sono state realizzate delle funzioni di traduzione, capaci di generare, a partire dal file XML prodotto da UNISIM, codice in altri formati. In particolare, nella corrente versione di UNISIM, è possibile esportare codice verso il sistema di sviluppo STEP7 della Siemens [6].

In questa memoria verranno presentati l'architettura software di UNISIM ed una sua applicazione per la validazione hardware-in-the-loop. In particolare verranno descritte la struttura dell'interfaccia grafica, il motore di simulazione, ed il modulo di controllo. Quest'ultimo componente permette di interfacciare il motore di simulazione con delle schede di ingresso/uscita, consentendo la verifica degli algoritmi di controllo direttamente sull'impianto da controllare, oppure utilizzando un simulatore mandato in esecuzione su un secondo calcolatore, come verrà illustrato nel paragrafo Validazione hardware-in-the-loop con UNISIM.

XML formats for IEC 61131-3

In questo paragrafo verrà brevemente introdotto lo schema XML Formats for IEC 61131-3. La gerarchia di oggetti che compongono l'applicazione UNISIM, infatti, è stata progettata in modo da aderire il più possibile a questo schema XML, il quale offre un'interfaccia aperta per l'interscambio di informazioni (progetti, programmi e librerie) tra differenti piattaforme.

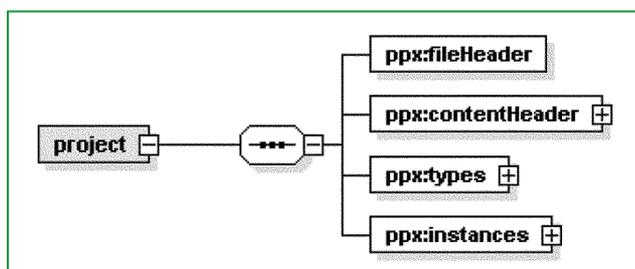


Figura 2 - Vista parziale dell'elemento project dello schema XML format for IEC 61131- 3

L'XML nasce come meta-linguaggio per la definizione dei linguaggi di markup. Esso fornisce un insieme di regole sintattiche per modellare il contenuto dei documenti. Queste regole sono indipendenti dalla piattaforma hardware e software facendo dell'XML uno strumento universale per lo scambio di dati. L'XML Formats for IEC 61131-3 definisce lo schema di un documento [7] contenente tutte le informazioni di un progetto d'automazione. Questo formato ha potenzialmente la capacità di aumentare la portabilità del software d'automazione. In seguito viene presentata la struttura tipica di un documento XML. Per maggiori informazioni si faccia riferimento a [4]. L'elemento project rappresenta la radice del documento e contiene quattro

¹Il linguaggio previsto dallo standard IEC sono Instruction List (IL), Function Block Diagram (FBD), Ladder Diagram (LD) e Sequential Functional Diagram (SFC).

nodi (cfr figura 2): fileHeader; contentHeader; types; instances. I primi due nodi contengono informazioni generali sul progetto (nome, versione, data dell'ultima modifica, nome del tool con il quale è stato generato, ecc.).

Il nodo types contiene (cfr figura 3): le definizioni dei tipi di dati utilizzati nel progetto (nodo dataTypes); le definizioni delle varie unità di programma (nodo pous), vale a dire dei blocchi Program,

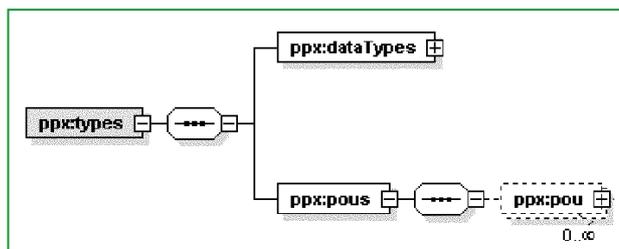


Figura 3 - Vista parziale dell'elemento types dello schema XML for IEC 61131-3

Functional Block e Function previsti dalla norma [2].

Ogni elemento pous nella figura 3 è composto da un corpo, che può essere specificato in uno dei cinque linguaggi previsti dallo standard¹. Il nodo instances contiene una lista delle configurazioni, nelle quali sono specificate le risorse. Ogni risorsa può contenere a sua volta più liste di variabili globali (visibili da tutti i POU della risorsa), istanze di POU (pouInstance) e task (cfr figura 4).

Architettura software

UNISIM è stato progettato sulla base di un metodo orientato agli oggetti, utilizzando la piattaforma .NET di Microsoft ed il linguaggio VB.NET [8].

Dal diagramma delle classi riportato nella figura 5 si può notare come l'architettura software di UNISIM sia strettamente legata al modello di progetto d'automazione previsto dallo standard IEC, ed allo schema XML presentato nel paragrafo precedente.

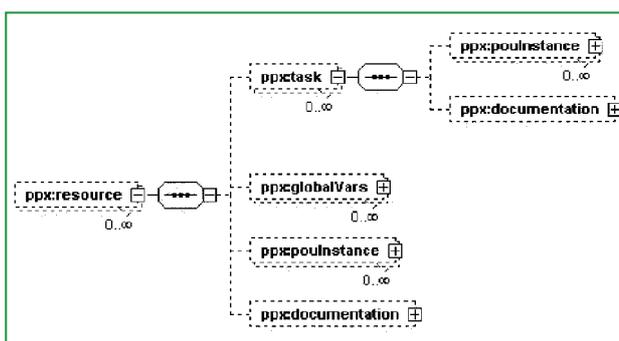


Figura 4 - Vista parziale dell'elemento resource dello schema XML for IEC 61131-3

In particolare l'oggetto SimulationEngine implementa il motore di simulazione utilizzato per la validazione offline del software di controllo. Il componente ControlEngine, invece, realizza l'interfaccia con le schede di I/O, permettendo di collegare UNISIM

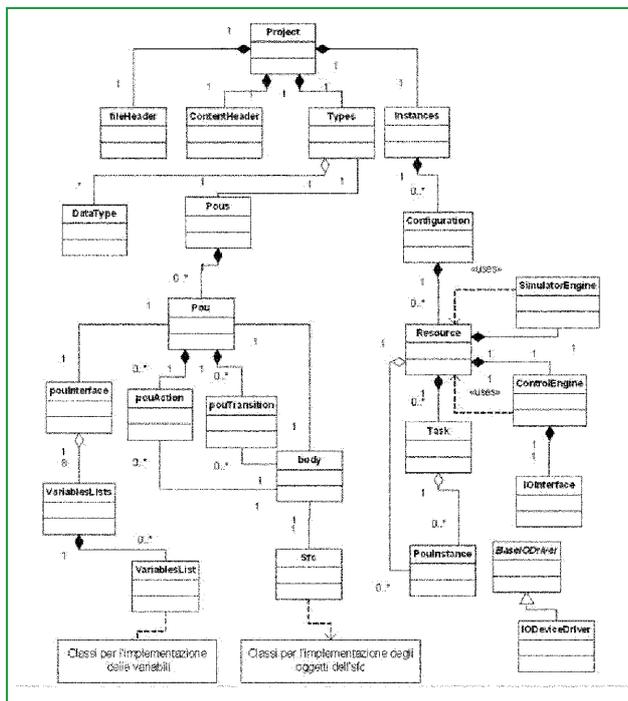


Figura 5 - Estratto del diagramma delle classi di UNISIM

ad un simulatore di processo per la validazione hardware-in-the-loop, oppure direttamente al processo da controllare, realizzando le funzionalità di un softPLC. L'implementazione di questi due oggetti è indipendente dal linguaggio utilizzato per specificare le POU, quindi lo sforzo implementativo richiesto per aggiungere nuovi linguaggi è minimo. Questo paragrafo descrive i componenti principali dell'architettura software di UNISIM: l'editor grafico per lo sviluppo dei progetti; il motore di simulazione; il modulo d'interfaccia con dispositivi di I/O.

L'interfaccia utente

L'interfaccia utente consente di: creare e modificare i progetti; effettuare la simulazione e monitorare a run-time l'esecuzione delle POU; importate ed esportare i progetti utilizzando documenti XML. Nella figura 6 è riportata una schermata dell'interfaccia utente di UNISIM. Durante la fase di editing, l'utente può creare i task e le POU, scegliendo il linguaggio desiderato per ognuna di esse. In particolare, ad ogni task è associata la priorità, la modalità di esecuzione (ciclica oppure periodica) e la lista di istanze di POU da eseguire. Oltre alle variabili globali, che possono essere organizzate in liste diverse, ad ogni POU è possibile associare delle variabili locali. Durante l'esecuzione del codice è possibile monitorare i valori di tutte le

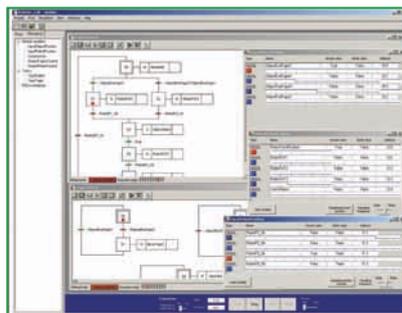


Figura 6 - Interfaccia utente di UNISIM

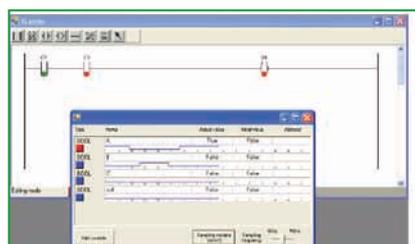


Figura 7 - Esempio di programma ladder con UNISIM

variabili, mentre la loro modifica è consentita solo durante la validazione offline con il motore di simulazione. È possibile, infine, monitorare sia l'evoluzione dei grafici SFC (stato delle fasi e delle transizioni) come riportato nella figura 6, che il valore dei singoli contatti di un programma ladder (cfr figura 7).

Il motore di simulazione

La classe SimulatorEngine implementa un processore virtuale per ognuna delle risorse presenti nel progetto, e fornisce i metodi per l'avvio, l'arresto, il reset, e l'esecuzione a passi delle POU.

Il motore di simulazione consente di effettuare modifiche "a caldo" sul codice di controllo e sulla configurazione delle risorse, senza dover interrompere l'esecuzione. Questa scelta progettuale ha reso necessaria l'eliminazione della fase di compilazione del progetto. Le operazioni proprie di questa fase, come la verifica della correttezza sintattica e semantica o la traduzione dei nomi simbolici in indirizzi fisici di memoria, vengono eseguite contestualmente alle operazioni di editing.

Il modulo di interfaccia con l'I/O

Il modulo d'interfaccia realizza la comunicazione con i dispositivi di I/O, consentendo di utilizzare UNISIM per la prototipizzazione rapida e la validazione hardware-in-the-loop del software di controllo. L'evoluzione dello stato dei programmi, dei valori delle variabili, dell'esecuzione dei task, ecc. può essere monitorato con gli stessi strumenti grafici utilizzati durante la simulazione. Ogni risorsa di un progetto contiene un'istanza della classe ControlEngine, la quale implementa il processore virtuale che esegue le funzioni di controllo. La classe ControlEngine, inoltre, contiene un'istanza della classe IOInterface, che implementa l'interfaccia di comunicazione verso i dispositivi di I/O. Questa classe rappresenta l'interfaccia astratta da implementare per collegare una qualsiasi scheda di I/O ad UNISIM. L'algoritmo di esecuzione è simile a quello utilizzato dal motore di simulazione. A differenza di quest'ultimo vengono eseguite le istruzioni per la lettura e la scrittura dei dati sui dispositivi fisici. In particolare per ogni ciclo vengono effettuate le seguenti operazioni: lettura degli ingressi dal dispositivo fisico; schedulazione dei task ed esecuzione di un singolo ciclo di controllo; scrittura delle uscite sul dispositivo fisico. Se una delle operazioni di lettura o di scrittura fallisce, l'evento viene registrato con la relativa descrizione in una lista degli errori. È importante tenere conto delle limitazioni di UNISIM quando utilizzato per controllare un processo fisico. Il modulo di controllo, infatti, viene eseguito da un sistema operativo non real-time, i cui tempi di risposta sono influenzati dalle caratteristiche del sistema e dal carico di processi da eseguire; inoltre la precisione dei timer è influenzata da quella dell'orologio di sistema.

Nel caso di controllo di processi semplici, che non presentino criticità dal punto di vista delle specifiche temporali (sistemi soft real-time), il modulo di controllo di UNISIM può essere utilizzato per la realizzazione di sistemi d'automazione basati su PC equipaggiati con schede di acquisizione a basso costo (softPLC).

Validazione hardware-in-the-loop con UNISIM

Nel paragrafo precedente è stato presentato il modulo di interfaccia con l'I/O, che consente ad UNISIM di comunicare con le schede di acquisizione. In questo modo, quindi, UNISIM può

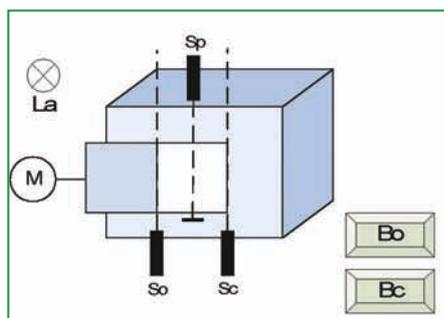


Figura 8 - Impianto da controllare

essere collegato ad un sistema esterno, il quale può essere sia il processo da controllare, che un simulatore da utilizzare per la validazione del software.

In questa sezione, con l'ausilio di un esempio

didattico, verrà presentata un'architettura di test per la validazione hardware-in-the-loop del software d'automazione. In particolare la soluzione presentata prevede l'utilizzo di un simulatore di impianto collegato direttamente al PC sul quale viene eseguito il codice sviluppato con UNISIM.

Si supponga di voler controllare l'apertura e la chiusura della porta del forno riportato nella figura 8. I segnali di ingresso e uscita del processo sono:

- i due ingressi Mo ed Mc del motore M, per effettuare rispettivamente l'apertura e la chiusura della porta;
- i due pulsanti Bo e Bc per comandare rispettivamente l'apertura e la chiusura della porta;
- So ed Sc sono i segnali provenienti rispettivamente dai finecorsa di apertura e chiusura;
- Sp è il segnale proveniente da un sensore che segnala la presenza di oggetti che intralciano il movimento della porta;
- La è una lampada di segnalazione da utilizzare in caso di emergenze o anomalie.

La porta del forno è supposta inizialmente chiusa e le specifiche di funzionamento per il sistema di controllo sono le seguenti:

- la porta si deve aprire in seguito alla pressione del pulsante Bo;
- la porta si deve chiudere in seguito alla pressione del pulsante Bc, oppure se è rimasta aperta per un tempo maggiore di 10 secondi;
- se il sensore Sp rileva un oggetto mentre la porta si sta muovendo, bisogna accendere la lampada di segnalazione La.

Nella figura 9 è riportata l'architettura di test per il software di controllo (mostrato sotto forma di un diagramma SFC nella

figura 10). I due PC utilizzati sono collegati tra loro attraverso due schede di I/O National Instruments PCI-6527. In particolare: l'HOST esegue sia l'algoritmo di controllo, utilizzando UNISIM, che l'interfaccia utente del simulatore, sviluppata in Matlab; il TARGET, esegue il simulatore di processo attraverso il sistema operativo real-time xPC Target della Mathworks [9]. I sensori e gli attuatori del processo nella figura 8 sono mappati sulla scheda di I/O del TARGET, direttamente collegata alla scheda di acquisizione gestita da UNISIM su l'HOST.

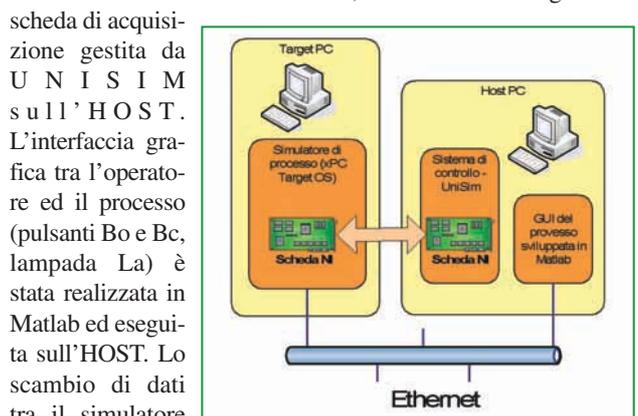


Figura 9 - Architettura di test per la validazione hardware-in-the-loop

Lo scambio di dati tra il simulatore di processo e questa interfaccia avviene utilizzando il protocollo TCP/IP sulla connessione Ethernet. L'architettura presentata in questo paragrafo prevede la disponibilità del simulatore di processo e dell'interfaccia operatore. La loro implementazione non è vincolata all'utilizzo di particolari tecnologie, dato che il collegamento tra UNISIM e questi componenti avviene attraverso i segnali elettrici delle schede di I/O. Si noti che, qualora l'impianto non presenti criticità dal punto di vista dei requisiti temporali (cfr paragrafo *Il modulo di interfaccia con l'I/O*), è sufficiente sostituire i collegamenti tra l'HOST ed il TARGET, con quelli tra l'HOST ed il processo reale per poter mettere in opera il sistema di controllo.

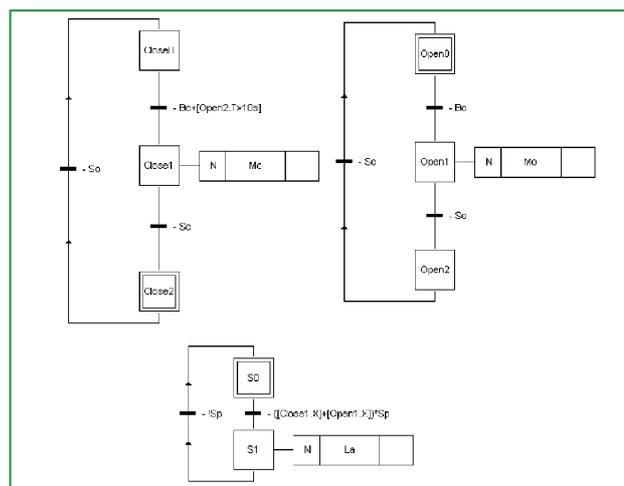


Figura 10 - SFC di controllo per l'esempio del paragrafo 'Validazione hardware-in-the-loop con UNISIM'

cia con l'I/O), è sufficiente sostituire i collegamenti tra l'HOST ed il TARGET, con quelli tra l'HOST ed il processo reale per poter mettere in opera il sistema di controllo.

Conclusioni

UNISIM è un tool di sviluppo per la creazione di progetti d'automazione secondo le direttive dettate dalla norma IEC 61131-3. Le sue caratteristiche principali sono: la presenza di un modulo per la prototipizzazione rapida del sistema di controllo; l'adozione dello standard XML Formats for IEC 61131-3 per l'interscambio dei dati relativi al progetto d'automazione. Il modulo di prototipizzazione permette di effettuare la validazione hardware-in-the-loop dell'algoritmo di controllo, mentre l'adozione del formato XML consente di riutilizzare i progetti sviluppati e validati con UNISIM su altre piattaforme commerciali. La versione attuale di UNISIM presenta ancora alcune limitazioni, tra le quali, le più importanti sono: l'implementazione solo parziale dei linguaggi e dei tipi di dato previsti dalla norma IEC 61131-3; la possibilità di gestire un solo dispositivo (risorsa) all'interno del progetto d'automazione. Queste limitazioni saranno superate nelle successive versioni. Al fine di aumentarne la diffusione e per poter approfittare di eventuali contributi da parte di una più vasta comunità di sviluppatori, UNISIM è stato rilasciato con licenza GPL.

Infine si vuole fare una considerazione circa l'utilizzo di UNISIM come supporto all'insegnamento dello standard IEC-61131-3 nei corsi di automazione industriale. Sebbene il suo impiego non possa sostituire quello delle piattaforme commerciali in sistemi di controllo complessi, UNISIM rappresenta una valida alternativa dal punto di vista didattico. Grazie al suo simulatore interno, infatti, UNISIM può essere utilizzato sia dall'insegnante in classe per illustrare gli esempi di programmazione, che dagli studenti per risolvere e verificare i propri esercizi, senza la necessità di avere a disposizione un PLC con i relativi tool di sviluppo. La capacità di interfacciamento di UNISIM con le schede di I/O, inoltre, consente, di allestire un laboratorio di automazione utilizzando semplici PC.

Questo comporta una notevole riduzione dei costi necessari per l'allestimento di una singola postazione di laboratorio, soprattutto se comparati a quelli necessari per l'acquisto di un PLC commerciale con relativo il software proprietario.

Bibliografia

- [1] P. Chiacchio, F. Basile, *Tecnologie informatiche per l'automazione*, 2a edizione, McGraw-Hill, 2004.
- [2] IEC 61131-3 *International Standard 2nd Edition*, IEC publishing, 2003.
- [3] Sito UNISIM, <http://wpage.unina.it/detommas/unisim/>
- [4] PLCopen Technical Committee 6, *XML Formats for IEC 61131-3*, Technical report, PLCopen, 2005.
- [5] Sito PLCOpen, http://plcopen.org/pages/fr_tc6.htm.
- [6] *Programmazione con Step 7*, Siemens AG, 1998.
- [7] E. van der Vlist, "XML Schema", O'Reilly, 2002.
- [8] Sito Microsoft .NET, <http://www.microsoft.com/net/>.
- [9] *xPC Target User's Guide*, Mathworks, 2005. ■

Tamagawa

SmartSYN[®] FA-SOLVER[®]

resolver

Resolver Brushless - Synchro

- versioni pancake per impiego su motori brushless
- versioni con albero per impiego su macchine automatiche e applicazioni militari
- versioni a riluttanza variabile per alte velocità
- diametri esterni da 20 a 290 mm
- possibilità di versioni custom per l'automazione e la robotica

garnet

Via Magellano 14
20049 Concorezzo (MI)
Tel. +39 039 6666158
Fax +39 039 6906081
garnet@garnetitalia.com
www.garnetitalia.com

resolverservices.it n. 04323