LINGUAGGI DI PROGRAMMAZIONE

NOTE STORICHE

I linguaggi di programmazione del PLC hanno avuto uno sviluppo diverso da quello dei calcolatori.

Il PLC è stato introdotto inizialmente per sostituire i quadri di comando a relè, per cui dovevano essere gestiti non da informatici, bensì da elettricisti con scarse conoscenze di informatica.

Gli elettricisti progettavano i quadri di comando con degli schemi a contatto per cui si organizzò un linguaggio di programmazione del PLC molto simile (LADDER) e oggi molto diffuso. Le industrie costruttrici di PLC scelsero linguaggi proprietari, con conseguenti problemi di portabilità.

La programmazione a contatti non consente tuttavia di sfruttare al meglio la potenza hardware del PLC. Per questo e anche per consentire la programmazione a tecnici informatici e elettronici furono sviluppati in seguito altri due tipi di linguaggi: IL (*Instruction List*) e FBD (*Functional Diagram Block*).

Restava comunque la difficoltà di realizzare una facile ed efficace automazione dei grandi impianti.

Inoltre, a causa della incompatibilità tra i modelli dei diversi costruttori e dell'aumento della complessità dell'automazione, si è rivelata sempre più critica la manutenzione del software di controllo.

Nel tentativo di risolvere questa problematica fu implementato un nuovo linguaggio di programmazione, simile a uno di quelli di alto livello per la programmazione del PC e standardizzato, con sintassi vicina a quella del Pascal: **ST** (*Structured Text*).

Un processo di automazione, e quindi la scrittura di un software di controllo, può essere visto come un **automa a stati finiti** (DES, *Discrete Event System*).

La rappresentazione grafica di un DES mediante i tradizionali ***diagrammi di transizione degli stati*** presentava dei limiti che li rendevano incapaci di una descrizione completa dell'automazione.

I diagrammi erano stati di valido aiuto nella:

- descrizione del funzionamento di un automa a stati finiti
- progettazione dell'hardware.

Ma nella progettazione del software di controllo i limiti dei diagrammi emersero a mano a mano che l'automazione si rendeva sempre più complessa; i diagrammi con consentivano la rappresentabilità di

- più stati attivi contemporaneamente
- processi paralleli
- funzioni di controllo logiche.

Nel solco di questa problematica, nel 1975 in Francia si cominciò a sviluppare il **GRAFCET** (Graphe de Coordination Etapes Transitions), una rappresentazione del processo di automazione mediante un *diagramma funzionale* standardizzato, *capace di rappresentare processi* sequenziali (processi il cui svolgimento avviene a **passi** e la transizione da un passo al successivo è subordinato a certe condizioni).

Il **Grafcet:**

- era un formalismo che consentiva una buona progettazione del controllo degli impianti
- era indipendente dalla tecnologia con cui il controllo sarebbe stato implementato
- forniva una rappresentazione immediata del processo di automazione.

Nel corso degli anni '80 il Grafcet fu ulteriormente sviluppato e il successo fu tale che nel 1987 fu assunto come standard internazionale dall'IEC (Comitato Elettrotecnico Internazionale).

Infine la norma IEC 1131 – 3 del 1993 (recepita in Italia nel 1996), cercando di mettere ordine nei vari linguaggi di programmazione dei PLC, ha incluso anche il Grafcet tra i linguaggi di programmazione con la nuova denominazione di Sequential Functional Chart (**SFC**).

Per la programmazione dei PLC i costruttori possono quindi implementare i seguenti linguaggi:

- IL (Istruction List)
- ST (Structured Text)
- LD (Ladder Diagram)
- FBD (Function Block Diagram)
- SFC (Sequential Functional Chart)