

IEC 61131-3: una risorsa di programmazione standard

IEC 61131-3 rappresenta il primo vero tentativo di standardizzare i linguaggi di programmazione nell'automazione industriale. Grazie alla sua diffusione mondiale, questo supporto è indipendente dalle singole aziende.

IEC 61131-3 è la terza parte della famiglia IEC 61131, e consiste di:

- Parte 1: Panoramica generale
- Parte 2: Hardware
- Parte 3: Linguaggi di programmazione
- Parte 4: Linee guida per l'utente
- Parte 5: Comunicazione
- Parte 7: Logica fuzzy
- Parte 8: Linee guida dell'applicazione

Diversi sono i modi con cui guardare la parte 3 di questo standard. Per nominarne solo alcuni:

- il risultato della Task Force 3, Linguaggi di programmazione, contenuto in IEC TC65 SC65B
- il risultato di un duro lavoro svolto da sette società internazionali che apportano dieci anni di esperienza nel settore dell'automazione industriale
- circa 200 pagine di testo, con oltre 60 tabelle, comprese le tabelle delle funzioni
- la specifica della sintassi e della semantica di una suite unificata di linguaggi di programmazione, che comprende il modello software generale e un linguaggio di strutturazione.

Un altro modo elegante di vedere la questione consiste nel dividere lo standard in due parti:

1. Elementi comuni
2. Linguaggi di programmazione



Guardiamo queste parti nel dettaglio:

Elementi comuni

Dati, tipizzazione

I tipi di dati sono definiti all'interno degli elementi comuni. La tipizzazione dei dati impedisce che vengano commessi errori in una prima fase. Serve per definire il tipo dei parametri utilizzati. In questo modo, ad esempio, si evita la divisione di una data per un intero.

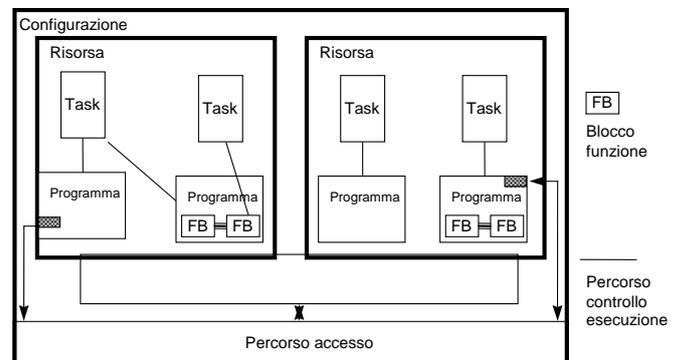
I tipi di dati comuni sono Booleano, Intero, Reale e Byte e Parola, ma anche Data, Ora_del_giorno e Stringa. Sulla base di questi tipi si possono definire tipi di dati personalizzati, chiamati tipi di dati derivati. Si può così definire come tipo di dati un canale di ingresso analogico da riutilizzare più e più volte.

Variabili

Le variabili sono assegnate soltanto a indirizzi hardware espliciti (ad es. ingressi e uscite) di configurazioni, risorse o programmi. In questo modo si crea un'indipendenza hardware di alto livello, in grado di supportare la riutilizzabilità del software. Il valore delle variabili normalmente è limitato all'unità organizzativa nella quale è dichiarato, ad esempio l'unità logica. Ciò significa che i loro nomi possono essere riutilizzati in altre parti senza creare conflitto, eliminando un'altra fonte di errori, ad esempio la memoria parziale ausiliaria. Se le variabili devono assumere valore globale, devono essere dichiarate come tali (VAR_GLOBAL). Per garantire la correttezza dell'impostazione, ai parametri si può assegnare un valore iniziale di avviamento e riavviamento a freddo.

Configurazione, Risorse e Task

Per comprendere meglio questi concetti, possiamo osservare il modello software definito nello standard (vedi sotto).



Al livello più alto, l'intero software che deve risolvere un particolare problema di controllo può essere formulato come una *Configurazione*. Una configurazione è specifica per un tipo particolare di sistema di controllo, che comprende la configurazione dell'hardware, ovvero le risorse di elaborazione, gli indirizzi di memoria per i canali I/O e le capacità di sistema. All'interno di una configurazione si possono definire una o più *Risorse*. Una risorsa può essere considerata come un elemento di elaborazione in grado di eseguire programmi IEC. All'interno di una risorsa si possono definire uno o più *Task*. I task controllano l'esecuzione di un gruppo di programmi e/o blocchi funzione. Questi possono essere eseguiti in base ad un'attivazione periodica oppure in presenza di un trigger specifico, ad esempio la modifica di una variabile.

I *Programmi* sono formati da un certo numero di diversi elementi software scritti in uno dei linguaggi IEC definiti. Un programma è generalmente formato da una rete di *Funzioni* e *Blocchi funzione*, che sono in grado di scambiare dati. Funzioni e blocchi funzione rappresentano i blocchi costruttivi fondamentali che contengono una struttura dati e un algoritmo.

Confrontiamolo con un PLC tradizionale: questo contiene una risorsa che esegue un task, che controlla un programma, che gira in un circuito chiuso. IEC 61131-3 aggiunge molto di più a tutto ciò, aprendo le porte al futuro. Ed è un futuro che comprende programmi multi-processing e programmi eseguiti al verificarsi di un determinato evento. Questo futuro non è poi così lontano: basta guardare ai sistemi distribuiti o ai sistemi di controllo in tempo reale. IEC 61131-3 è utilizzabile in un'ampia gamma di applicazioni, senza dover apprendere ulteriori linguaggi di programmazione.

Program Organization Units (POU)

Nello standard IEC 61131-3, i Programmi, Blocchi funzione e le Funzioni sono chiamati Program Organization Units (POU, unità di organizzazione del programma).

Funzioni

IEC ha definito funzioni standard e funzioni definite dall'utente. Le funzioni standard sono, ad esempio, ADD (addizione), ABS (assoluto), SQRT, SINus e COSinus. Le funzioni definite dall'utente, una volta definite, possono essere utilizzate più e più volte.

Blocchi funzione, FB

I blocchi funzione sono l'equivalente degli IC (circuiti integrati) che rappresentano una funzione di controllo specializzata. Questi contengono sia i dati sia l'algoritmo, quindi possono "ricordarsi" del passato, differenziandosi anche in questo dalle funzioni. Hanno un'interfaccia ben definita e nascondono elementi interni, come un IC o una black box. In questo modo permettono di differenziare nettamente i livelli di programmatori o di addetti alla manutenzione.

Un esempio eccellente di blocco funzione è il loop di controllo della temperatura, o PID. Una volta definito, può essere riutilizzato più volte all'interno dello stesso programma, di programmi diversi o anche di progetti diversi. Questo lo rende estremamente riutilizzabile.

I blocchi funzione possono essere scritti in tutti i linguaggi IEC e nella maggior parte dei casi anche in "C". Possono quindi essere definiti dall'utente. I blocchi funzione derivati si basano sui FB definiti dallo standard, ma nello standard possono rientrare anche FB personalizzati, completamente nuovi: lo standard ne rappresenta soltanto la struttura.

Analogamente si descrivono le interfacce di funzioni e blocchi funzione:

```
FUNCTION_BLOCK Esempio
VAR_INPUT:
    X : BOOL;
    Y : BOOL;
END_VAR

VAR_OUTPUT
    Z : BOOL;
END_VAR

(* dichiarazioni del corpo blocco funzione *)

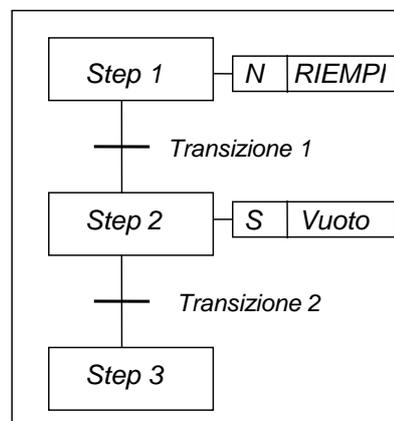
END_FUNCTION_BLOCK
```

Le dichiarazioni sopra indicate descrivono l'interfaccia ad un blocco funzione con due parametri di ingresso booleani ed un parametro di uscita booleano.

Programmi

Considerando i suddetti blocchi costruttivi base, si può affermare che un programma altro non è che una rete di funzioni e blocchi funzione. Un programma può essere scritto in qualunque linguaggio di programmazione definito.

Il linguaggio SFC, Sequential Function Chart



Il linguaggio SFC descrive graficamente le sequenze di un programma di controllo. È derivato dalle reti di Petri e dal linguaggio Grafcet IEC 848, dopo aver apportato le modifiche necessarie per convertire la rappresentazione da uno standard di documentazione ad un gruppo di elementi di controllo dell'esecuzione.

SFC struttura l'organizzazione interna di un programma e aiuta a decomporre un problema di controllo in parti gestibili, mantenendo intatto il suo prospetto.

SFC è formato da step collegati a blocchi di azione e transizioni. Ogni step rappresenta uno stato particolare dei sistemi attualmente sotto controllo. Una transizione è associata a una condizione che, se vera, produce la disattivazione dello step prima della transizione e l'attivazione dello step successivo. Gli step sono collegati a blocchi di azione, che eseguono una determinata azione di controllo. Ciascun elemento può essere programmato in qualunque linguaggio IEC, compreso il linguaggio SFC stesso.

Si possono utilizzare sequenze alternative e anche sequenze parallele, come è richiesto in genere nelle applicazioni batch. Ciò significa ad esempio, che si può utilizzare una sequenza per il processo primario e la seconda sequenza per monitorare i vincoli di funzionamento globale.

Grazie alla sua struttura generale, SFC dispone anche di uno strumento di comunicazione che permette di mettere in contatto persone di ambienti, reparti o paesi diversi.

Linguaggi di programmazione

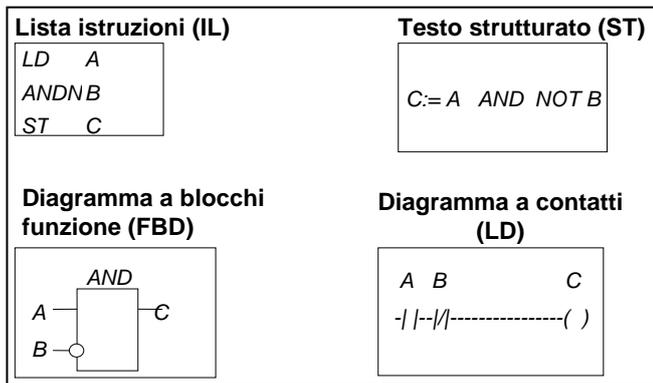
Nello standard sono definiti quattro linguaggi di programmazione. Ciò significa che ne sono state definite sintassi e semantica, rendendo così impossibile l'uso di dialetti. Una volta appresi, sulla base di questo standard si possono utilizzare svariati sistemi. I linguaggi sono formati da due versioni testuali e due versioni grafiche:

Testuale:

- Lista istruzioni, IL
- Testo strutturato, ST

Grafico:

- Diagramma a contatti, LD
- Diagramma a blocchi funzione, FBD



Tutti i quattro linguaggi della figura qui sopra descrivono la stessa parte di programma semplice.

Quale sia il linguaggio di programmazione da scegliere dipende da:

- le conoscenze di base dei programmatori
- il problema da risolvere
- il livello di descrizione del problema
- la struttura del sistema di controllo
- l'interfaccia con altre persone/altri reparti

Tutti i quattro linguaggi sono interconnessi: forniscono una suite comune con un collegamento all'esperienza a disposizione. Offrono così anche uno strumento di comunicazione che permette di mettere in contatto persone di ambienti diversi.

Il *Diagramma a contatti* ha origine negli USA. Si basa sulla presentazione grafica della Relay Ladder Logic.

La *Lista istruzioni* è il suo corrispondente europeo. Come linguaggio testuale, ricorda assembler.

Il *Diagramma a blocchi funzione* è molto diffuso nell'industria dei processi. Esprime il comportamento di funzioni, blocchi di funzione e programmi come un gruppo di blocchi grafici interconnessi, come nei diagrammi dei circuiti elettronici. Considera un sistema dal punto di vista del flusso di segnali che agisce tra elementi di elaborazione.

Il *Testo strutturato* è un linguaggio di alto livello molto potente che ha origine in Ada, Pascal e "C". Contiene tutti gli elementi fondamentali di un linguaggio di programmazione moderno, comprese le istruzioni selettive (IF-THEN-ELSE e CASE OF) e i loop iterativi (FOR, WHILE e REPEAT). Questi elementi possono essere anche nidificati.

Un utilizzo eccellente del testo strutturato consiste nella definizione di blocchi funzione complessi che possono essere a loro volta utilizzati all'interno di tutti gli altri linguaggi.

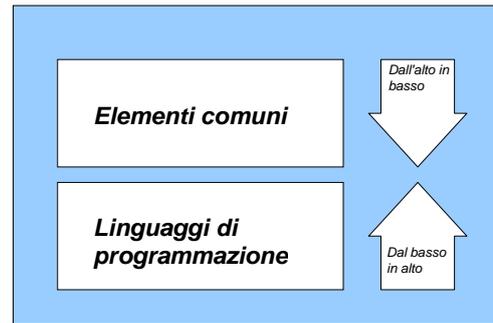
Esempio di programma in ST:

```
I:=25;
WHILE J<5 DO
    Z:= F(I+J);
END_WHILE

IF B_1 THEN
    %QW100:= INT_TO_BCD(Display)
ENDIF

CASE TW OF
    1,5: TEMP := TEMP_1;
    2:   TEMP := 40;
    4:   TEMP := FTMP(TEMP_2);
ELSE
    TEMP := 0;
    B_ERROR :=1;
END_CASE
```

Dall'alto in basso o dal basso in alto



Lo standard permette dunque di sviluppare il programma in due modi: dall'alto verso il basso e dal basso verso l'alto. Si specifica l'intera applicazione e la si divide in sotto-parti, si dichiarano le variabili, e così via; oppure si inizia a programmare l'applicazione dal basso, ad esempio tramite funzioni derivate e blocchi di funzione. A prescindere da quale delle due soluzioni si scelga, l'ambiente di sviluppo assiste il programmatore durante tutto lo svolgimento del processo.

Implementazioni

Non è semplice soddisfare i requisiti generali dello standard IEC 61131-3. Per questo motivo, lo standard consente implementazioni parziali in diversi aspetti. Tra questi il numero di linguaggi, funzioni e blocchi funzione supportati. Il fornitore ha dunque libertà di manovra, ma un utente dovrà tenerne conto durante il proprio processo di selezione. Una versione nuova, inoltre, può presentare un livello di implementazione decisamente superiore.

Molti ambienti di programmazione IEC attuali offrono tutto quello che ci si può attendere da ambienti moderni: uso del mouse, menù a comparsa, schermate di programmazione grafica, supporto multi-finestra, funzioni di ipertesto incorporate e verifica durante la progettazione. Si ricorda che ciò non è specificato nello standard stesso, poiché è proprio una di quelle parti in cui i fornitori hanno libertà di azione.

Conclusioni

Le implicazioni tecniche dello standard IEC 61131-3 sono consistenti e lasciano spazio sufficiente per la crescita e la differenziazione. Ed è proprio questo che fa prevedere una buona evoluzione nel prossimo secolo.

IEC 61131-3 avrà un forte impatto sull'intera industria del controllo industriale. Sicuramente non si limiterà al mercato tradizionale dei PLC. Oggi lo possiamo vedere in uso nel mercato del controllo movimentazione, nei sistemi distribuiti e nei sistemi di controllo basati su PC/softlogic, compresi i pacchetti SCADA. Ma i settori d'impiego stanno aumentando ancora.

La possibilità di poter disporre di un'area applicativa così vasta, non può che portare numerosi vantaggi agli utenti/programmatori. Diversi sono i vantaggi che derivano dall'adozione di questo standard e dipendono dalle aree in cui si applica lo standard. Per nominarne solo alcuni:

- riduzione di spreco di risorse umane in formazione, debugging, manutenzione e consulenza
- creazione di un fulcro per la soluzione dei problemi tramite un elevato livello di riutilizzabilità del software
- riduzione delle incomprensioni e degli errori
- tecniche di programmazione utilizzabili in un ambiente ampio: il controllo industriale generale
- combinazione di componenti diversi rilevati da programmi, progetti, ubicazioni, società e/o paesi diversi

Perfezionamenti PLCopen dello standard

PLCopen è un'associazione mondiale indipendente di venditori e produttori, con sede nei Paesi Bassi. Come tale offriamo efficienza nei processi di automazione grazie all'armonizzazione durante le attività di sviluppo, installazione e manutenzione.

PLCopen è un'associazione che dipende dal proprio introito rappresentato dalle rette dei propri membri. Per questo motivo PLCopen supporta membri multi-livello che vanno da fornitori a istituti educativi. PLCopen supporta fortemente la comunità di utenti. Per questo motivo ha creato ulteriori categorie di appartenenza.

Per raggiungere i propri obiettivi, l'organizzazione PLCopen si concentra sul perfezionamento della programmazione del controllo industriale, per riuscire a creare una maggiore efficienza. Qui di seguito si descrivono in breve i comitati che lavorano all'interno di PLCopen e i risultati che hanno ottenuto.

Risultati tecnici

I Comitati tecnici, TC, con rappresentanti di membri PLCopen, lavorano su argomenti specifici.

In **TC1 - Standard** PLCopen raccoglie le proposte dei propri membri per il gruppo di lavoro IEC 65B WG7, sviluppa una posizione congiunta e distribuisce le informazioni di interesse. Questa attività si è concentrata in modo specifico sulla seconda edizione dello standard che è uscita all'inizio del 2003 ed oggi è alla terza edizione.

Il **TC2 - Funzioni** definisce le librerie comuni di funzioni (blocchi funzione) per aree di applicazione specifiche. Ne sono esempio le definizioni di libreria di Function Blocks for Motion Control. Questa standardizzazione integra il controllo di movimentazione nel controllo industriale. Come tale fornisce agli utenti un supporto interattivo: programmatori e addetti all'installazione e alla manutenzione. Le implementazioni multiple di questa libreria permettono di semplificare la riutilizzabilità del software e la scalatura del sistema di controllo, anche in architetture e/o marche di controller diverse.

Test di certificazione e conformità

Il **TC3 - Certificazione** definisce un sistema di certificazione per PSE (Program Support Environments) IEC 61131-3 (ambienti di sviluppo). Ogni PSE può essere testato per verificarne la conformità con un sotto gruppo specifico PLCopen dello standard IEC 61131-3. Per ciò, per parecchi livelli la definizione **Conformity Level, CL** (livello di conformità) sta ad indicare il livello massimo.

Il **Reusability Level, RL**, inoltre, serve per rendere riutilizzabili in un PSE diverso le unità di programmazione funzioni e blocchi funzione utilizzando lo stesso linguaggio di programmazione. Storicamente esiste un livello di entrata, chiamato **Base Level**, che indica un impegno rispetto allo standard.

Sono già terminate specifiche dettagliate per la maggior parte dei linguaggi IEC 61131-3. Sono state definite la procedura di verifica della conformità e dell'accreditamento dei laboratori di prova. Sono stati accreditati laboratori di prova indipendenti ed è stato certificato un numero crescente di prodotti. Per una lista completa, fare riferimento al sito web www.PLCopen.org.

Il **TC4 - Comunicazioni** si occupa della interazione tra comunicazione e linguaggi di programmazione. Lo scopo è la definizione del modello software di IEC 61131-3 sul modello informazioni OPC UA, per delineare il modo standard con il quale i controllori basati su server OPC UA espongono le strutture dati ed i blocchi funzionali ad unità client OPC UA, come HMI. Questo lavoro è riportato nelle specificazioni OPC UA modello informazioni per IEC 61131-3.

Un nuovo scopo è la connettività MES (Manufacturing Execution System), ossia sistema di gestione della produzione, definita come integrazione della macchina e della gestione dati per l'utilizzazione di sistemi MES basati su OPC UA e PLCopen e la specificazione della comunicazione dei Blocchi Funzionali basati su OPC UA.

Il **TC5 - Software sicuro** prepara i consigli per l'applicazione dello standard IEC 61131-3 alle applicazioni relative alla sicurezza, con maggiore attenzione dedicata alle macchine, e ai nuovi standard IEC 61508 e 61511 e IEC 62061. Fornisce inoltre le linee guida per l'utente in relazione all'utilizzo degli aspetti inerenti la sicurezza nelle loro applicazioni, supportati dai loro ambienti di sviluppo, nonché definizioni di libreria per blocchi funzione da utilizzare in applicazioni di sicurezza.

Il **TC6 - XML** lavora sulla specifica di schemi XML per tutti i linguaggi e per progetti globali. Questa specifica costituisce la base per lo scambio e l'associazione ad altri strumenti software, compresi gli strumenti di sviluppo di livello superiore, gli strumenti di simulazione, gli strumenti di documentazione e gli strumenti di verifica.

Eventi promozionali

Un impegno importante di PLCopen consiste nell'informare gli utenti/programmatori circa i vantaggi che può offrire la standardizzazione a livello di programmazione di controllo industriale. Le informazioni sono reperibili sul

- sito web PLCopen: www.PLCopen.org
- pubblicazione di una newsletter elettronica
- pubblicazioni sulla stampa
- partecipazione a conferenze e fiere
- organizzazione di conferenze e seminari propri

Queste attività sono supportate dai comitati di promozione **PC1, PC3, PC4, e PC5**.

Il **PC2 - Common training** ha definito una base comune per la formazione. Nella formazione è compresa la certificazione. In questo modo si possono identificare con semplicità i centri di formazione certificati per lo IEC 61131-3.

I vantaggi dell'appartenenza

L'appartenenza a PLCopen presenta numerosi vantaggi per utenti, venditori e istituti. Unendosi a PLCopen, si fa esplicita dichiarazione di impegno volto all'implementazione di architetture aperte. Si può inoltre partecipare ai comitati e disporre così di informazioni dirette e contribuire attivamente al lavoro svolto; si possono inoltre utilizzare i logo PLCopen.

Il contributo annuo dipende dai diritti di voto e non-voto, dal numero di dipendenti attivi e dalla natura dell'organizzazione. I diversi tipi organizzativi sono:

- Società: organizzazioni commerciali attive nel IC
- Utenti: concentrati sull'applicazione e sull'utilizzo di IC
- Integratori/Rivenditori di sistemi: il valore aggiunto non va ricercato nel IC o nel suo ambiente di programmazione
- Organizzazioni no profit
- Istituti educativi

Per ulteriori informazioni, consultare il sito www.PLCopen.org. Qui si può richiedere la sottoscrizione alla newsletter elettronica per ricevere aggiornamenti sulle attività di PLCopen.