



Open Source e Automazione

Lo stato dell'arte



Scopo della presentazione

- Qual è lo stato dell'arte del software libero/Open Source nel settore dell'automazione industriale e degli impianti?
- Status e numerosità dei progetti OS, anche rispetto ad altri settori e campi di applicazione (Internet, networking, desktop)
- Quindi: carrellata sulle possibilità e sui progetti OS nel campo dell'automazione – Esistono progetti OS che possono essere usati e implementati professionalmente nel settore Automazione?



Cos'è il software libero?

- L'espressione "software libero" descrive un particolare modo di "licenziare" il software, che permette ampie libertà all'utente.
- L'aspetto più evidente è l'accessibilità dei sorgenti (open source)
- La contrapposizione usuale è quella con il software "proprietario" - licenze con limiti ben precisi



Cos'è il software libero?

- Il software Open Source è distribuito con una ben precisa licenza, NON E' di pubblico dominio.
- Un software (o altra opera dell'ingegno) che sia di pubblico dominio per qualsiasi motivo (es. Scadenza del Copyright per le opere letterarie) può essere utilizzata da tutti senza alcuna limitazione, incorporato etc.
- PS: rimandiamo ad altra sede le distinzioni tra Open Source / FOSS / etc etc ;)



La licenza GPL

La GNU General Public License (GPL)

E' la licenza più famosa e utilizzata per distribuire il software libero, ma non certo l'unica. Creata dalla FSF (Usata attualmente la versione 2, in gestazione la 3).



La licenza GPL

Concetti di base della GPL: Libertà di uso, vendita, cessione, distribuzione, modifica del software licenziato a condizione che venga distribuito insieme il sorgente.



La licenza GPL

La GPL è stata definita "virale": chiunque ha la libertà di effettuare modifiche al software, ma in caso di redistribuzione (gratuita o commerciale che sia) è obbligato a rendere disponibile il sorgente.



La licenza GPL

Libero è diverso da gratis: la licenza non mi impedisce in alcun modo di rivendere il prodotto (anche senza modifiche) a condizione che il sorgente sia reso disponibile.



La licenza GPL

- le librerie, anche se hanno licenza GPL, possono essere linkate da software proprietario. Diciamo che in questo caso il “contagio” non si estende (Esempio: libnodave per la comunicazione PLC)
- LGPL (Lesser GPL): in questo caso – oltre alle libertà concesse dalla GPL – posso anche incorporare il prodotto in un mio prodotto proprietario e distribuirlo senza rendere il sorgente disponibile.
- La GPL è già stata messa alla prova legalmente (D-Link x appliances)



La licenza GPL: da GPL2 a GPL3

- GNU introduce la GPL3
- Differenze: più “ideologica” e programmatica
- Obiettivi: TiVoizzazione e DRM. Per i sistemi embedded potrebbero esserci conseguenze.



Carrellata dei progetti Open Source più significativi



Sistemi Operativi Real Time

- Definizione di Real Time: sistema capace di eseguire processi che effettuano le operazioni a ben precise scadenze temporali (Deterministico): non vuol dire “molto veloce”
- Linux RTAI www.rtai.org

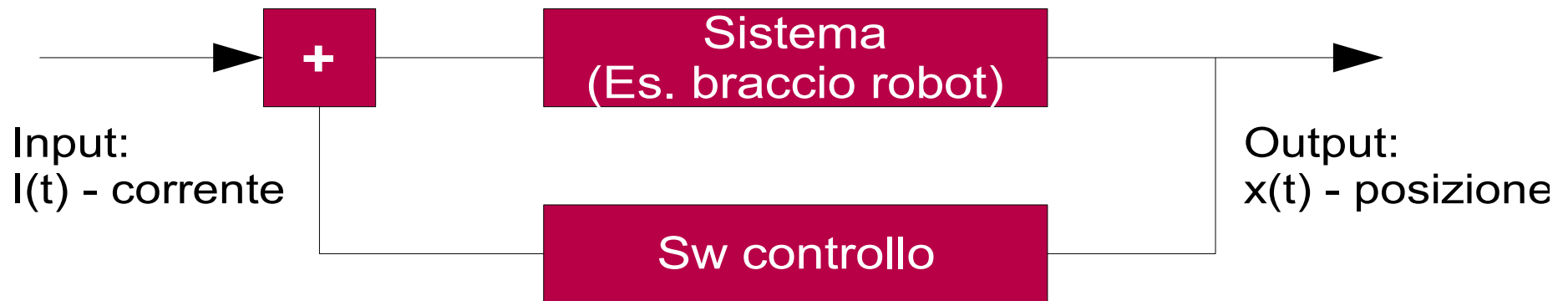


Sistemi Operativi Real Time

- Il kernel Linux standard permette il cosiddetto “soft realtime”: il fallimento del real time per un processo non comporta il fallimento dell'applicazione.
- Il realtime “soft” è implementato per mezzo delle politiche di scheduling (si settano in user-space).
- Il kernel Linux RTAI permette l' “hard realtime” : rispetto assoluto delle scadenze temporali programmate (prestazioni fino a 1 microsecondo di latenza)



Sistemi Operativi Real Time



ESEMPIO DI APPLICAZIONE: controllo in retroazione di una macchina (es. Braccio robot), dove è necessario rilevare la posizione ad intervalli di tempo precisi e costanti.



Sistemi Operativi Real Time

- Progetto sviluppato al Polimi – Dip. Ing. Aerospaziale (prof. P. Mantegazza)
- Architettura: HALd (hardware abstraction layer tra l'hardware e il kernel). Sopra HAL girano il kernel “non-RT” e poi le applicazioni. I processi RT (che possono anche comunicare con il kernel “non-RT” o le applicazioni non rt), girano sopra lo strato HALd, allo stesso “livello” del kernel.
- I processi RTAI possono essere sviluppati in user space o in kernel space: il tradeoff è: maggior difficoltà di programmazione ma maggiori prestazioni (microsecondo contro millisecondo di latenza massima) per i processi in kernel space.

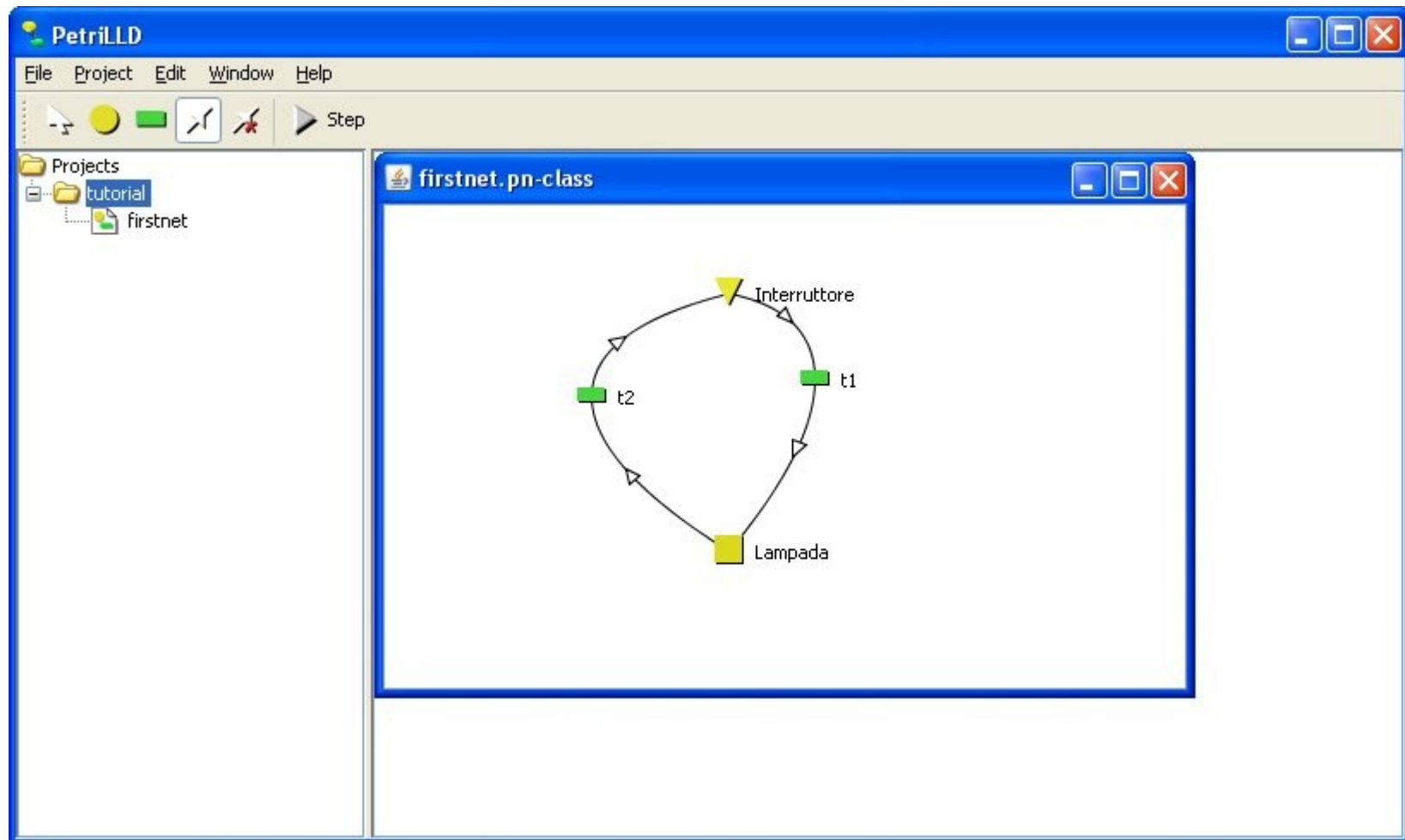


Strumenti per lo sviluppo

- Classic Ladder
<http://membres.lycos.fr/mavati/classicladder/>
- Ambiente di sviluppo Ladder, non paragonabile agli ambienti professionali.
- PetriLLD <http://petrilld.sourceforge.net>
- Modellazione del software tramite reti di Petri ed esportazione in formati standard PLC.

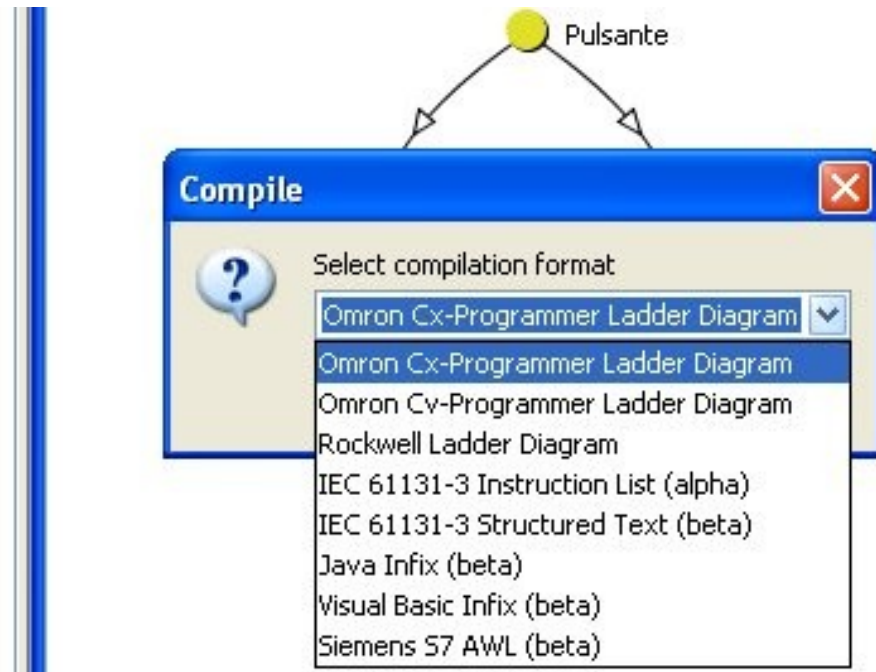


Strumenti di sviluppo - PetriLLD



Strumenti di sviluppo - PetriLLD

- Possibilità di esportazione



Librerie di comunicazione con i PLC

libnodave

- Libreria (biblioteca) di funzioni per la comunicazione con PLC Siemens S7 via adattatori MPI/PPI seriali o via Ethernet (Cpx43 montata sul PLC o collegamento IBH/MHJ) – Alternativa a PRODAVE
- Target: sviluppatori
- GPL: sorgenti e precompilati disponibili per Linux (.so) e Win32 (.dll)
- Supporto per .NET
- Linguaggi supportati:
C,C++,C#,Delphi,Pascal,Perl,VB(A),PHP



Librerie di comunicazione con i PLC

- Interfaccia MPI – usata per la programmazione del PLC ma anche per lo scambio dati. Strato fisico: doppino, differenziale non dissimile da RS485, baud rate normalmente 187,7 kBaud -> necessità di un adattatore Siemens con CPU: MPI dalla parte PLC, altro protocollo lato PC (ce ne sono almeno due, la libreria supporta solo quello più semplice (No CRC) per il momento)
- Nessuna documentazione ufficiale sul protocollo -> sorgenti www.runmode.com e reverse engineering.



Librerie di comunicazione con i PLC

- Supporto: Linux e Win32 (compilabile dai compilatori Borland C su command line)
- Supporto adattatori: per il momento
 - gli adattatori seriali
 - 6ES7 972-0CA22-0XAC
 - 6ES7 972-0CA23-0XAC
 - 6ES7 972-0CA11-0XAC
 - adattatore teleservice
 - 6ES7 972-0CA33-0XAC
- CPU: Maggior parte delle CPU serie 300 e 400 (Anche VIPA Speed7 CPU 318)



Libnodave: esempio minimo di programmazione

```
Inizializzazione interfaccia seriale:  
fd=setPort("/dev/ttyS1","19200','0');  
Inizializzazione di una interfaccia dave:  
di=daveNewInterface(fd,"Nome");  
Inizializzazione dall'adattatore MPI:  
error=daveInitAdapter(di);  
Creazione di una connessione:  
dc=daveNewConnection(di, 2);  
Connessione a un PLC: daveConnectPLC(dc);  
Scambio dei dati con un PLC:  
daveReadBytes(dc,daveFlags,0,0,24);  
Scambio dei dati con un PLC:  
daveWriteBytes(dc,daveFlags,0,0,24,buf);  
Disconnect: daveDisconnectPLC(dc);  
daveDisconnectAdapter(di);
```



Altre librerie di comunicazione

- AS511 <http://as511.sourceforge.net/>
- libopensrtp (GE Fanuc)
<http://libopensrtp.sourceforge.net>



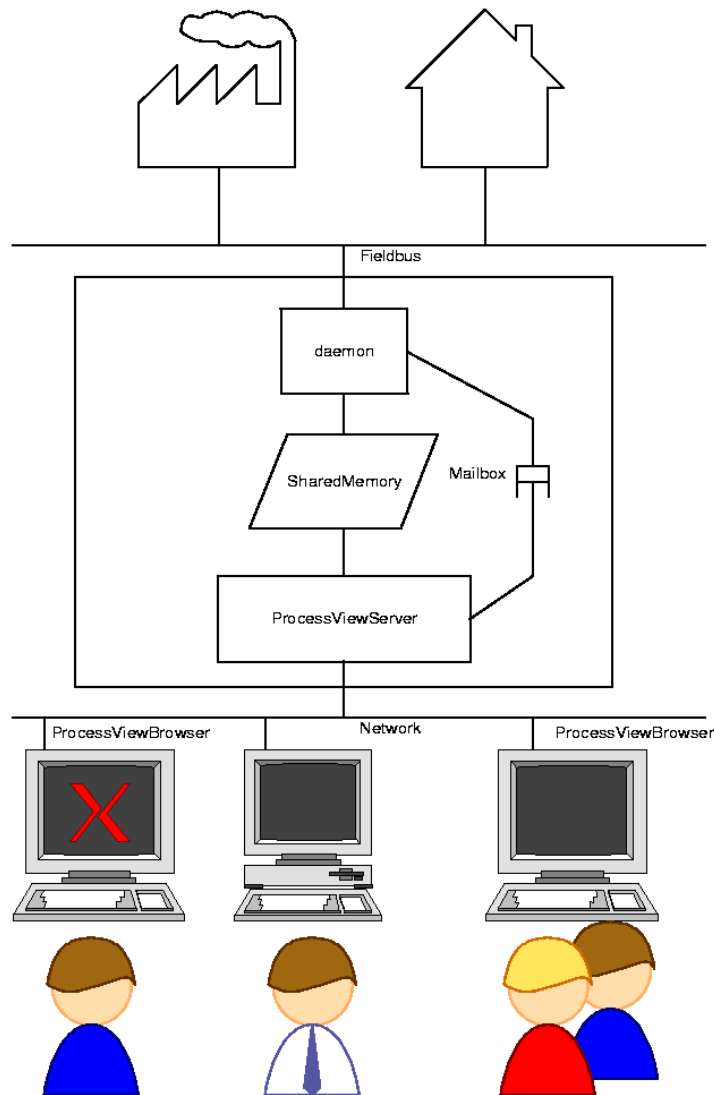
HMI e SCADA - pvbrowser

- L'unico progetto realmente utilizzabile (e dispiegato sul campo). (www.pvbrowser.org)
- Architettura client-server. Fornisce un browser dedicato per applicazioni SCADA, che accede a “URL” del tipo pv://host
- Quindi modello “thin client” (software) vs “fat client”, come ad esempio WinCC/Modicon (anche se sono presenti moduli per l'accesso via HTTP)



HMI e SCADA - pvbrowser

ProcessViewBrowser - principle



Tratto da: www.pvbrowser.org

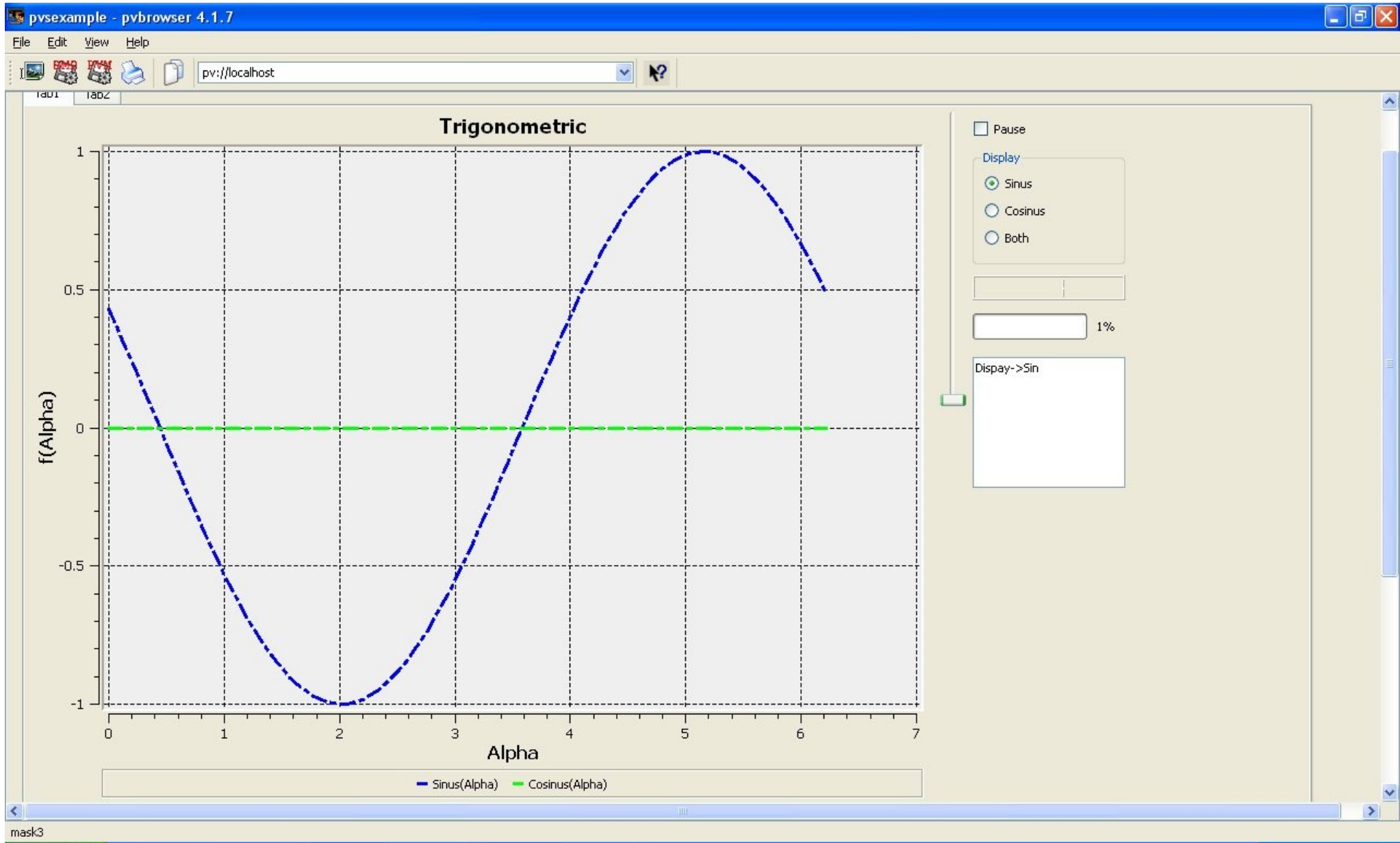


HMI e SCADA - pvbrowser

- Un'alternativa per usare il web per queste applicazioni è per es. Java (applets) o controlli ActiveX. Non molto performante e tempi di caricamento lunghi.
- I server sono sviluppati in ANSI-C; il codice viene autogenerato per la maggior parte.
- Presente un editor grafico per lo sviluppo delle pagine (sinottici).
- Multiplatforma (Win32 / Linux)



HMI e SCADA - pvbrowser



HMI e SCADA - pvbrowser

- Tecnicamente: lo scambio di informazioni riguarda solo la manipolazione dei widget QT (oggetti nelle pagine) per mezzo di un protocollo testuale leggero, mentre l'”albero dei widget” (WT widget tree) viene mantenuto in memoria dal browser (lato client)
- Tecnicamente: programmazione di un server
 - Codice ANSI-C generato automaticamente dal sistema per la maggior parte, da completare con le azioni relative agli oggetti visualizzati



HMI e SCADA - pvbrowser

- Collegamento col campo: possibilità di usare librerie esterne C/C++ (Modbus e Siemens già collaudate)
- Esempio minimo:

```
valore=modbus.readBit(offset, numero);
```



HMI e SCADA - Scada-TGZ

- scada TGZ <http://scada-tgz.sourceforge.net/>
- al momento progetto molto limitato
(eventualmente da usare per esperimenti)
- Piattaforma .NET su Win32 e Linux (via Mono)



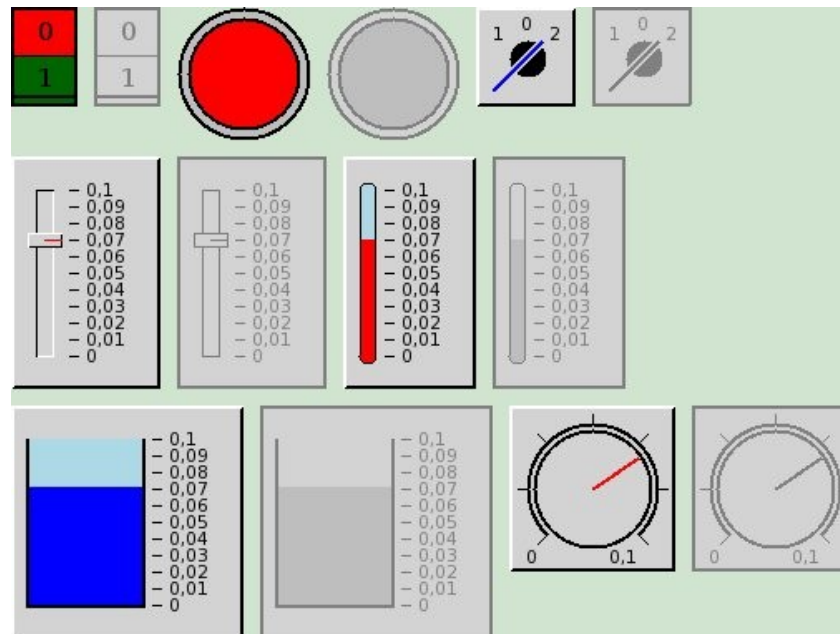
HMI e SCADA - Scada-TGZ

- 2 Server di comunicazione presenti:
 - Modbus RTU over RS232
 - GeSNP (GE Fanuc) over RS232
 - Testati con GE Fanuc 90-30



HMI e SCADA - Scada-TGZ

- 7 Controller (widget) grafici sviluppati
- Interruttore, Lampada, Tristat (Selettore a tre stati), Slider, Term (Livello ad es. Di temperatura), Serbatoio, Potenzziometro.



HMI e SCADA - Scada-TGZ

- Scada-tgz
- Punto positivo: multipiattaforma
- Negativi: ancora immaturo
- Richiede molta programmazione per il server:
ASP.NET



SCADA: Altri progetti

- Altri progetti in fase per lo più embrionale:
- visual (pre-alpha)
- phpmyscada (pre-alpha)
- free SCADA
<http://sourceforge.net/projects/free-scada/>
- (Beta ferma a febbraio 2007). Basato su OPC servers, bypassa le complicazione della comunicazione col campo, però gli OPC server vanno comunque acquisiti...
- Usabile, ma non in produzione; editor grafico
- GUIDE <http://sourceforge.net/projects/guide/>



Linux CNC

- Linux CNC
- Distribuzione Linux dedicata al controllo delle macchine a controllo numerico



Domotica / Building Automation

- Building Automation
 - X10 home automation system (pre-alpha)
<http://sourceforge.net/projects/x10homecontrol/>
 - EIB control <http://eibcontrol.sourceforge.net>
 - Software di comunicazione con il bus KNX, sviluppato in ambiente Linux. Base per lo sviluppo di un sistema di supervisione su PC, accessibile da illimitati client.



Domotica / Building Automation

- Architettura di EIB Control
- Componenti:
 - homedriver: driver di comunicazione con il bus (solo via seriale)
 - homeserver: interfaccia di comunicazione tra homedriver e il software applicativo (client)
- Quindi il flusso di informazione è:
 - KNX bus<->Interfaccia seriale<->homedriver<->homeserver<->Client



Domotica / Building Automation

- Due interfacce di comunicazione con i client:
 - Applicazione TCP/IP (porta 8081)
 - HTTP (porta 8080)



Conclusioni

- Nell'automazione diffusione più limitata di altri settori, per vari motivi (branding, specializzazione ...)
- PLUS: Sistemi operativi anche RT per sistemi embedded e controllori (Linux / BSD)
- PLUS: Networking e sicurezza
- PLUS: Alcuni progetti SCADA/Supervisione
- MINUS: Ambienti di sviluppo PLC!
- MINUS: librerie di collegamento con PLC (funzionano ma basate su reverse engineering)
- = : Building Automation

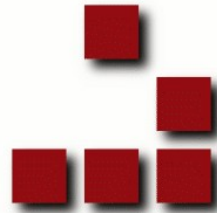


SAVE 2007

Grazie!



www.plcforum.it



www.studioag.eu