

PROGETTO DI

“EMULAZIONE DELLE FUNZIONALITA'
DI UN PLC TRAMITE PC MEDIANTE AMBIENTE
LINUX E REALIZZAZIONE DI APPOSITO TOOL PER
LA PROGRAMMAZIONE VISUALE DELLO STESSO,
IL TUTTO SOTTO LICENZA GPL”

STATO DELL'ARTE.

Nelle macchine automatiche e in quelle a controllo numerico le funzione di controllo e di attuazione sono svolte molto spesso da PLC.

Alleghiamo una breve descrizione del PLC:

The PLC mainly consists of a CPU, memory areas, and appropriate circuits to receive input/output data. We can actually consider the PLC to be a box full of hundreds or thousands of separate relays, counters, timers and data storage locations. Do these counters, timers, etc. really exist? No, they don't "physically" exist but rather they are simulated and can be considered software counters, timers, etc. These internal relays are simulated through bit locations in registers.

A PLC works by continually scanning a program. We can think of this scan cycle as consisting of 3 important steps. There are typically more than 3 but we can focus on the important parts and not worry about the others. Typically the others are checking the system and updating the current internal counter and timer values.

Il che dimostra come il funzionamento interno del PLC sia molto semplice.

Quello che piace dei PLC è che sono altamente affidabili e non si bloccano praticamente mai.

Per quanto riguarda l'aspetto di programmazione dei sistemi PLC, è stato messo a punto uno standard (IEC 61131) da parte dei vari produttori di PLC, con il contributo di università ed utenti finali. La terza parte di questo standard descrive proprio i linguaggi di programmazione per PLC :

- Ladder Diagram(LD)
- Sequential Function Charts (SFC)
- Function Block Diagram (FDB)
- Structured Text (ST)
- Instruction List (IL)

I primi 3 linguaggi sono di tipo visuale a blocchi e sono caratterizzati (in particolare Ladder) da

facilità di utilizzo, mentre gli ultimi due sono più simili a linguaggi di programmazione (Instruction List simili all'assembler e Structured Text simile al Pascal).

Sul mercato attualmente esistono strumenti CASE (Computer Aided Software Engineering environment) come ISAGRAF® che prevedono il supporto per tutti e cinque i linguaggi previsti dallo standard. La programmazione dei PLC viene così resa più semplice e accessibile anche a personale addetto, che non abbia specifiche competenze di programmazione nei linguaggi informatici.

Fra i linguaggi previsti dallo standard il più diffuso è Ladder (che per la sua natura orientata ai circuiti risulta particolarmente comprensibile a periti industriali e alle altre figure tecniche).

Aspetti a favore dell'impiego dei PLC:

1. costituiscono uno standard;
2. sono reperibili in molti mercati;
3. i manutentori ed i tecnici delle aziende clienti o coloro che effettuano l'assistenza sono già formati sulla tecnologia a PLC;
4. sono di facile gestione e programmazione.

Di contro i PLC presentano i seguenti principali problemi:

- bassi clock di campionamento e impossibilità di funzionare in real time;
- per risolvere controlli di media complessità i PLC presentano un costo elevato (4/5000 euro o superiore);
- impossibilità di utilizzo di browser per collegamenti alla rete internet;
- impossibilità di elaborazione;
- ridondanza dell'hardware a bordo macchina.

Bibliografia e link collegati:

<http://www.tech.it/twire/mesi/free/tw-9807/aless.htm>

http://www.plcopen.org/intro_iec/intro_iec_61131_2.htm

http://www.rational.com/media/products/rosert/rosert_IORC_paper.pdf

http://www.sixnetio.com/html_files/oems/ipm_isagraf.htm

<http://www.nautsilus.ru/engl/is-linux-e.htm>

SOLUZIONI ATTUALMENTE PRESENTI SUL MERCATO.

Per la risolvere le problematiche legate all'uso dei PLC sono nate diverse soluzioni proprietarie, che implementano lo standard IEC 61131. Queste soluzioni proprietarie spesso utilizzano versioni di Linux embedded (spesso con caratteristiche real time come RTAI e RTLinux) e metalinguaggi intermedi (fra linguaggio visuale e macchina).

Commercialmente è possibile trovare soluzioni di emulazione PLC (softPLC). Che utilizzano metalinguaggi intermedi per la produzione di linguaggi macchina a PC impiegando Ladder ricorrendo alle seguenti aziende:

Sixnet, <http://www.sixnetio.com/index.html>

EAC, <http://www.eac-com.de/flyer/dflymhpc.htm>

AuTech, <http://www.au-tech.de/>

Autronix, <http://www.autronix.de/>

Ed i seguenti metalinguaggi per Ladder e IEC 61131:

<http://www.nautsilus.ru/engl/is-linux-e.htm>

<http://www.eilers.net/sps/>

ed i software di interfaccia:

<http://sites.inka.de/heisch/en-index.html>

<http://libnodave.sourceforge.net/index.php>

PROPOSTA DI INNOVAZIONE.

Come già evidenziato in precedenza, al momento attuale, sono presenti sul mercato soluzioni **proprietarie** di emulazione PLC basate su architetture tipo PC dotate di sistemi operativi embedded. A queste soluzioni, spesso denominate “softPLC” (per indicare appunto l'emulazione via software delle funzionalità tipiche dei PLC), sono spesso associati sistemi/tools di programmazione visuale a blocchi, anch'essi proprietari (come il già citato ISAGRAF®), il cui scopo è di rendere facile la programmazione.

Lo scopo del nostro progetto è la messa a punto di un'architettura analoga ai già esistenti softPLC, che possa quindi unificare i vantaggi dei due mondi (PLC e PC con sistemi embedded).

Una simile architettura realizzerà l'emulazione delle funzionalità di un PLC tramite hardware tipo

PC 104 (o analoghi) e manterrà la facile programmazione visuale (con LADDER ed eventualmente gli altri linguaggi previsti dallo standard), oltre a mantenere la flessibilità tipica dei sistemi embedded.

La vera novità consiste nel fatto che i tool realizzati saranno resi disponibili con **licenza Open Source compatibile con la GPL**. L'utilizzo di tale licenza consente il libero utilizzo, la personalizzazione del prodotto a proprio piacimento, nonché una sicura longevità. La comunità di sviluppatori Open Source potrà in fasi successive aggiungere funzionalità, supporto per nuovi linguaggi, e magari adattare il prodotto a usi per il quale non era stato inizialmente concepito.

Il tool di sviluppo visuale sarà organizzato in maniera modulare (architettura a plug-in) in maniera tale da consentire la facile aggiunta di funzionalità.

Studio di prefattibilità.

L'hardware

L'hardware presente in un PLC è molto semplice e quindi semplicemente «emulabile» tramite computer.

L'emulazione della CPU e della memory areas è evidentemente banale.

Si possono inoltre emulare i dispositivi di I/O:

- digitali,
- analogici
- porte seriali o parallele
- porte ethernet
- dispositivi tipo CAN bus o bus di campo;
- ecc...

L'unico problema potrebbe essere sul numero di dispositivi simultaneamente installabili su di un PC104, ma anche questo aspetto è di facile soluzione considerando che esistono molti costruttori di sistemi PC104 e lo standard è diffuso e quindi farsi realizzare schede ad-hoc.

Inoltre occorre sviluppare una piattaforma che, similmente al PLC, sia altamente affidabile e che non si blocchi praticamente mai. Il Sistema Operativo in grado di garantire questa altissima stabilità/affidabilità è Linux.

Un altro aspetto da simulare è la sequenza azione/reazione del sistema di controllo di un PLC. Occorre disporre di una altissima «prevedibilità» della sequenza azione->reazione. I PLC sono senz'altro molto stabili su questo aspetto, essendo stati sviluppati ad-hoc per compiti di controllo, ma anche utilizzando Linux (per alte latenze ~1ms) e alcune versioni embedded come RTAI (per basse latenze ~10us) sono perfettamente adeguate per questi compiti.

Programmazione visuale.

Per quanto riguarda l'aspetto di programmazione «visuale», come già accennato in precedenza, con la maggior parte dei sistemi PLC è reso disponibile per la programmazione, un «linguaggio» visuale chiamato «Ladder». Anche se vi può essere qualche piccola differenza fra le implementazioni delle case costruttrici (dialetti leggermente diversi), tale linguaggio è di fatto molto diffuso (è fra i linguaggi previsti dallo standard IEC 61131).

Si prevede inizialmente quindi di implementare Ladder come principale “linguaggio” di programmazione, ma la modularità dell'architettura consentirà, la successiva facile aggiunta di altri linguaggi (quelli previsti dallo standard ed eventualmente altri), nonché di altre funzionalità.

Quello che faremo è allora realizzare un «tool» di sviluppo che implementa l'architettura descritta, tale linguaggio di programmazione visuale e permette l'eventuale aggiunta successiva del supporto ad altri linguaggi.

La utilizzo di un metalinguaggio di passaggio tra il ladder e il codice macchina sembra essere molto utile. Consente infatti:

- la realizzazione di un'architettura a strati
- la realizzazione di un'architettura modulare: lo strato di più alto livello consente la traduzione da uno o più linguaggi dello standard al formato intermedio e quindi la più facile aggiunta di altri linguaggi
- migliore ottimizzazione del codice macchina grazie all'ottimizzazione/traduzione effettuata in più passate (a partire dal codice intermedio).

L'impiego di metalinguaggi è utile e necessario per mantenere l'impiego degli strumenti visuali come Ladder e degli altri linguaggi previsti dallo standard IEC 61131 per la programmazione (ISAGRAF® uno strumento di programmazione commerciale molto diffuso, per esempio utilizza un formato intermedio chiamato TIC (Target Independent Code)).

Quello che potremmo fare è quindi un sistema che, dato un diagramma di programmazione in ladder, restituisce un metalinguaggio interpretabile per poi ottimizzare il tutto passando dal metalinguaggio direttamente al codice C.

Il tool LADDER permetterà di salvare il programma visuale e quindi lo esporterà nel metalinguaggio, il passo successivo sarà quello di tradurre il metalinguaggio in C e compilarlo.

Questo per poter separare la parte di progetto del tool grafico dalla parte di implementazione del PLC tramite PC104.

Periferiche pilotabili.

Le periferiche che si potranno pilotare sono principalmente le seguenti:

* I/O digitali/analogici

* Schede di rete ethernet

* porte seriali, parallele ecc.

* schede can bus

* schede video

* schede sonore

* frame grabber ed encoder MPEG

* schede GSM

etc

LICENZA DEL PRODOTTO

Proponiamo di rilasciare il tool sotto licenza GPL in quanto dal punto di vista dell'utilizzatore del software è senz'altro la migliore. Questa licenza permette infatti di portare all'interno dell'azienda un know-how maggiore rispetto ad una licenza di tipo «proprietario» in quanto vengono rilasciati i codici sorgenti, ed è perciò possibile apportare qualsiasi personalizzazione e modifica.

Utilizzare licenze GPL non significa necessariamente che il prodotto software sia gratis e nemmeno che questo verrà pubblicato sul web. È chiaro che se verrà pubblicato potremmo anche usufruire degli aiuti della comunità del software libero che son sicuro troverà il progetto interessante e quindi potremmo avere un sistema ancora più testato e sicuro.

Un simile prodotto con licenza libera risulta ancora più utile per un paese come l'Italia, che nel settore dell'automazione dipende in maniera pressochè completa dall'estero.

Non ci risultano infatti esistere aziende italiane produttrici di PLC, e la diffusione di sistemi embedded al posto dei PLC, spesso è limitata dalla complessità della programmazione.

Il nostro progetto ovvierebbe ad queste difficoltà:

- la programmazione diverrebbe facile come quella dei PLC, pur mantenendo le potenzialità dei sistemi embedded
- Non si sarebbe più vincolati ad un singolo produttore (essendo disponibili i sorgenti del software e quindi modificabili da chiunque ne abbia le capacità)

MODALITÀ DI PARTECIPAZIONE

Phiap Srl si è fatta promotore presso diverse imprese del settore del sistema PC embedded qui presentato, nelle prime visite ha ottenuto, da parte di queste l'interesse per il progetto.

Gli accordi che regolano le fasi future sono:

1. promuovere una serie di riunioni per la messa a punto dei contenuti tecnici e la preventivazione dei costi in modo da predisporre una sorta di capitolato tecnico su cui raccogliere le adesioni;
2. predisporre lo strumento consortile per lo sviluppo del progetto;
3. predisporre la strategia di sviluppo del progetto tale per cui identificare i finanziamenti agevolati e le modalità di copertura dei costi precedenti alla erogazione del finanziamento;
4. tempistica del progetto e relativi obiettivi realizzativi disposti in un grafico di GANTT;
5. chiusura del consorzio e del progetto o eventualmente rilancio di un nuovo progetto.