



Lezione #7

Architettura SW e
programmazione di PLC



- OS e modalità operative
- Modalità di esecuzione – ciclo a copia massiva di ingressi e uscite
- Tempi di esecuzione
- Dati forniti dai costruttori
- Soft PLC

Controllori a Logica Programmabile

- Nella scorsa lezione abbiamo introdotto i **PLC** ed esaminato la loro struttura HW

→ Un PLC è un **dispositivo di controllo industriale (ri)programmabile e modulare** specializzato per il **controllo logico/sequenziale**

- In questa lezione ci concentreremo sull'aspetto SW

- Il **Sistema Operativo** di un PLC è un insieme di programmi che consente il corretto funzionamento del sistema e ne supervisiona l'evoluzione
- È conservato in maniera permanente all'interno della ***system memory***

Il SO provvede a

- **Controllo** delle attività ed **esecuzione** dei programmi
- Gestione dei protocolli di **comunicazione** con i moduli inseriti
- **Diagnostica**
 - interna (e.g. controlli di parità sulle memorie)
 - esterna (e.g. controllo della tensione di alimentazione e gestione di *power failures*)
- **Interfaccia** con l'operatore

Il SO inoltre provvede alla gestione delle **modalità operative** del PLC

- Modalità di **programmazione**
 - il SO abilita il download del programma utente in memoria impedendo ogni altra azione

Il SO inoltre provvede alla gestione delle **modalità operative** del PLC

- Modalità di **convalida/debug**
 - il SO permette l'esecuzione del codice disabilitando le uscite (eventualmente passo-passo o con l'inserimento di *breakpoint*)
 - in alcuni casi è possibile abilitare le uscite e simulare via SW l'occorrenza di specifici input

Il SO inoltre provvede alla gestione delle **modalità operative** del PLC

- Modalità di **esecuzione**
 - è la modalità di funzionamento nominale del PLC
 - spesso la commutazione avviene tramite una **chiave fisica**

Modalità di esecuzione

Possibili modalità di esecuzione

(vedi lezione precedente...)

- Periodica
- Ciclica
- Ad eventi

- La modalità più diffusa è quella **ciclica**

Modalità di esecuzione

Ciclo a copia massiva di I/O

- Esecuzione ciclica
→ **ciclo a copia massiva degli ingressi e delle uscite**
- All'inizio di ogni ciclo, il PLC provvede a **copiare ingressi e uscite in un'apposita zona di memoria**
- Il PLC crea un'**immagine dell'impianto** all'inizio di ogni ciclo

Modalità di esecuzione

Ciclo a copia massiva di I/O

- Il SO legge i dati dall'immagine, calcola i valori delle **uscite di controllo** e li **salva** in un'altra area di memoria
- Al termine del ciclo, provvede a **prelevare** tali valori per inviarli agli attuatori
- La **comunicazione** con le periferiche di I/O è gestita in modo **trasparente** sia all'utente che al progettista SW

Modalità di esecuzione

Ciclo a copia massiva di I/O

- La copia massiva garantisce che i valori degli ingressi e delle uscite **restino invariati** durante l'esecuzione dei programmi
- Lettura e scrittura sono **completamente a carico del SO**
- Il sistema è “**cieco**” tra una scansione e l'altra del ciclo
- Esistono istruzioni per realizzare l'**accesso immediato** agli I/O qualora necessario (e.g. in caso di emergenza)
- È possibile gestire degli *interrupt event-driven*

Modalità di esecuzione

Tempi di esecuzione

- Il **tempo di scansione** del ciclo real-time è un parametro importante di un sistema PLC
- Tipicamente i produttori indicano un **tempo “medio” di scansione** (in *ms per kiloword di programma*, con word di 8 o 16 bit)
- In genere questo tempo è dell'ordine di 1÷10 ms

Modalità di esecuzione

Tempi di esecuzione

- Il **tempo di risposta** è invece il massimo intervallo temporale che può trascorrere tra la rilevazione di un evento e l'azione di risposta
- Nel caso peggiore è pari a **2 cicli di scansione**, a cui vanno sommati i ritardi introdotti dai moduli di I/O

Modalità di esecuzione

Tempi di esecuzione

- Nei primi PLC, si utilizzavano microprocessori *custom* con **tempi di esecuzione noti** (le istruzioni dipendevano dal microprocessore stesso)
- I linguaggi erano **interpretati** (ogni istruzione è convertita in linguaggio macchina subito prima dell'esecuzione)
- La scelta di un'esecuzione **periodica time-driven** consentiva di sfruttare al meglio i dispositivi disponibili

Modalità di esecuzione

Tempi di esecuzione

- Il passaggio a microprocessori *general purpose* e la programmazione su PC ha richiesto l'introduzione di **compilatori** e **SO più complessi** capaci di **eseguire in RT** programmi scritti in linguaggi di **alto livello**
- Il rispetto del tempo di esecuzione massimo consentito è garantito da ***watchdog timer***

Dati forniti dai costruttori

- **Memoria dichiarata**
(in genere fa riferimento all'Area Utente)
- **Limite sulla memoria disponibile per altri dati**
- **Massimo numero di ingressi/uscite gestibili**
- **Massimo numero di strutture speciali**
(timer, contatori...)
- **Possibilità di espandere la memoria**

Dati forniti dai costruttori

- Numero di armadi di I/O gestibili direttamente
- Numero di armadi gestibili in remoto e relativi tempi di scansione
- Numero e tipo delle porte disponibili
(seriali, parallele, di rete...)
- Linguaggi supportati
- Possibilità di *multitasking*
- Possibilità di gestire *interrupt*

PLC di sicurezza

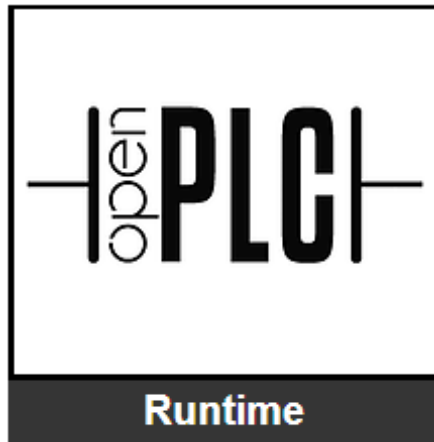
- Esistono PLC **di sicurezza** progettati per l'utilizzo in applicazioni particolarmente rischiose (e.g. l'automazione di presse)
- In genere prevedono una **ridondanza di unità elaborative**, che eseguono uno stesso programma e abilitano le uscite solo se c'è pieno accordo

- È possibile **emulare** il funzionamento di un PLC “classico” su architetture diverse (e.g. su PC)
→ **Soft PLC**
- In questo caso, MMI e *programming unit* sono in genere sulla stessa macchina
- I Soft PLC sono facilmente interfacciabili con software non direttamente legati al controllo dell'impianto (e.g. software aziendali, Office...)

- La principale limitazione dei PC è **la ridotta quantità di I/O**
 - Esistono schede *ad hoc* che sfruttano bus industriali
- Spesso le soluzioni SoftPLC adottano SO **misti**, che gestiscono in RT il controllo del processo industriale e senza specifiche RT le interfacce utente, i software di produttività, ecc.

- Per le esercitazioni di questo corso useremo un *Soft PLC*

→ **OpenPLC**



- OpenPLC è un PLC Open Source aderente allo standard IEC 61131-3
- È composto da un *Editor* e da un programma *Runtime* che simula l'esecuzione del PLC
- Inizialmente sviluppato per Arduino, OpenPLC supporta diverse architetture, inclusi PC Windows e Linux e consente l'utilizzo di dispositivi I/O slave

- OpenPLC ha anche un software SCADA “compagno”
→ **ScadaBR**
- ScadaBR è una *webapp* in grado di interfacciarsi con diversi PLC e usando diversi protocolli (OpenPLC usa **Modbus**)

Soft PLC

ScadaBR

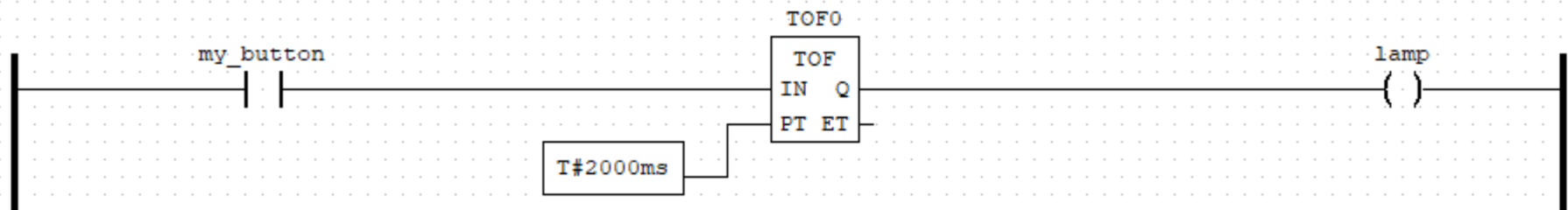
The screenshot shows a web browser window with the following elements:

- Browser Tab:** ScadaBR - 1.0 CE
- Address Bar:** Not secure | 192.168.137.49:8080/ScadaBR/login.htm
- Page Header:** **SCaDaBR** powered by Serotonin's Mango M2M. On the right, it says **ScadaBR - 1.0 CE**.
- User Input:** A form with "User id" containing "admin" and "Password" containing five dots.
- Browser Status:** A green checkmark icon followed by "Chrome 66 on Windows" and "This browser is supported."
- Login Button:** A button labeled "Login" with a help icon.
- Footer:** ©2009-2011 Fundação Certi, MCA Sistemas, Unis Sistemas, Conetec. All rights reserved.



- Installare [OpenPLC](#) + [ScadaBR](#)
- Compilare il [progetto Hello World](#) di OpenPLC

NB: il progetto è scritto in **ladder** e potrebbe apparire poco familiare a prima vista. Dalla prossima lezione introdurremo i vari **linguaggi di programmazione** per PLC



Risorse e Riferimenti

- [1] Cap. 6
- [3] Cap. 8
- [OpenPLC](#)
- [Manuale ScadaBR](#)
- [Introduzione a openPLC](#)



Fine Lezione #7

Architettura SW e
programmazione di PLC

