PARLANO DI NOI 2003

RACCOLTA ARTICOLI TECNICI SU RIVISTE DEL SETTORE

SCIENZA E TECNOLOGIA AL TUO SERVIZIO !

INTELLISYSTEM TECHNOLOGIES

Prefazione

Intellisystem Technologies è da sempre presente in riviste prestigiose di carattere nazionale ed estero, presentando sempre nuove idee prodotti e soluzioni all'avanguardia per divulgare le nuove tecnologie. Il presente volume rappresenta la raccolta completa di tutte le pubblicazioni della nostra azienda inerenti l'anno 2003.

Presentazione Azienda

Intellisystem Technologies nasce nel 2000 come engineering office per apportare un contributo concreto allo sviluppo ed alla diffusione di sistemi che riescano ad interagire con l'uomo per soddisfare quanto più possibile le sue necessità. Nel 2003 diviene una società il cui obiettivo fondamentale è la progettazione, lo sviluppo e la commercializzazione di soluzioni Hi-Tech inerenti problematiche di tipo informatico, elettronico, di telecomunicazioni e di qualsiasi altra disciplina che necessiti di un continuo supporto alle nuove tecnologie.

La nostra filosofia è quella di indirizzare a tutti le nuove tecnologie presenti sul mercato e di abbinarle al rigore scientifico, espandendo così in tutte le direzioni gli apporti di utilità che da essa ne possono derivare. La ricerca scientifica è il piedistallo su cui poggia l'intento di evoluzione della nostra azienda, completandosi e complementandosi sinergicamente con attività collaterali di Sviluppo, Diffusione, Vendita ed Assistenza: RDDSA (*Research, Development, Diffusion, Selling and Assistance*).

Intellisystem Technologies vanta al suo attivo numerose attività che spaziano dalla ricerca nell'ambito della fisica nucleare alla progettazione di sistemi "intelligenti", alla didattica mirata alle specificità, alla pubblicazione scientifica, alla consulenza di piccole, medie e grandi aziende, alla progettazione ed alla realizzazione di sistemi software ed hardware per lo sviluppo informatico dei sistemi di monitoraggio, controllo industriale, militare e domestico. Grazie al suo Team di ingegneri, ricercatori e sviluppatori, è oggi in grado di offrire alla propria clientela soluzioni tecnologicamente all'avanguardia totalmente personalizzabili in funzione delle più variegate esigenze.

La nostra azienda ha istituito una rete di penetrazione nel mercato nazionale avvalendosi della collaborazione di aziende partner in grado di soddisfare la risposta a qualunque quesito di necessità, prestando assistenza consultiva, didattica e operativa in qualunque parte del territorio Nazionale con mire espansionistiche rivolte all'estero.

La costante presenza e partecipazione a progetti prestigiosi presso autorevoli istituti di ricerca nazionali ed esteri (quali: INFN "***Istituto Nazionale di Fisica Nucleare***", CERN "***Organisation Europèenne pour la recerche nuclèare*, *Ginevra, Svizzera***", ERSF "***European Synchrptron Radiation Facility, Grenoble, Francia***", KVI "***Kernfysisch Versneller Institut,Groningen, Netherlands***.", ecc.), permette ad Intellisystem Technologies di fornire un costante impulso alla diffusione ed integrazione delle più moderne ed innovative tecnologie presenti nel mercato.

Intellisystem Technologies vanta partnership di carattere scientifico e tecnico-commerciale con aziende leader quali: Telecom Italia, TIM Business, ASTRA, Globalstar, mettendo a disposizione il proprio knowhow tecnologico ed i propri prodotti ad alto contenuto tecnologico.

Tra i nostri clienti spiccano: Honeywell, Alcatel Lucent, la Presidenza del Consiglio dei Ministri, il Ministero dell'Ambiente - Area marina protetta "Isole Ciclopi" Acitrezza (CT), l'orto Botanico dell'Università degli studi di Catania, aziende nazionali ed estere che lavorano per ERG petroli, ecc.

" Intellisystem Technologies nata dall'amore per la ricerca scientifica applica le moderne tecnologie per il benessere di tutti. "

Ing. Cristian Randieri
Amministratore Unico

Indice

FE

REALIZZAZIONI PRATICHE • TUTORIALS • RADIANTISTICA • COMPUTER HARDWARE

Fare ELETTRONICA

N° 212 - FEBBRAIO 2003 - ANNO 19

€ 4,13 - Frs 8,00

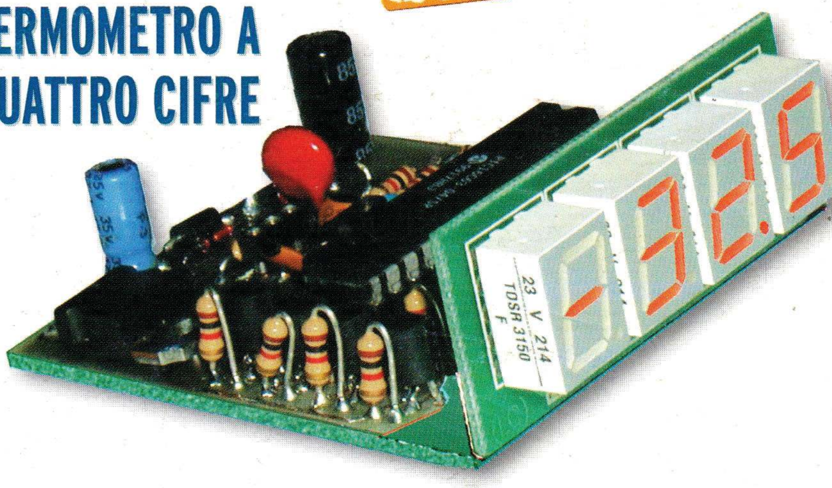
ALL'INTERNO LE PAGINE DI:

MHz

FareELETTRONICARADIO

NUOVA GRAFICA
NUOVI CONTENUTI

TERMOMETRO A QUATTRO CIFRE



ELETTRONICA GENERALE

- ANTIUMIDITÁ PER MURATURE
- AMPLIFICATORE PER MICROFONO AD ELEVATA SENSIBILITÁ
- TERMOMETRO ELETTRONICO A COLONNINA
- CIRCUITO ELETTRONICO ANTIBALBUZIE

TUTORIAL

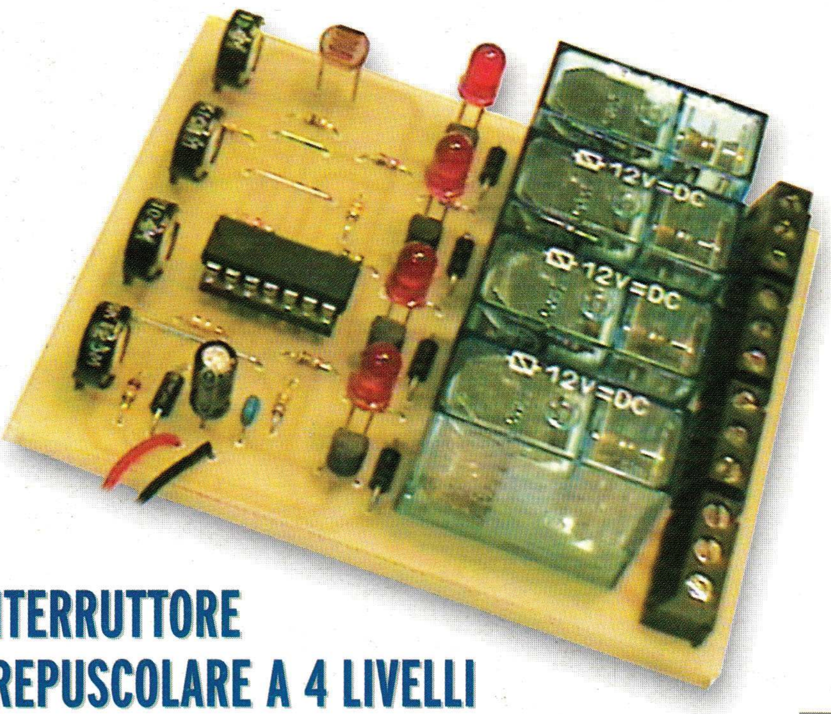
- LE INTERFACCE SERIALI RS-422 E RS-485

HARDWARE

- INTRODUZIONE AI CCD (IIª parte)
- RABBIT BY EXAMPLE
- GENERARE SEGNALI VIDEO IN REAL-TIME UTILIZZANDO UN PIC16F84

AUTOMAZIONE

- RECS 101



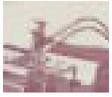
INTERRUTTORE CREPUSCOLARE A 4 LIVELLI

Spedizione in A.P. - 45% - ART. 2 COMMA 20/B LEGGE 662/96 - Filiale di Milano. In caso di mancata consegna restituire all'editore che si impegna a pagare la relativa tassa presso il CMP di Roserio - Milano

DTP studio editrice

30212

9 771591 227008



RECS 101: UN WEB SERVER EMBEDDED PER APPLICAZIONI DI CONTROLLO REMOTO TRAMITE TCP/IP

prima parte

di Cristian Randieri
randieri@intellisystem.it

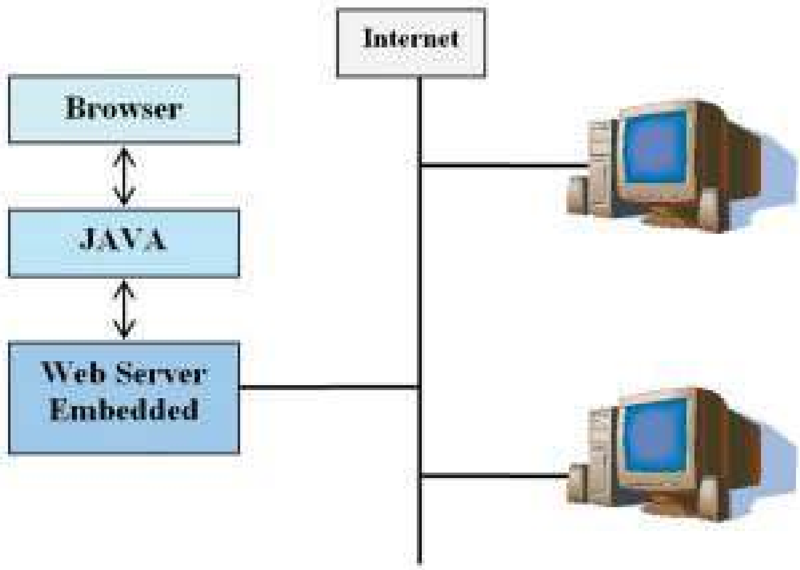
Un web server embedded è un web server gestito all'interno di un sistema embedded caratterizzato da risorse di calcolo limitate capace di gestire documenti ed applicazioni web. L'integrazione di un web server all'interno di un dispositivo di rete fornisce un'interfaccia utente grafica costruita tramite il linguaggio HTML. L'applicazione della tecnologia Web ad un sistema embedded permette la creazione di interfacce utente che sono user friendly, a basso costo, cross platform, e network ready. Aggiungendo alle potenzialità di un web server embedded la tecnologia Java si ottiene un sistema capace di gestire vere e proprie applicazioni che possono essere programmate con un linguaggio ad alto livello quale il Java. Scopo di questo articolo è quello di presentare una soluzione web server embedded capace di gestire la Java Virtual Machine. Viene presentata l'architettura di un web server embedded che può fornire un'interfaccia API (Application Program Interface) semplice e al tempo stesso potente. In particolare si discute la progettazione e l'implementazione di RECS 101, che è un web server embedded (prodotto da Intellisystem Technologies) sviluppato al fine di poter gestire piccole applicazioni di controllo remoto. In conclusione vengono presentate alcune applicazioni pratiche del dispositivo, che prevedono la realizzazione di circuiti elettronici d'interfaccia, uno studio riguardante dei test di performance di RECS 101 ed un'analisi delle problematiche di protezione da attacchi alla sicurezza da parte di hacker.

I SISTEMI WEB SERVER EMBEDDED ED INTERNET

Il World Wide Web (o Web) è in continua evoluzione. Appare chiaro ed evidente che tale tecnologia assume delle nuove funzionalità che vanno molto oltre la semplice visualizzazione delle pagine Web. Per molte applicazioni commerciali e scientifiche il browser web è diventato uno standard per lo sviluppo di interfacce utente di numerose applicazioni. Questo perché i browsers web sono capaci di fornire interfacce GUI a varie applicazioni client/server senza il bisogno di andare ad implementare del software per il lato client. Negli ultimi anni è sempre più cresciuto il numero di tecnologie web che possono essere applicate ad elementi gestibili dalla rete.
Come ben noto la maggioranza delle reti di computer viene gestita mediante il protocollo TCP/IP.
In realtà TCP e IP sono due protocolli utilizzati per interconnettere le reti. TCP sta per Transport Control

Protocol e IP sta per Internet Protocol. Per essere più precisi quando si parla di protocollo TCP/IP si intende una suite di protocolli che è stata promossa come standard di Internet dall'Unites States Secretary of Defence. Obiettivo della suite di protocolli TCP/IP è quello di consentire la comunicazione di reti simili o eterogenee. Mediante i protocolli i progettisti possono specificare i processi di comunicazione senza essere a conoscenza dei dettagli delle entità che compongono la rete. I sistemi di comunicazione più complessi utilizzano una suite di protocolli per la gestione dello scambio dei dati. Si può pensare che la suite dei protocolli sia stratificata nel sistema di rete del computer dell'utente. Sebbene più protocolli possano coesistere all'interno di un unico strato di rete, tutti i protocolli che compongono la suite devono cooperare tra di essi. Una suite di protocolli può essere anche definita come "famiglia di protocolli". La suite dei protocolli TCP/IP ha il compito di gestire lo scambio dei dati su Internet e quindi fornire soluzioni a problemi che si verificano quando si presentano errori di comunicazione durante la trasmissione dei dati (ad esempio dovuti a guasti generici delle unità hardware o difficoltà connesse alla banda passante offerta dalla rete).

La gestione di apparati elettronici tramite web fornisce all'utente l'abilità di configurare e monitorare variegati dispositivi tramite Internet mediante l'uso di un comune browser. La soluzione migliore a questo tipo di esigenze è sicuramente data dall'utilizzo di server web embedded connesso ad un infrastruttura di rete al fine di fornire un interfaccia utente basata su web costruita mediante l'utilizzo dell'ormai noto linguaggio HTML [1] unitamente a grafici e ad altre caratteristiche comuni ai web browsers [2]. Se si pensa di aggiungere alle funzionalità ormai consolidate di un web server embedded la capacità di poter gestire applicazioni Java ecco che questi sistemi aprono le frontiere a capacità inesplorate, che rendono essi capaci di eseguire i più variegati compiti quali, ad esempio, quelli di controllo remoto, supervisione e gestione di sistemi elettronici **(fig. 1)**.



Architettura di un web server embedded **Figura 1**

Nelle applicazioni di controllo remoto si fa sempre più presente l'esigenza di interconnettere apparecchiature e strumentazioni tramite web server embedded al fine di avere una gestione quanto più decentralizzata possibile delle loro funzionalità [2]. Ognuno di questi controller programmato opportunamente diviene capace di eseguire differenti algoritmi di controllo.

APPROCCIO MEDIANTE L'UTILIZZO DELLA TECNOLOGIA JAVA

Il concetto della Virtual Machine di Java è particolarmente indicato per questo approccio permettendo l'uso di una strategia di controllo indipendente dalla piattaforma hardware del sistema in cui viene gestita. Questa metodologia è stata da tempo adoperata nelle applicazioni Internet dove non sono richiesti stringenti vincoli di real-time. L'uso del linguaggio di programmazione Java per le applicazioni di controllo remoto fornisce il vantaggio di integrare sistemi general purpose con internet permettendo la supervisione ed il controllo di sistemi. Oggigiorno i sistemi che si basano su web server embedded richiedono sempre applicazioni più complesse, facili da programmare, al fine di eseguire i compiti di supervisione e gestione. Per realizzare il concetto di network computing viene presentata la tecnologia Java al fine di ottenere la combinazione sinergica di sistemi di controllo realtime distribuiti che siano gestibili tramite la rete Internet.

Con l'incessante sviluppo della microelettronica i sistemi embedded sono stati applicati a molteplici prodotti industriali ed elettronici,

poiché presentano le caratteristiche di essere economici, affidabili con buone performance se comparati con il software utilizzato nei Personal Computers [3].
Il vantaggio delle tecnologie Internet permette di interconnettere tra loro dispositivi e sistemi all'interno della rete internet. Tutto questo facilita l'accesso ai dispositivi permettendo di effettuare operazioni di monitoraggio, di controllo, di reporting, start up, shutdown di qualsiasi dispositivo semplicemente premendo dei tasti all'interno di un interfaccia GUI gestita da un comune browser [4].
Il nuovo concetto che intendiamo introdurre si basa sull'esecuzione di Applet Java per eseguire operazioni di controllo o di monitoraggio di dispositivi remoti . In questo tipo di sistemi il controllo distribuito si ottiene mediante il trasferimento di pagine HTML e l'esecuzione di applet Java **(fig. 2)**.

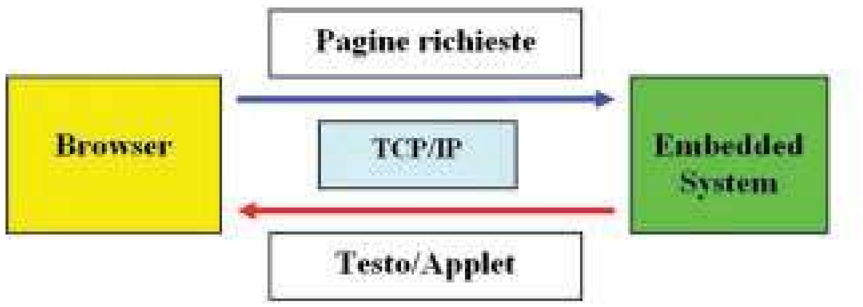


Figura 2 *Esecuzione di Applet Java per eseguire operazioni di controllo o di monitoraggio di dispositivi remoti*

Questo nuovo concetto permette di espandere le comuni capacità dei sistemi di controllo fornendo un sistema remoto distribuito per il controllo di sistemi elettronici.
La progettazione di sistemi embedded richiede l'integrazione e lo sviluppo di componenti hardware e software: spesse volte queste sono particolarmente difficili da realizzare poiché ogni controller possiede la sua piattaforma hardware e software. Anziché adoperare linguaggi differenti e non standard per l'implementazione del software nella maggior parte dei casi è preferibile adoperare un linguaggio comune [5]. Il Java rappresenta una scelta ottimale per differenti motivi: è un linguaggio standard completo di librerie, è un linguaggio molto semplice che riduce le problematiche inerenti l'analisi dei programmi, la loro ottimizzazione e trasformazione [5],[6].

I VANTAGGI DELL'UTILIZZO DI JAVA

Il linguaggio di programmazione Java si sta diffondendo sempre più all'interno dell'industria dell'information technology particolarmente per le applicazioni che prevedono l'utilizzo di database. Il Java è un linguaggio di programmazione che permette di installare un'applicazione all'interno di un server ed essere quindi eseguita su diverse piattaforme hardware. Questi vantaggi possono essere brevemente riassunti nei seguenti punti:

- **Indipendenza dalla piattaforma:** diversamente dai comuni compilatori che producono codice per CPU specifiche, il Java produce del codice per una CPU virtuale. Al fine di rimanere indipendente da specifiche piattaforme hardware il sistema runtime di Java fornisce un'interfaccia universale per qualsiasi applicazione che si desidera sviluppare. Tale interfaccia denominata JVM (Java Virtual Machine) è una sorta di processore virtuale che si interpone tra il processore fisico del PC e l'applicazione scritta in Java. Tuttavia, l'indipendenza dalla piattaforma non è sufficiente per assicurare il successo di un linguaggio di programmazione. La JVM è da diverso tempo inclusa all'interno dei browser più popolari quali, ad esempio, Microsoft Explorer e Netscape. Alcuni sistemi operativi real-time includono al loro interno la tecnologie Java e tutto ciò permette di giungere alla seguente conclusione "la JVM è una risorsa universale" [5,7];

- **Potenza**: Il Java racchiude in se nuove caratteristiche che includono la gestione dei database, l'invocazione dei metodi remoti ed altre caratteristiche inerenti la gestione della sicurezza.
- **Networking**: Il Java nasce come linguaggio di programmazione distribuito, il che si traduce nel fatto che la sua progettazione includeva sin dall'inizio la gestione di particolari funzioni inerenti il networking. Il Java ha una libreria vastissima di routine per la gestione dei protocolli quali, ad esempio, il TCP/IP, l'HTTP, l'FTP. Le applicazioni Java possono avere accesso ad oggetti attraverso la rete Internet per mezzo di URL (Universal Resource Locator, più comunemente noto come indirizzo del sito web) in un modo molto simile all'accesso ad un comune file system locale. Unendo la tecnologia Java alle potenzialità dei sistemi basati su web server embedded si possono ottenere dei sistemi molto potenti per la gestione di applicazioni di controllo.
- **Efficienza**: Le moderne JVM superano la forte limitazione delle passate dove veniva evidenziata la problematiche dell'estrema lentezza di esecuzione dei programmi.

Attualmente grazie all'utilizzo della

tecnologia Just in Time (JIT) compiler le performance d'esecuzione delle applet sono state fortemente migliorate [7].

L'UTILIZZO DI JAVA ALL'INTERNO DI SISTEMI WEB SERVER EMBEDDED

L'uso di linguaggi object-oriented assieme alle loro tecniche di progettazione permettono di ottenere un codice che risulta essere facilmente riutilizzabile e mantenibile. Normalmente, un'applicazione che va trasferita ad un controller si compone di un file binario eseguibile che viene direttamente eseguito dalla CPU. Il vantaggio principale di un approccio di questo tipo si traduce in una maggiore velocità d'esecuzione.

Il Java permette di ottenere la funzionalità "compila una sola volta e utilizza più volte". E' virtualmente possibile utilizzare lo stesso codice compilato in piattaforme differenti e, quindi, eseguire il codice su differenti Sistemi Operativi per fare i test ed il debugging del software per poi trasferire il tutto all'interno di un dispositivo di controllo [3].

Il Java si presenta come un linguaggio di programmazione fortemente adottato per la programmazione applicazioni che fanno di internet un punto di forza.

I vantaggi principali inerenti l'utilizzo delle applet Java all'interno di un web server embedded possono essere riassunte nei seguenti punti:

- Non occorre sviluppare una Graphical User Interface (GUI) poiché i browser web di per se suppliscono a tale funzionalità;
- Le dimensioni del codice Java sono minori rispetto alle istruzioni di codice macchina rendendo particolarmente attrattivo l'utilizzo di tale tecnologia all'interno di web server embedded dove, sicuramente, la dimensioni della RAM messa a disposizione per le applicazioni è limitata;
- Essendo l'esecuzione delle applet in locale, e quindi non all'interno del sistema embedded, l'efficienza e la complessità d'esecuzione degli algoritmi da eseguire sono a carico del client e non del sistema embedded che, sicuramente, avrà risorse di calcolo molto più limitate rispetto a quelle di un comune PC;
- Molti costruttori di microprocessori per sistemi embedded hanno investito nell'implementazione della JVM all'interno dei loro dispositivi: di conseguenza gran parte del software è già disponibile;
- E' intuitivo prevedere che in un immediato futuro la maggior parte dei kernel dei microcontrollori includerà la JVM.

LA JAVA VIRTUAL MACHINE

Attualmente esistono diversi modi di implementare la JVM. La **fig. 3** mostra due possibili implementazioni.

Codice Java	Codice C/C++
JVM	Librerie Native
Piattaforma Hardware	

Codice Java	
Sistema Operativo Java	
JVM	Librerie Native
Piattaforma Hardware	

Possibili implementazioni della JVM **Figura 3**

Nella prima la JVM viene integrata all'interno di un ambiente di sviluppo software. L'altra incorpora un Sistema Operativo Java che viene particolarmente indicata per quelle applicazioni che prevedono l'utilizzo di un unico ambiente di sviluppo sia per i programmatori che per gli utilizzatori [3]. Una volta disponibile l'interfaccia JVM è possibile includerla all'interno di un sistema di sviluppo embedded integrandola con le librerie del codice nativo. Ciò può essere ottenuto utilizzando qualsiasi linguaggio di programmazione quale, ad esempio, il Java bitcode interpreter e, quindi, compilare il tutto in accordo ai vincoli hardware del sistema in cui la si vuole integrare.

L'utilizzo della JVM all'interno di un web server embedded presenta il vantaggio che coloro che svilupperanno il codice per la programmazione dell'applicazione all'interno del sistema embedded utilizzeranno istruzioni Java e non dovranno tener contro delle relazioni che intercorrono tra la JVM e la microprogrammazione del sistema embedded. In più si può fornire all'utente finale un'interfaccia tipo applet parametrica che richiede semplicemente il setup di alcuni parametri. In quest'ultimo caso l'utente finale non necessita di avere alcuna conoscenza riguardo il linguaggio di programmazione Java.

IMPLEMENTAZIONE DELLA JVM ALL'INTERNO DI UN WEB SERVER EMBEDDED

L'applicazione che viene presentata in questo articolo, pur essendo stata implementata all'interno di un architettura basata su processore UBICOM, può essere virtualmente implementata all'interno di qualsiasi microprocessore o microcontrollore che dir si voglia. La **fig. 4** mostra lo schema architetturale semplificato di un possibile scenario d'applicazione in cui sono richiesti dei sistemi per il controllo di 16 ingressi digitali e 16 uscite digitali. In particolare ci si riferisce al dispositivo RECS 101 prodotto da Intellisystem Technologies.

L'architettura presentata permette la simulazione e lo studio di procedure tipiche dei sistemi di controllo quali,

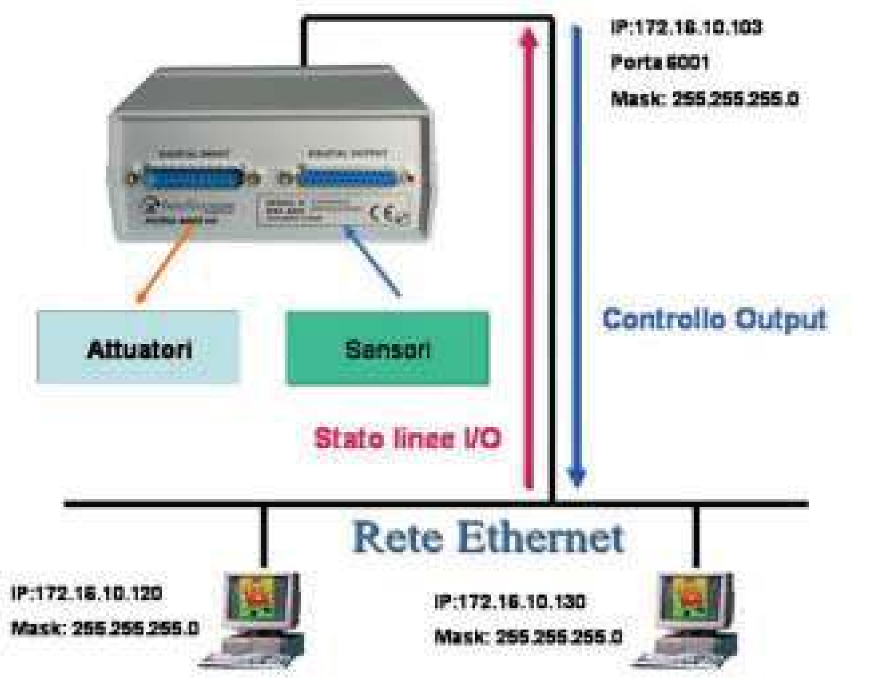


Figura 4 *Scenario d'applicazione del dispositivo RECS 101*

ad esempio: acquisizione di segnali, azioni di controllo per mezzo di attuatori, l'elaborazione e la presentazione delle informazioni acquisite o manipolate.
Il sistema embedded presentato si basa su un software di sviluppo scritto in C. Tale programma può far eseguire dei task preprogrammati all'interno della ROM o far eseguire delle applicazioni Java.
La capacità di far eseguire applicazioni Java viene fornita dai seguenti componenti software che devono essere prevaricati nella ROM del dispositivo mediante il su citato software di sviluppo [6]:
- Il file loader .class, che permette di fare il downloads del codice Java da eseguire nella RAM del dispositivo;
- L'implementazione della JVM stessa;
- Implementazioni di classi per la gestione dell'hardware locale;
- Classi Java riferite al sistema embedded.

La JVM è stata implementate in accordo alle specifiche dettate dallo standard [7], **fig. 5**.

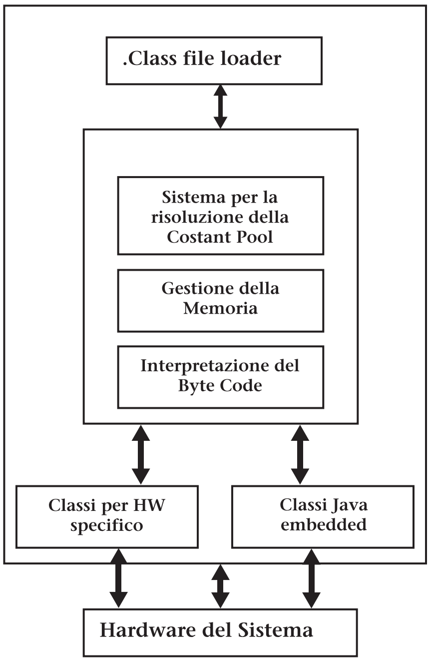


Figura 5 *Architettura della JVM implementata*

Il programma monitor che risiede all'interno del sistema embedded scarica il file .class che deve essere eseguito all'interno della memoria RAM; dopodiché l'informazione viene processata e si provvede alla costruzione della Constant Pool Table. La Constant Pool è quindi risolta ed il programma passa alla ricerca del metodo d'inizializzazione del programma main che dovrà essere eseguito. Dopo aver trovato questi metodi, la JVM ricerca il Byte Code che deve essere eseguito e, di conseguenza, invoca l'interprete di bytecode. Quando i metodi delle classi Java sono stati invocati la JVM richiama delle subroutine del firmware del microprocessore che provvederanno all'implementazione del metodo specifico.
La JVM di per se stessa non comunica direttamente con l'hardware del sistema ma usa delle classi per fare ciò. Un fattore molto limitante di questi sistemi è dovuto alla scarsità della memoria che si ha a disposizione. Di conseguenza ciò porta a fare delle scelte su quali metodi e classi devono essere implementati all'interno del sistema.

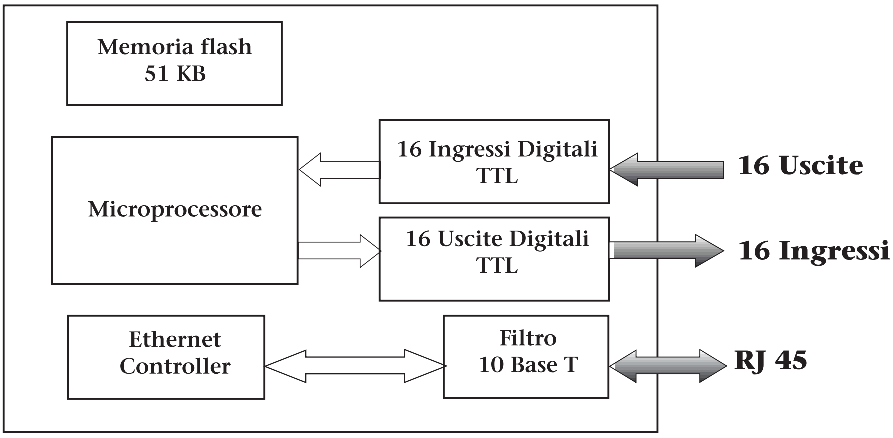
UN IMPLEMENTAZIONE PRATICA RECS 101

RECS 101 rappresenta una realizzazione pratica di quanto appena esposto [8]. La **tab. 1** riporta le principali caratteristiche e specifiche del sistema proposto.

RECS 101 integra al suo interno un network processor dotato di interfaccia di rete Ethernet per connettersi direttamente a qualsiasi rete locale sia essa Internet che Intranet. Ciò permette agli integratori di sistemi e alle aziende produttrici di connettere i loro dispositivi direttamente ad Internet attraverso una rete Lan e, di conseguenza, di gestire da remoto il controllo totale dei loro dispositivi attraverso interfacce grafiche utente personalizzabili, direttamente accessibili mediante i comuni browser quali, ad esempio, Microsoft Internet Explorer e Netscape Navigator.

Specifica	RECS 101
CPU	Ubicom SX52BD (8 bit microprocessor, 50 MIPS)
Memoria	512 Kb flash memory (Utilizzata per contenere le pagine web dell'utente
Connessione di Rete	Interfaccia Ethernet 10 Base-T (IEEE802-3)
Connessione Utente	16 Ingressi digitali / 16 Uscite digitali
Protocolli Internet Supportati	HTTP / BOOTP / TCP / UDP / IP ICMP / ARP Ethernet 802.3
Software di Utilità	**RECS Utility** (Piattaforma Windows) Web page uploader e cambio indirizzo IP

Specifiche del diapositivo RECS 101 **Tabella 1**



Schema funzionale di RECS 101 **Figura 6**

RECS 101 si basa sullo schema hardware presentato in **fig. 6**.
RECS 101 viene fornito con una pagina web precaricata all'interno della memoria flash del dispositivo che può essere modificata a piacimento in modo da personalizzarne le applicazioni.
RECS 101 contiene un web server integrato capace di gestire fino a 512k di documenti ed applicazioni web: tali risorse sono precaricate all'interno della memoria flash del dispositivo. La **fig. 7** è un esempio di una pagina web gestita da RECS 101 che può essere utilizzata per fornire informazioni statiche sul dispositivo quali, ad esempio, immagini, testi, files etc.
La pagina visualizzata può essere personalizzata a piacimento mediante l'uso dei più comuni editor di pagine HTML. Le pagine web possono contenere al loro interno file di immagini del tipo JPG, GIF, BMP, file video tipo SWF di Flash e qualsiasi altro file si ritenga opportuno che l'HTTP server di RECS 101 debba gestire. Selezionando il link "RECS 101

Home page personalizzabile del dispositivo RECS 101 **Figura 7**

Control Panel" si accederà alla pagina web dedicata al controllo dell'applicazione.
La caratteristica che rende unico tale dispositivo consiste nell'utilizzare un web server all'interno di un'applicazione embedded con la possibilità di eseguire del codice Java per la gestione dell'interfaccia relativa al controllo delle 16 porte di input e delle 16 porte di output (**fig. 8**). Tale caratteristica permette di poter gestire l'interfaccia utente tramite un'Applet Java parametrica: in questo modo l'utente finale può sviluppare la propria applicazione di controllo in modo molto veloce e sicuro senza dover essere in grado di programmare in Java.

Esempio di una possibile interfaccia GUI implementata in RECS 101 **Figura 8**

All'interno del pannello di controllo (**fig. 8**) si può notare un LED aggiuntivo specificato "Network". La sua funzionalità è quella di fornire all'utente lo stato della rete: una connessione senza problemi provoca il suo continuo lampeggiare. Nel caso di perdita momentanea del collegamento il LED non lampeggerà e, se la connessione non si ristabilisce entro qualche minuto il sistema chiuderà la connessione con RECS

101. Problematiche di questo tipo normalmente non sorgono in reti Intranet ma possono capitare se si collega RECS 101 alla rete Internet .

PERSONALIZZAZIONE DELL'INTERFACCIA UTENTE

RECS 101 è un dispositivo totalmente personalizzabile. Viene fornito con tutto il software necessario allo sviluppatore per poter sviluppare rapidissimamente la propria applicazione in maniera facile e sicura. Il software in dotazione contiene alcuni files ed un'APPLET (RECS.jar) di controllo che possono essere personalizzati mediante i parametri riportati di seguito:

- PDFOOK: Stringa d'inizializzazione Applet. Non è possibile effettuare nessuna modifica
- host: Indirizzo IP associato a RECS 101(Es. host value="172.16.10.103" vuol dire che l'indirizzo IP di RECS è 172.16.10.103
- port: Porta TCP adoperata dall' applicazione per comunicare con RECS 101. Il valore di tale porta è fisso e pertanto non modificabile (Es. port value=6001)
- polling: Intervallo di Polling. Ha una risoluzione di 10 ms e può essere settato in funzione dell'applicazione. Per es. "polling value=1" significa che il check dello stato d' I/O del dispositivo verrà controllato ogni 10 ms
- NumLed: Numero ingressi da monitorare mediante LED bicolore (Es. NumLed value=16, verranno visualizzati 16 LED indicatori di stato)
- NumB: Numero di pulsanti di comando per la modifica dello stato delle uscite (Es. NumB value=16, verranno visualizzati 16 pulsanti)

Per comodità del lettore la **Tab. 2** riassume tutti i parametri gestiti dall'applet in questione.

Di seguito si riporta il frammento del codice HTML del file index.html relativo alla personalizzazione dell'Applet in cui si evidenzia il setup dei parametri di inizializzazione.

Parametro	*Funzione*	*Esempio*	*Obbligatorio*	*Possibilità di modifica*
PDFOOK	Stringa inizzializzazione Applet		*SI*	*NO*
Host	Indirizzo IP di RECS	host value="172.16.10.103"	*SI*	*SI*
Port	Porta TCP per la comunicazione con RECS 101	port value=6001	*SI*	*NO*
Polling	Intervallo di polling	polling value=16	*SI*	*SI*
Num Led	Numero di ingressi da monitorare	NumLed value=16	*SI*	*SI*
NumB	Numero di pulsanti per il controllo delle uscite	NumB value=16	*SI*	*SI*

Tabella 2 *Parametri di configurazione dell'Applet*

L'esempio in questione prevede l'utilizzo di tutte le 16 uscite e di tutti i 16 ingressi messi a disposizione dall'hardware di RECS 101.

```
<APPLET CODE=Applicazione.class
ARCHIVE=RECS.jar WIDTH=850
HEIGHT=500>
<param name=PDFOOK
value="Intellisystem
Technologies Device">
<param name=host
value="172.16.10.103">
<param name=port value=6001>
<param name=polling value=1>
<param name=NumLed value=16>
<param name=NumBot value=16>
</APPLET>
```

La **fig. 8** rappresenta l'interfaccia utente che si ottiene applicando il codice appena esposto. Le limitazioni di quest'Applet consistono nel fatto che non è possibile modificare i testi ed i colori dei vari componenti che formano l'interfaccia utente.

Per gli utenti più esperti viene messa a disposizione un' Applet più elaborata che permette di personalizzare ulteriormente l'interfaccia grafica utente mediante altri parametri che permettono di definirne colori e testi (**fig. 9**).

Di seguito si riassumono i parametri che permettono la personalizzazione dell'Applet in questione (RECS.jar versione avanzata):

- PDFOOK : Stringa d'inizializzazione Applet. Non è possibile effettuare nessuna modifica

Figura 9 *Interfaccia GUI avanzata implementata in RECS 101*

- host: Indirizzo IP associato a RECS 101 (Es. host value= "172.16.10.103". Vuol dire che l'indirizzo IP di RECS è 172.16.10.103
- port: Porta TCP adoperata dall' applicazione per comunicare con RECS 101. Il valore di tale porta è fisso e pertanto non modificabile (Es. port value=6001)
- polling: Intervallo di Polling. Ha una risoluzione di 10 ms e può essere settato in funzione dell'applicazione. Per es. "polling value=1" significa che il controllo dello stato d' I/O del dispositivo verrà controllato ogni 10 ms
- Title: Stringa intestazione applicazione. (Es. Title value="RECS I/O DEMO ")
- ColTit: Colore da associare alla strin-

ga impostata nel parametro "Titolo". (Es. ColTit value="green" , il testo verrà stampato in verde)

- CAPL: Colore di sfondo Applet. (ES. CAPL value="yellow", lo sfondo sarà giallo)
- NumLed: Numero ingressi da monitorare mediante LED bicolore (Es. NumLed value=16, verranno visualizzati 16 LED indicatori di stato)
- NumB: Numero di pulsanti di comando per la modifica dello stato delle uscite (Es. NumB value=16, verranno visualizzati 16 pulsanti)
- TBT*: Testo da associare al pulsante * relativo all'uscita * (Es. TBT1 value="Comando 10" è il testo da associare al pulsante 10 per modificare lo stato dell' uscita 10)
- CTBT*: Colore del testo associato al titolo del pulsante *. (Es. CTBT10 value="red", il colore associato al testo relativo al pulsante 10 è rosso)
- CLBF*: Colore associato al LED di stato dell' uscita * quando quest'ultima è nello stato "OFF" (Es. CLBF10 value="gray", il colore del LED associato allo stato "OFF" dell' uscita 10 sarà grigio)
- CLBT*: Colore associato al LED di stato dell' uscita * quando quest'ultima è nello stato "ON" (Es. CLBT10 value= "blue", il colore del LED associato allo stato "ON" dell' uscita n.10 sarà blu)
- TLD*: Testo da associare al LED * relativo all' ingresso *. (Es. TLD1 value="Luce Camera" è il testo da associare al LED 1 per effettuare la lettura dello stato dell' ingresso 1)
- CTLD*: Colore del testo associato al titolo del LED * relativo all'ingresso *. (Es. CTLD1 value="black", il colore associato al testo relativo al LED 1 sarà nero)
- CLIF*: Colore associato al LED di stato dell' ingresso * quando quest'ultimo è nello stato "OFF" (Es. CLIF10 value="green", il colore del LED associato allo stato "OFF" dell' ingresso 10 sarà verde)
- CLIT*: Colore associato al LED di stato dell' ingresso * quando quest'ultimo è nello stato "ON" (Es. CLIT10 value="red", il colore del LED associato allo stato "ON" dell' ingresso 10 sarà rosso)

Per comodità del lettore la tab. 3 riassume in forma tabulare i parametri personalizzabili dell'Applet per la gestione avanzata di RECS 101.
Di seguito si riporta il frammento del codice HTML del file index.html relativo alla personalizzazione dell'Applet in cui si evidenzia il setup dei parametri di inizializzazione.

```
<APPLET CODE=Applicazione.class
ARCHIVE=RECS.jar WIDTH=850
HEIGHT=500>
<param name=PDFOOK
value="Intellisystem
Technologies Device">
<param name=host
value="172.16.10.103">
<param name=port value=6001>
<param name=polling value=1>
<param name=Title value="RECS
101 I/O Demo">
<param name=ColTit
value="black">
<param name=CAPL value="white">
<param name=NumLed value=16>
<param name=NumBot value=16>
```

Un esempio di personalizzazione dei pulsanti e degli indicatori LED è rappresentato dal seguente codice contenuto all'interno del file index.html:

```
<param name=TBT1 value="Comando
1">
<param name=CTBT1 value="red">
<param name=CLBF1 value="gray">
<param name=CLBT1 value="blue">
<param name=TLD1 value="Ingresso
1">
<param name=CTLD1 value="black">
<param name=CLIF1 value="green">
<param name=CLIT1 value="red">
```

Parametro	*Funzione*	*Esempio*	*Obbligatorio*	*Possibilità di modifica*
PDFOOK	Stringa inizzializzazione Applet		*SI*	*NO*
Host	Indirizzo IP di RECS	host value="172.16.10.103"	*SI*	*SI*
Port	Porta TCP per la comunicazione con RECS 101	port value=6001	*SI*	*NO*
Polling	Intervallo di polling	polling value=1	*SI*	*SI*
Title	*Intestazione Applet*	Title value="RECS I/O DEMO"	*NO*	*SI*
ColTit	*Colore da associare alla stringa Title*	Coltit value="green"	*NO*	*SI*
CAPL	*Colore background Applet*	CAPL value="yellow"	*NO*	*SI*
Num Led	Numero di ingressi da monitorare	NumLed value=16	*SI*	*SI*
NumB	Numero di pulsanti per il controllo delle uscite	NumB value=16	*SI*	*SI*
*TBT**	*Testo da associare al pulsante**	TBT1 value="Comando 10"	*NO*	*SI*
*CTBT**	*Colore del testo associato al titole del pulsante**	CTBT10 value="red"	*NO*	*SI*
*CLBF**	*Colore LED di stato dell'uscita* quando questa si trovi nello stato "OFF"*	CLBF10 value="gray"	*NO*	*SI*
*CLBT**	*Colore LED di stato dell'uscita* quando questa si trovi nello stato "ON"*	CLBT10 value="blue"	*NO*	*SI*
*TLD**	*Testo da associare al LED* relativo all'ingresso*	TLD 1 value="Luce Camera"	*NO*	*SI*
*CTLD**	*Colore del testo associato al titolo del LED* relativo all'ingresso**	CTLD1 value="black"	*NO*	*SI*
*CLIF**	*Colore associato al LED* di stato dell'ingresso* quando quest'ultimo è nello stato "OFF"*	CLIF10 value="green"	*NO*	*SI*
*CLIT**	*Colore associato al LED* di stato dell'ingresso* quando quest'ultimo è nello stato "ON"*	CLIT10 value="red"	*NO*	*SI*

Parametri di configurazione dell'Applet per la gestione avanzata di RECS 101 **Tabella 3**

Poiché non occorre RECS 101 per simularne il suo funzionamento, collegandosi al seguente indirizzo http://www.intellisystem.it/recs/Interfaccia.htm si possono provare le due interfacce proposte.
Per chi invece volesse dilettarsi a sperimentare la personalizzazione delle interfacce, Intellisystem Technologies mette a disposizione nel proprio sito tutto il software necessario.
Per fare ciò occorre:

• scaricare una delle versioni delle interfacce proposte dal seguente indirizzo http://www.intellisystem.it/recs/download.htm;
• decompattare i file in una cartella;
• modificare i parametri dell'interfaccia agendo sul file index.html utilizzando un qualsiasi editor web;
• richiamare la pagina 101.html mediante un qualsiasi Web Browser.

I files necessari per la personalizzazione dell'interfaccia utente di RECS 101 sono essenzialmente tre (a parte tutti quelli necessari per il contenuto grafico delle pagine web): 101.html, index.html, RECS.jar.

La fig. 10 riassume quanto detto in precedenza, ovvero:

1. La pagina 101.html rappresenta la home page del sito web contenuto in RECS 101. Al suo interno è presente un collegamento alla pagina Index.html;

2. La pagina Index.html contiene al suo interno i parametri di setup dell'Applet per la gestione degli ingressi e delle uscite di RECS 101;

3. Tramite l'applet RECS.jar si interviene sulle porte d'input e di output per la gestione dell'hardware che si intende controllare.

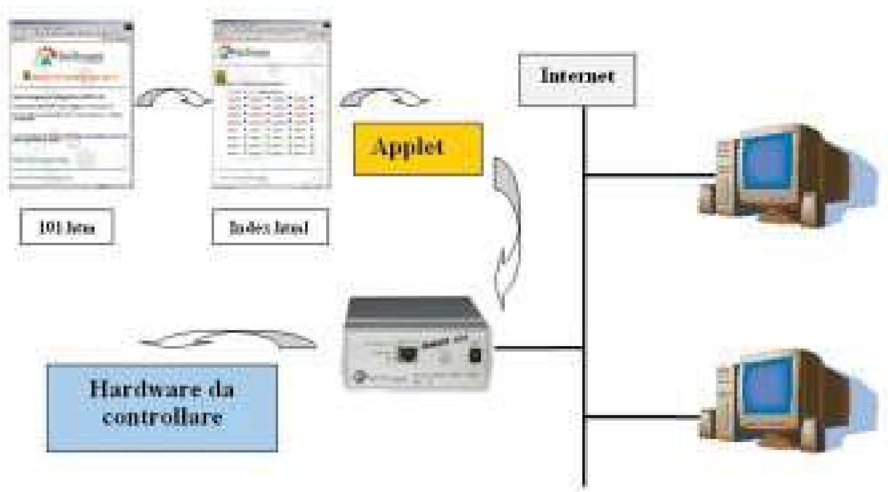


Tabella 2 *Files necessari per la personalizzazione dell'interfaccia utente di RECS 101*

NEL PROSSIMO NUMERO SI DISCUTERÀ DEI SEGUENTI ARGOMENTI RIGUARDANTI RECS 101:

1) Configurazione dei parametri di rete

2) Upload dell'interfaccia utente personalizzata

3) Implementazione delle interfacce hardware sulle porte di Input/Output

BIBLIOGRAFIA

[1] McCombie, B.,"Embedded Web server now and in the future," Real-Time Magazine, no.1 March 1998, pp. 82-83.

[2] Wilson, A.,"The Challenge of embedded Internet", Electronic Product Design, January 1998, pp. 31-2,34.

[3] D. Mulchandani, "Java for Embedded Systems", in IEEE Computer Magazine, pp. 30-39, May June 1998.

[4] Aptronix, "Bring Embedded System to the Internet", http://www.aptronix.com.

[5] J. Gosling, B. Joy, G. Steele,"The Java Laguage Specification", http://java.sun.com

[6] J.S. Young et All., "Design and specification of embedded system in java using Successive, formal Refinement", Proceedings of DAC'98, 1998 Design Automation Conference. San Francisco, C.A.,june 15-19.

[7] T. Lindholm, F. Yellin "The Java Virtual Machine Specification", 1996. http://java.sun.com

[8] Intellisystem Technologies. http://www.intellisystem.it

REALIZZAZIONI PRATICHE • TUTORIALS • RADIANTISTICA • COMPUTER HARDWARE • ROBOTICA

Fare ELETTRONICA

N° 213/214 - MARZO/APRILE 2003 - ANNO 19

€ 6,00 - Frs 12,00

NUMERO DOPPIO

ALL'INTERNO LE PAGINE DI:

MHz FareELETTRONICARADIO

NUOVO INSERTO

ELETTRONICA GENERALE

- 100 LUCI A SCORRIMENTO CON 2 INTEGRATI
- MAGNETOTERAPIA BFC
- PIC-PONG
- LIGHT INTERFACE

TUTORIAL

- LE INTERFACCE SERIALI RS-422 E RS-485

HARDWARE

- GUIDA ALL'USO DEI DISPLAY LCD INTELLIGENTI
- L'INTERFACCIA MIDI E IL COMPOSITORE DELL'ERA DIGITALE

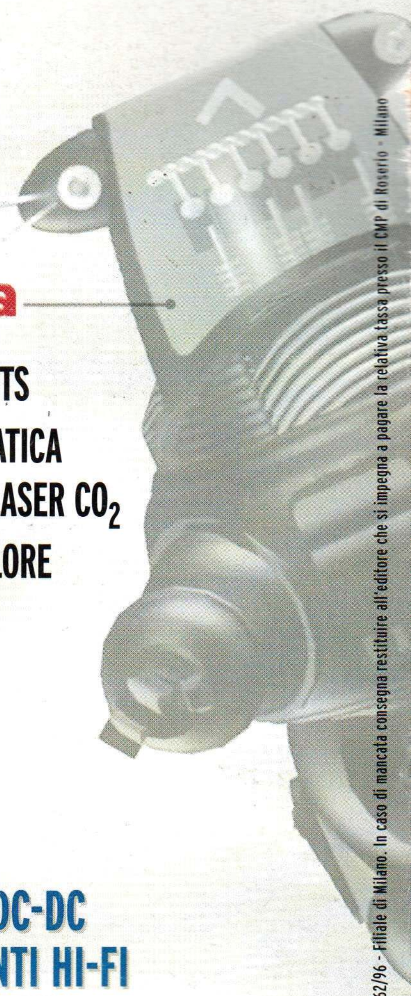
AUTOMAZIONE

- PLC 51

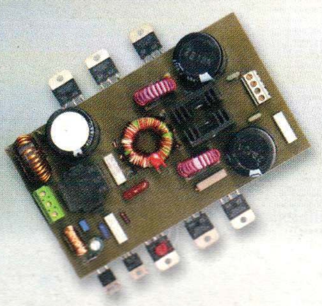
MHz

- RICEVITORE 27 MHz 6 CANALI + VFO

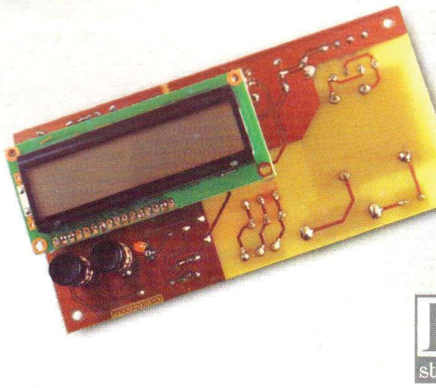
ROBOMANIA



- INTRODUZIONE AI BEAM ROBOTS
- INTRODUZIONE TEORICA E PRATICA ALLA TECNOLOGIA DI TAGLIO LASER CO_2
- BASICX BX24 MICROCONTROLLORE PROGRAMMABILE IN BASIC
- SFR04 MODULO SONAR AD ULTRASUONI



INVERTER DC-DC PER IMPIANTI HI-FI IN AUTO



CRONOTERMOSTATO PER RISCALDAMENTO

Spedizione in A.P. - 45% - ART. 2 COMMA 20/B LEGGE 662/96 - Filiale di Milano. In caso di mancata consegna restituire all'editore che si impegna a pagare la relativa tassa presso il CMP di Roserio - Milano

9 771591 227008 30213

DTP studio editrice

RECS 101: UN WEB SERVER EMBEDDED PER APPLICAZIONI DI CONTROLLO REMOTO TRAMITE TCP/IP

seconda parte

di Cristian Randieri
randieri@intellisystem.it

In questa seconda parte della presentazione del dispositivo RECS 101 sono affrontati i seguenti argomenti[1]: le problematiche inerenti la configurazione dei parametri di rete per il corretto utilizzo del dispositivo all'interno di una rete Intranet/Internet; Come effettuare le operazioni di upload dell'interfaccia GUI personalizzata all'interno della memoria flash del dispositivo; In conclusione vengono presentati due possibili circuiti d'interfaccia che permettono allo sviluppatore di implementare in modo sicuro un proprio hardware per interfacciare RECS 101 a qualsiasi dispositivo/impianto da controllare via Internet.

CONFIGURAZIONE DEI PARAMETRI DI RETE

Prima di poter utilizzare RECS 101 occorre configurare correttamente i suoi parametri di rete utilizzando il programma shareware di utilità RECS Utility, scaricabile al seguente indirizzo http://www.intellisystem.it/recs/download.htm. RECS Utility è stato progettato e realizzato per lavorare su piattaforma Microsoft Windows sui sistemi operativi delle versioni 95/98/ME/NT/2000 e XP Home/Professional.
Una volta installato ed eseguito RECS Utility, verrà visualizzata la maschera iniziale del programma che permette di impostare l'indirizzo IP di RECS 101 (**fig. 1**) Prima di configurare l'indirizzo IP da assegnare al dispositivo è necessario avere informazioni sulla struttura degli indirizzi IP della rete in cui si andrà ad installare RECS 101. Pertanto occorrerà conoscere:

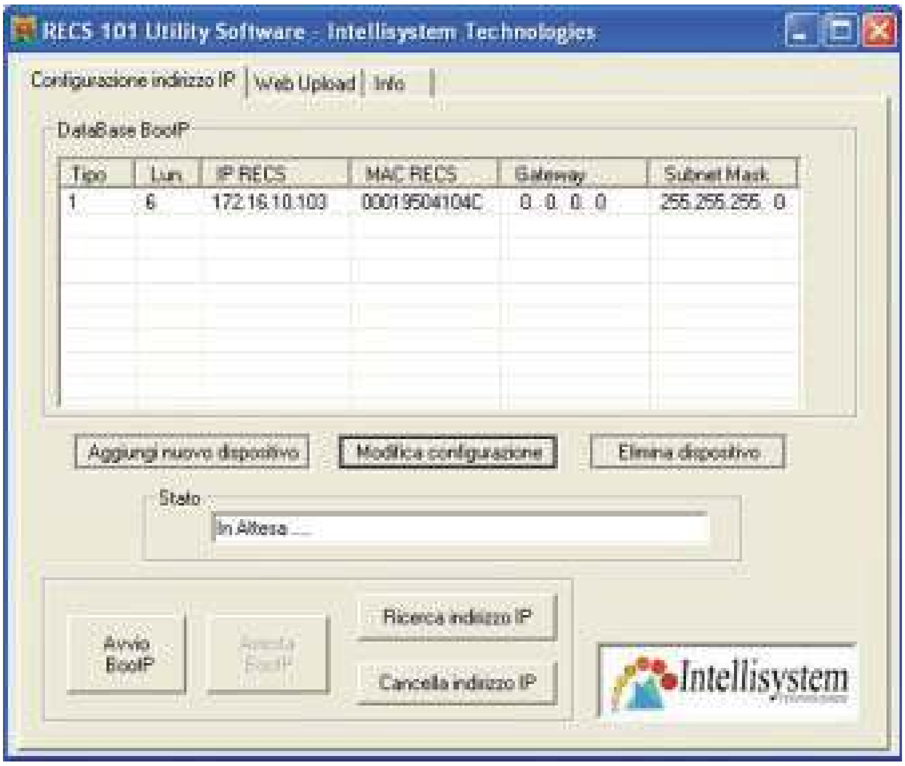
- Un indirizzo IP non utilizzato all'interno della struttura;
- La Subnet Mask della struttura.

RECS 101 è preconfigurato in fase di testing con le seguenti impostazioni di fabbrica:

Indirizzo IP: 172.16.10.103
Subnet Mask: 255.255.255.0

Per impostare o cambiare l'indirizzo IP di RECS 101 occorre prima reinizializzare l'indirizzo IP al valore di default 0.0.0.0 (operazione che verrà descritta di seguito). Inizializzare un nuovo indirizzo IP per RECS 101 implica la rimozione dell'associazione IP/MAC memorizzata all'interno del PC che s'intende adoperare. Tale informazione è memorizzata nella cache del protocollo ARP. Tramite il comando mostrato di seguito è possibile visualizzare tutte le associazioni IP/MAC presenti nel PC (ovviamente gli indirizzi IP riportati nell'esempio sono puramente fittizi):

```
>> arp -a
Interface: 192.168.1.100 on Interface 2
Internet Address    Physical Address    Type
192.168.1.15        00-01-95-04-02-03   dynamic
192.168.1.23        00-a0-95-14-12-23   dynamic
```



Schermata iniziale di RECS Utitlity. **Figura 1**

Finestra per la cancellazione dell'indirizzo IP. **Figura 2**

La rimozione di tale associazione va fatta utilizzando il comando ARP nel modo seguente;

```
>> arp -d 192.168.1.15
```

Adesso si è pronti a reinizializzare l'indirizzo IP all'interno del dispositivo RECS 101. Selezionando la finestra "Configurazione indirizzo IP" e premendo il pulsante "Cancella indirizzo IP" sarà visualizzata la finestra riportata in **fig. 2**. Dopo aver correttamente impostato l'indirizzo IP che si vuole cancellare, automaticamente alla pressione del pulsante "OK" l'indirizzo IP di RECS 101 sarà impostato al suo valore di default 0.0.0.0. Nel caso in cui non si conosce a priori il vecchio indirizzo IP del dispositivo, si può adoperare la funzione di ricerca indirizzo IP tramite la pressione del pulsante "Ricerca indirizzo IP". Sarà visualizzata la finestra riportata in **fig. 3**. Inserendo l'indirizzo MAC riportato nel pannello posteriore del dispositivo (**fig.4**) e premendo il pulsante "Trova" dopo qualche istante il software restituirà l'indirizzo IP correntemente impostato nel dispositivo RECS 101. Occorre sottolineare che la ricerca non avrà esito positivo se la Subnet Mask del PC adoperato per la configurazione è diversa da quella impostata all'interno del dispositivo RECS 101 che per default è 255.255.255.0. Verificare anche l'indirizzo IP del PC da cui accedete a RECS 101 facendo in modo che l'indirizzo del PC ricada all'interno della stessa Subnet Mask di RECS 101.Ad esempio, supponendo che RECS 101 sia configurato nel modo seguente:

Indirizzo IP: 172.16.10.103
Subnet Mask: 255.255.255.0

il PC con il quale si vuole accedere a RECS 101 dovrà avere la seguente configurazione:

Indirizzo IP: 172.16.10.xxx
Subnet Mask: 255.255.255.0

Dove "xxx" sta per qualsiasi numero compreso tra 0 e 255. Per modificare la configurazione dell'indirizzo IP e della Subnet Mask del PC occorre adoperare l'esatta funzionalità di Windows (**fig. 5**). Una volta rimosso con successo il vecchio indirizzo IP precedentemente assegnato a RECS 101 si è pronti per inserire in nuovo indirizzo IP selezionando la finestra "Configurazione indirizzo IP" (**fig. 1**). Se nella tabella "DataBase BootP" appare la vecchia configurazione del dispositivo, occorre premere il pulsante "Modifica configurazione" altrimenti premere il pulsante "Aggiungi nuovo dispositivo": si aprirà la finestra riportata in **fig. 6**. Tale finestra presenterà i valori in precedenza impostati nel caso in cui il dispositivo sia già presente nel DataBase BootP. Viceversa conterrà i campi vuoti. La **fig. 6** riporta l'esempio di quest'ultimo caso. Modificare il campo "Indirizzo IP" col nuovo indirizzo IP, il campo "Indirizzo MAC" (l'indirizzo MAC del vostro dispositivo è situato nel pannello posteriore, **fig. 4**), il campo "Gateway" (se esiste un Gateway nella vostra rete) ed in fine il campo "Subnet Mask" (**fig. 6**). Premendo il tasto "Aggiungi" le informazioni editate saranno memorizzate nel database degli indirizzi IP associato a RECS Utility.
Per completare la configurazione dell'indirizzo IP memorizzato occorre lanciare le funzionalità del server BootP premendo il pulsante "Avvio BootP" della finestra "Configurazione indirizzo IP" (**fig. 1**). Attendere qualche istante sino a quando lo stato del dispositivo evidenzia la scritta "Invio della replica BootP all'indirizzo IP

"[xxx.xxx.xxx.xxx]"(dove "xxx.xxx.xxx.xxx" sarà il nuovo indirizzo IP impostato). Attendere sino a quando il led Tx non finisca di lampeggiare, quindi premere il pulsante "Arresto BootP" (**fig. 1**). A questo punto l'indirizzo IP di RECS 101 è stato cambiato. Se l'inizializzazione dell'indirizzo IP è avvenuta con successo, il Led TX del dispositivo lampeggerà in continuazione. Per verificarne il corretto funzionamento utilizzare il comando Ping da DOS. Ad esempio, supponendo che il nuovo indirizzo sia 172.16.10.105, se il comando Ping risponde come di seguito riportato, allora la modifica dell'indirizzo IP ha avuto successo. In caso contrario ripetere tutto il procedimento descritto.

```
>> Ping 172.16.10.105
>> Pinging 172.16.10.105 with 32 bytes of data:

Reply from 172.16.10.105: bytes=32 time=10ms TTL=251
Reply from 172.16.10.105: bytes=32 time<10ms TTL=251
Reply from 172.16.10.105: bytes=32 time=10ms TTL=251
```

Potrebbe capitare il caso che il dispositivo RECS 101 sia settato con un indirizzo IP non compatibile con la rete nella quale RECS 101 è stato installato. Ciò si traduce nel fatto che RECS 101 non può essere indirizzato e di conseguenza non è possibile cambiare il suo indirizzo IP. In questo caso l'unica soluzione praticabile è quella di sconnettere RECS 101 dalla rete Lan nella quale era installato e connetterlo direttamente ad un PC dotato d'interfaccia Ethernet mediante un cavo di rete incrociato.

Le operazioni da compiere sono riassunte nei seguenti punti:

1. Scollegare RECS 101 dalla rete Lan.
2. Collegare RECS 101 ad un PC tramite un cavo di rete incrociato.

Le **figure 7** e **8** mostrano la differenza di connessione tra un cavo di rete dritto ed uno incrociato.
RECS 101 può essere collegato direttamente ad internet e quindi rendere le sue applicazioni visibili da tutte le parti del mondo se è configurato con un indirizzo IP statico.
La **fig. 9** ne rappresenta una possibile connessione.

In sintesi ciò che occorre è:

1. Un indirizzo IP statico, ovvero un indirizzo IP che abbia visibilità su Internet.
2. Una connessione diretta ad internet ad esempio ADSL o una rete LAN perennemente connessa.
3. Condividere la connessione ad internet tramite un Router e/o un Hub/Switch.
4. Collegare RECS 101 alla rete dopo averlo configurato con il relativo indirizzo IP statico.

UPLOAD DELL'INTERFACCIA UTENTE PERSONALIZZATA

Per sfruttare al massimo le potenzialità di RECS 101, occorre personalizzare l'interfaccia grafica del dispositivo agendo e/o modificando i files forniti dal costruttore. Definita l'interfaccia utente per l'applicazione che s'intende progettare non resta che fare l'upload all'interno della memoria flash di RECS 101. Si ricorda che la memoria totale a disposizione dell'utente è di 500 KByte, con supporto fino a 256 differenti file. Poiché RECS 101 utilizza un file system proprietario, i file relativi all'interfaccia web sono gestiti mediante una tabella interna di tipo

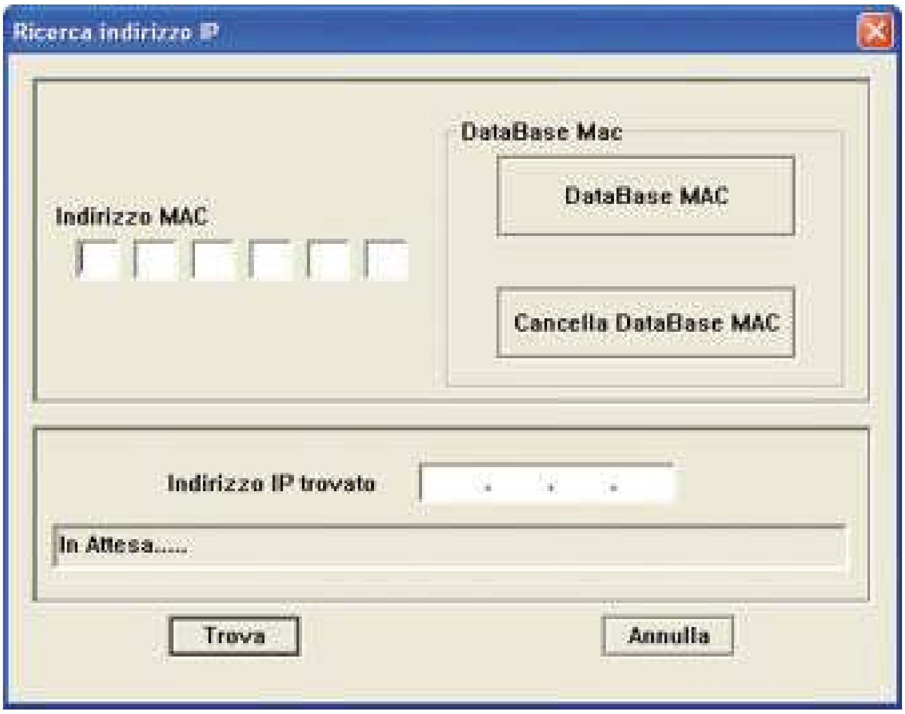
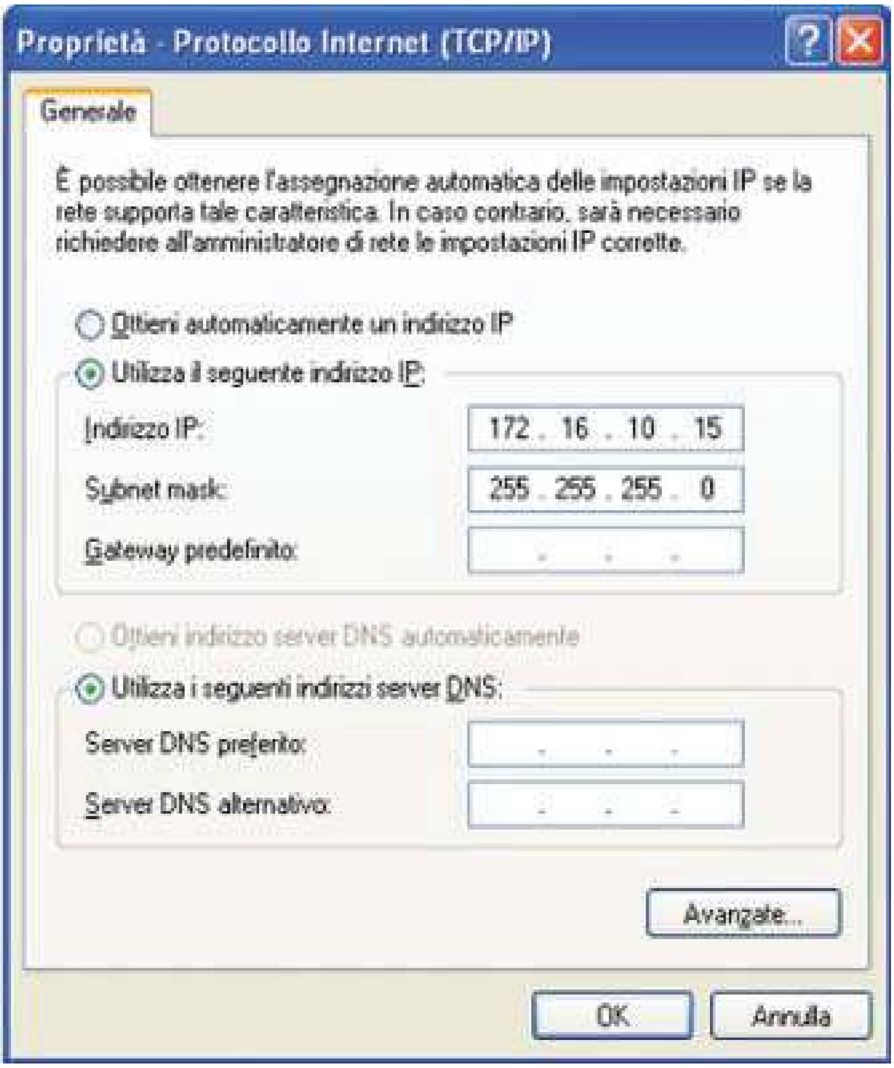


Figura 3 *Finestra per la ricerca dell'indirizzo IP impostato nel dispositivo.*

Figura 4 *Indirizzo MAC.*



Maschera di windows per il setup della connessione di rete. **Figura 5**

Finestra per la configurazione del nuovo indirizzo IP. **Figura 6**

"file index". Per trasferire i files all'interno di RECS 101 è necessario procedere prima alla creazione di un file di progetto che rappresenta l'immagine dei files che dovranno essere memorizzati all'interno della memoria flash. Il file di progetto, che presenta un'estensione *.REC, può essere unicamente gestito dal web server integrato in RECS 101. RECS Utility contiene al suo interno delle funzionalità dedicate alla costruzione e all'upload di questo tipo di file. Per procedere all'upload dell'interfaccia utente personalizzata occorre seguire i seguenti passi:

1) Creare e/o modificare le pagine web personalizzate con qualsiasi software di web-publishing.
2) Impostare i parametri dell'applet in funzione delle esigenze di progetto.
3) Utilizzare il software RECS Utility per creare il file di progetto *.REC.
4) Fare l'upload del file di progetto all'interno di RECS 101.

Di seguito è presentato un esempio illustrativo atto a descrivere l'operazione di upload dell'interfaccia personalizzata presente nel CD-Rom fornito in dotazione situata all'interno della cartella "CP" (il lettore può scaricare tale software direttamente da questo indirizzo:http://www.intellisystem.it/recs/downloads/CP.zip). Poiché l'upload del file di progetto avviene mediante il protocollo UDP (protocollo che non prevede la conferma della ricezione dei pacchetti inviati) il lettore, in realtà, pur non avendo a disposizione RECS 101 può emulare tale funzionamento anche se il dispositivo non è fisicamente connesso alla rete.

Le operazioni da compiere per procedere all'operazione di upload sono le seguenti:

1) Dopo aver lanciato RECS Utility selezionare l'opzione "Web Upload" come riportato in **fig. 10**.
2) Premere il pulsante "Seleziona Files

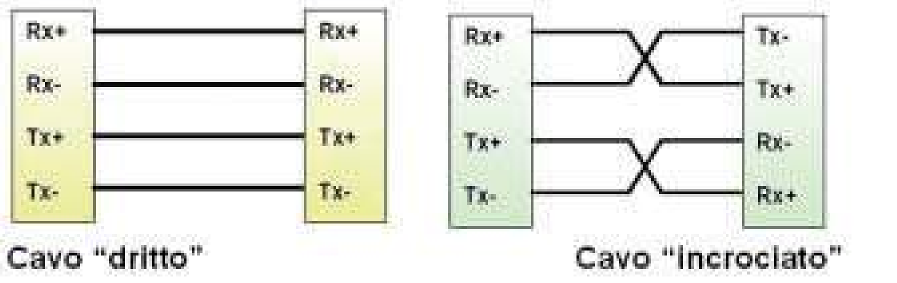


Figura 7 *Tipologie di cavi di rete cavo "dritto" e cavo "incrociato".*

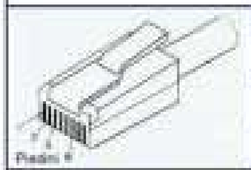
Conduttori da incrociare:				
	Nome	Pin del connettore 1	Pin del connettore 2	Nome
	TX+	1	3	RX+
	TX-	2	6	RX-
	RX+	3	1	TX+
	RX-	6	2	TX-
I rimanenti conduttori (il 4, 5, 7 e 8) devono rimanere invariati.				

Figura 8 *Realizzazione di un cavo incrociato.*

di progetto" e selezionare la cartella contenente i files (Ad esempio la cartella Control Panel "CP" contenuta all'interno del CD-Rom fornito in dotazione), premere "ok" per proseguire (**fig. 11**).

3) Inserire quindi il nome da assegnare al file di progetto e premere il pulsante "Salva" (**fig. 12**).

4) Premere il pulsante "Upload" per trasferire il file immagine all'interno di RECS 101. Questa procedura attiverà una barra di progressione che indica lo stato d'avanzamento dell'operazione di upload in corso. Al termine di tale fase sarà visualizzato un messaggio che comunica la chiusura dell'operazione.

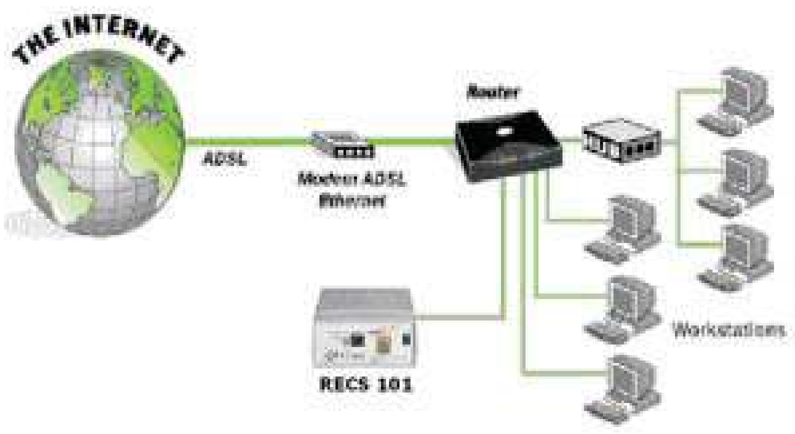


Figura 9 *Esempio di una possibile connessione di RECS 101 ad Internet tramite una linea ADSL.*

Nel caso si volessero programmare più RECS 101 con la stessa interfaccia utente e quindi col medesimo indirizzo IP si possono saltare le fasi descritte nei punti 1-4 a patto di possedere il file di progetto *.REC. In questo caso premere il pulsante "Seleziona File di progetto" (**fig. 10**), selezionare il file di progetto da trasferire in RECS 101 (**fig. 13**), quindi procedere con l'operazione di upload premendo il pulsante "Upload" (**fig. 10**).

IMPLEMENTAZIONE DELLE INTERFACCE HARDWARE SULLE PORTE DI INPUT/OUTPUT

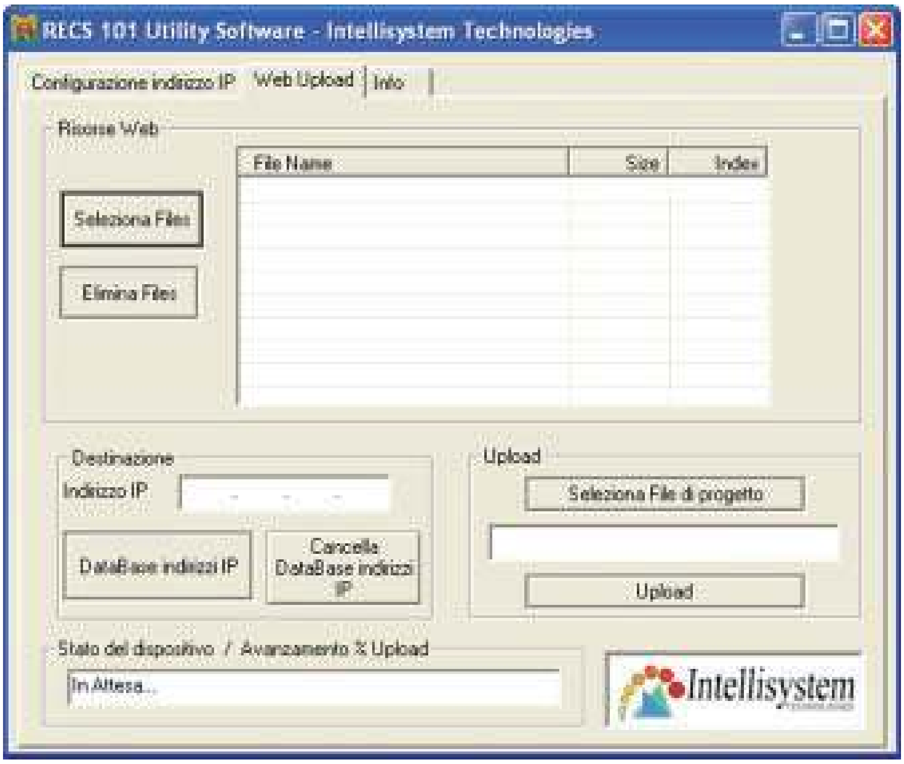
RECS 101 si interfaccia con l'impianto o dispositivo da controllare mediante due porte a 16 bit digitali, rispettivamente, una di Input ed un'altra di Output poste sul frontalino posteriore. La **fig. 14** riporta la piedinatura dei connettori Cannon a 25 poli che ospitano tali porte.

Il progettista che intende interfacciare RECS 101 deve predisporre delle interfacce che consentono il corretto rispetto delle caratteristiche elettroniche della logica TTL implementata nelle due porte. Di seguito distingueremo due tipi d'interfacce rispettivamente una per la porta di Input ed un'altra per la porta di Output.

UNITÀ D'INPUT

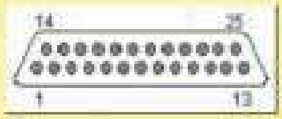
Poiché l'interfaccia di I/O di RECS 101 lavora con livelli logici TTL il dispositivo da interfacciare alla porta d'ingresso deve presentare anch'esso un interfaccia di tipo TTL. I 16 bit d'ingresso per l'applicazione fornita sono stati progettati per funzionare in logica TTL "Low Active".

Non sempre però i dispositivi hanno delle porte TTL e perciò, in questo caso, è opportuno adoperare un circuito che interponendosi tra RECS 101 e il dispositivo da interfacciare possa connettere i due dispositivi senza che essi corrano il rischio di danneggiarsi. Il circuito suggerito utilizza dei fotoaccopiatori che, garantendo



Maschera per la gestione dell'upload di RECS Utility. **Figura 10**

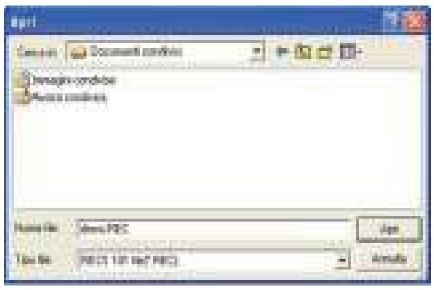
Digital Input			Digital Output		
LED	Pin	Note	Tasto	Pin	Note
1	2		1	2	
2	15		2	15	
3	3		3	3	
4	16		4	16	
5	4		5	4	
6	17		6	17	
7	5		7	5	
8	18		8	18	
9	6		9	6	
10	19		10	19	
11	7		11	7	
12	20		12	20	
13	8		13	8	
14	21		14	21	
15	9		15	9	
16	22		16	22	
	1	Vcc +5v		1	Vcc +5v
	14	Vcc +5v		14	Vcc +5v
	10	GND		10	GND
	23	GND		23	GND
	12, 13, 24, 25	Non usati		12, 13, 24, 25	Non usati



Piedinatura dei connettori di I/O di RECS 101. **Figura 14**

Selezione dei file di progetto. **Figura 11**

Creazione del file di progetto. **Figura 12**



Selezione del file di progetto. **Figura 13**

un totale isolamento tra i due dispositivi, ne assicurano il corretto funzionamento. La **fig. 15** mostra una possibile realizzazione del circuito proposto.

UNITÀ D'OUTPUT

RECS 101 è dotato 16 uscite che lavorano con livelli logici TTL progettati per funzionare in logica "High Active". Affinché RECS 101 possa essere correttamente interfacciato con un altro dispositivo che lavora con

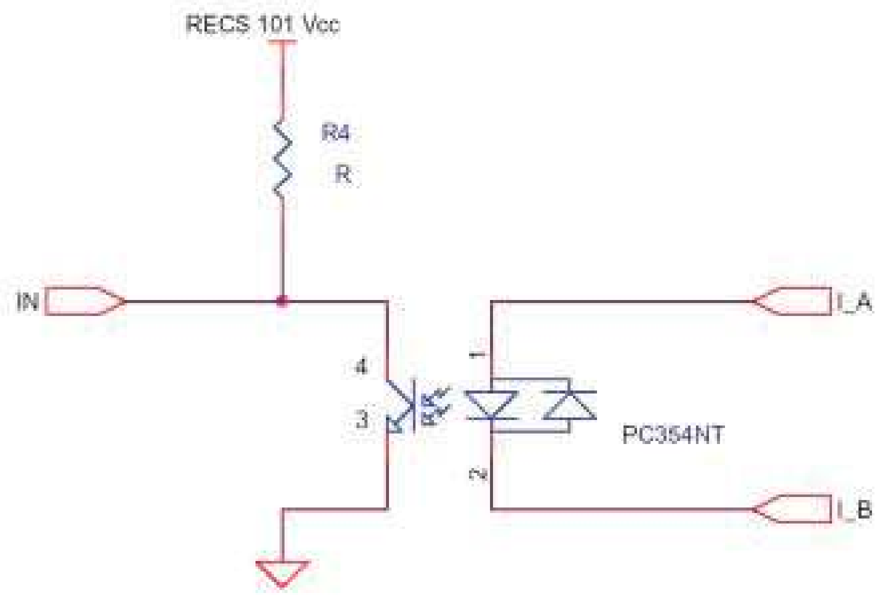


Figura 15 *Interfaccia per la connessione di un dispositivo alla porta d'ingresso di RECS 101.*

tensioni diverse si consiglia l'uso di fotaccopiatori che garantendo un totale isolamento tra i due dispositivi ne assicurano il corretto funzionamento. La **fig. 16** mostra lo schema elettrico di un circuito d'esempio per la realizzazione di un'interfaccia d'uscita da collegare a RECS 101. Tale circuito si presta benissimo per tutte quelle applicazioni nelle quali è necessario effettuare un controllo di tipo ON/OFF di carichi di qualunque tipo. Poiché il circuito contiene dei relay assieme agli optoisolatori si ottiene un circuito doppiamente isolato sia galvanicamente (per mezzo dei realy) che otticamente (mediante l'uso di fotoaccoppiatori). Questa proprietà è da non sottovalutare per prevenire possibili rischi di danneggiamento di RECS 101 o peggio ancora di tutti i sistemi presenti nella rete a cui è connesso RECS 101: in questo modo si è sicuri che per qualsiasi operazione errata compiuta a valle dell'interfaccia il danno è comunque confinato al danneggiamento dell'interfaccia stessa.

DEVELOPER'S BOARD

Su richiesta dell'utente, Intellisystem Technologies è in grado di fornire una developer's board per il dispositivo RECS 101 (**fig. 17**).
La developer's board di RECS 101 è una scheda che integra al suo interno 16 switches per la simulazione dei 16 ingressi digitali di RECS 101 e 16 display a LED per le relative 16 uscite. Gli switch relativi ai 16 ingressi sono stati progettati per lavorare secondo logica TTL "Low Active", e i 16 display relativi all'output sono stati progettati per lavorare secondo logica TTL "High Active" compatibilmente alle specifiche di RECS 101.
Le **tabelle 1** e **2** riportate di segui-

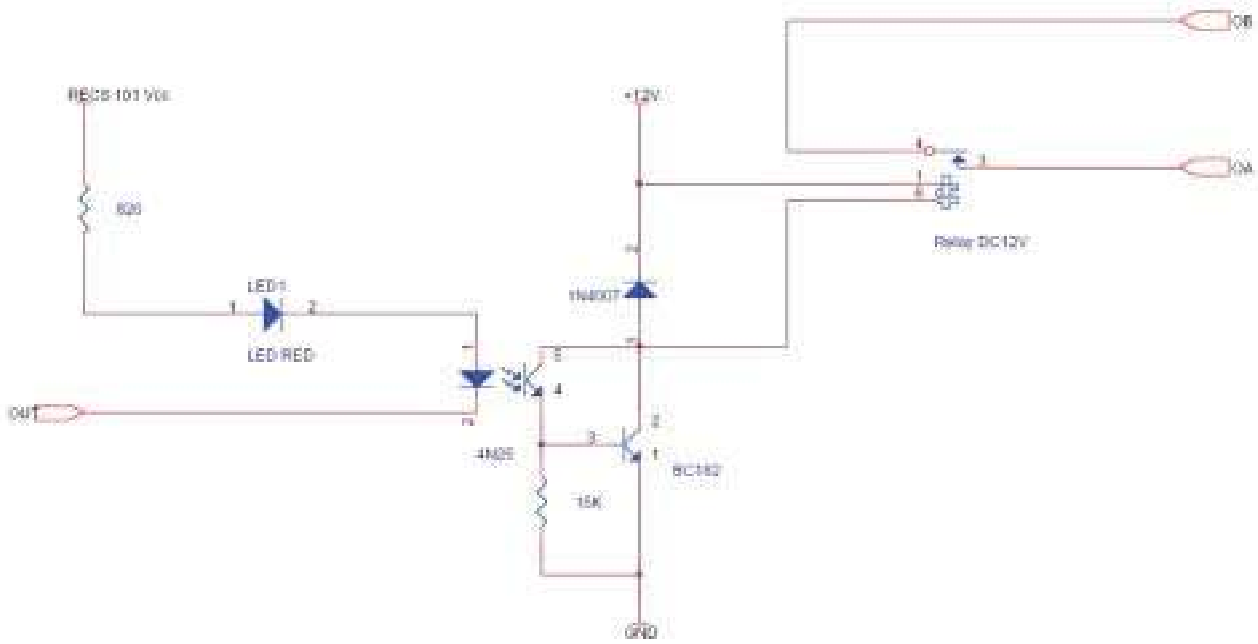
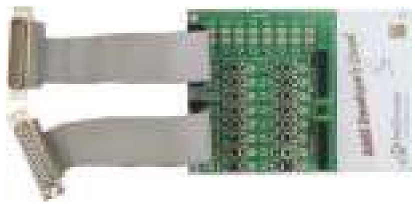


Figura 16 *Interfaccia per la connessione di un dispositivo mediante relay alla porta d'uscita di RECS 101*



RECS developer's board. **Figura 17**

to riassumono quanto detto in precedenza.
La RECS developer's board non necessita di alimentazione esterna poiché gli viene fornita direttamente da RECS 101 tramite i due connettori relativi all'IO.
Mediante la RECS developer's board è possibile progettare e sviluppare applicazioni senza aver a disposizione l'eventuale impianto da controllare. Dotata di due connettori ausiliari la RECS developer's board permette allo sviluppatore di estendere le sue funzionalità ad altri dispositivi elettronici in modo da poter effettuare velocemente le comuni operazioni di debugging delle applicazioni.
La **fig. 18** riporta la piedinatura dei connettori ausiliari descritti in precedenza.

Developer's board Switch	RECS input connector	Input LED Status
Chiuso	LOW	ON
Aperto	HIGH	OFF

Logica "Low Active". **Tabella 1**

RECS Switch (Application)	RECS output connector	Developer's board LED Status
Chiuso	HIGH	ON
Aperto	LOW	OFF

Logica "High Active". **Tabella 2**

Digital Input

Digital Output

LED	Pin	Note	Tasto	Pin	Note
1	3		1	3	
2	4		2	4	
3	5		3	5	
4	6		4	6	
5	7		5	7	
6	8		6	8	
7	9		7	9	
8	10		8	10	
9	11		9	11	
10	12		10	12	
11	13		11	13	
12	14		12	14	
13	15		13	15	
14	16		14	16	
15	17		15	17	
16	18		16	18	
	1	Vcc +5v		1	Vcc +5v
	2	Vcc +5v		2	Vcc +5v
	19	GND		19	GND
	20	GND		20	GND

Piedinatura dei connettori ausiliari presenti nella RECS developer's board. **Figura 18**

Nel prossimo numero si discuterà dei seguenti argomenti riguardanti RECS 101:

1) Protocollo di comunicazione implementato in RECS 101.
2) Monitor dello stato di I/O.
3) Controllo dei comandi di Output.
4) Comunicare con RECS 101: L'interfaccia Socket in C.
5) Comunicare con RECS 101: L'interfaccia Socket in Java.

BIBLIOGRAFIA

[1] Intellisystem Technologies "RECS 101 Manuale Utente", http://www.intellisystem.it

REALIZZAZIONI PRATICHE • TUTORIALS • RADIANTISTICA • COMPUTER HARDWARE • ROBOTICA

FareELETTRONICA

N° 216 - GIUGNO 2003 - ANNO 19

€ 4,13 - Frs 8,00

ALL'INTERNO LE PAGINE DI:

MHz FareELETTRONICARADIO

ROBOMANIA FareELETTRONICAROBOTICA

BASSA FREQUENZA

- FILTRO CROSSOVER ATTIVO 2 VIE
- UN ORIGINALE ANALIZZATORE DI SPETTRO

TUTORIAL

- LA PORTA PARALLELLA SPP

HARDWARE

- GUIDA ALL'USO DEI DISPLAY LCD INTELLIGENTI (III^a parte)
- IL BUS I^{2}C (III^a PARTE): TERMOMETRO DIGITALE CON LM75

TECNOLOGIE SPERIMENTALI

- MISSILISTICA AMATORIALE (I^a PARTE)

MHz

- RADIOCOMANDO CODIFICATO 433,92 MHz
- ASCOLTARE L'UNIVERSO VIA RADIO

ROBOMANIA

- IRON CLAW (II^a PARTE)
- REALIZZIAMO UN BEAM ROBOT EVOLUTO: IL SERVOCORE WALKER

Carillon

IL LETTORE PORTATILE DI MP3

Spedizione in A.P. - 45% - ART. 2 COMMA 20/B LEGGE 662/96 - Filiale di Milano. In caso di mancata consegna, restituire all'editore che si impegna a pagare la relativa tassa presso il CMP di Roserio - Milano

ISSN 1591-2272

30216

9 771591 227008

DTP studio editrice

www.farelettronica.com

RECS 101: UN WEB SERVER EMBEDDED PER APPLICAZIONI DI CONTROLLO REMOTO TRAMITE TCP/IP

terza parte

di Cristian Randieri
randieri@intellisystem.it

In questa terza parte della presentazione del dispositivo RECS 101 vengono affrontati i seguenti argomenti: il protocollo di comunicazione implementato in RECS 101 ed esempi di metodologie per la progettazione di applicazioni personalizzate mediante l'implementazione di socket Internet in C e in Java.

PROTOCOLLO DI COMUNICAZIONE IMPLEMENTATO IN RECS 101

RECS 101 effettua il controllo delle sue porte digitali mediante un interfaccia basata sui socket di Internet. Per ottenere il controllo remoto delle porte di I/O attraverso Internet, è necessario che l'interfaccia che gestisce i socket venga implementata nel PC dell'utente che intende collegarsi a RECS 101 attraverso il protocollo TCP/IP.
La potenzialità di RECS 101 consiste nel fatto che tale interfaccia può essere implementata indifferentemente mediante un' Applet Java (che viene eseguita all'interno del Web Browser che si collega al dispositivo RECS 101) o un'applicazione C/Java che utilizzi i socket di Internet (**figura 1**). Ovviamente per fare ciò occorre progettarle adeguatamente aderendo allo standard fissato dalle regole della suite di protocolli TCP/IP. Tali interfacce si occuperanno quindi di inviare e ricevere i comandi per il controllo delle porte di I/O attraverso l'indirizzo IP impostato su RECS 101 e la relativa porta fissata alla 6001.RECS 101 si occuperà dell'interpretazione dei comandi di controllo ricevuti o trasmessi dal dispositivo elettronico da controllare ad esso connesso.
I comandi di controllo si suddividono in due categorie che identificano due operazioni diverse:

Monitor Stato I/O

Tramite quest'operazione è possibile avere informazioni inerenti lo stato di tutte le linee di I/O contenute nelle due porte a 16 bit di RECS 101. I comandi relativi a quest'operazione sono essenzialmente due:

- **I/O Get Command**: È il comando mediante il quale l'interfaccia socket interroga RECS 101 sullo stato delle proprie porte.
- **I/O Get Command Responce**: È il comando di risposta mediante il quale RECS 101 comunica all'interfaccia socket lo stato delle sue porte di I/O.

Controllo dell'Output

Questo tipo di operazione, gestita unicamente dal comando Output Set Command è utilizzata dall'interfaccia socket per settare i valori della porta d'Output di RECS 101. La **tabella 1** riassume i comandi relativi alla comu-

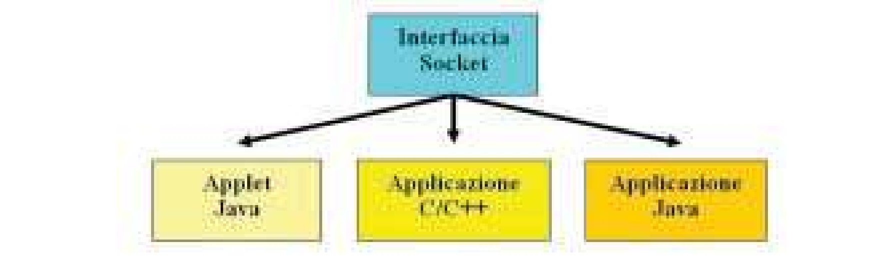


Figura 1: *Possibili scenari d'implementazione dell'interfaccia di comunicazione socket di RECS 101*

Figura 2: *Schema funzionale per la gestione di un dispositivo elettronico tramite RECS 101*

Figura 3: *Comandi di controllo di RECS 101*

Figura 4: *Comando di controllo della porta di Output di RECS 101*

nicazione e i tipi di messaggi che vengono scambiati tra l'interfaccia socket ed il dispositivo RECS 101.

Monitor dello stato di I/O

Lo stato della porta di I/O di RECS 101 è controllato mediante comandi gestiti tramite l'interfaccia socket che provvede a far dialogare il PC utente con RECS 101.
Più esattamente il comando che il PC utente deve inviare per ricevere da parte di RECS 101 lo stato delle porte di I/O è lo "0x75", che si compone di un byte.
Quando RECS 101 riceverà tale comando provvederà a comunicare lo stato delle porte di I/O utilizzando 4 byte come riportato in **tabella 2**.
Appare evidente che lo stato delle porte di I/O dipenderà dalla logica implementata dall'utilizzatore di RECS 101. Per esempio, supponendo che il circuito da interfacciare a RECS 101 sia stato progettato per lavorare secondo la tecnica "Active LOW" ciò equivale a dire che un ipotetico diodo Led collegato ad un'uscita della porta di Output sia acceso quando a quest'ultima viene inviato uno zero logico. Se adesso consideriamo il caso in cui i bit 0,2,4 e 10 della porta di Output siano nello stato logico alto e i bit 1,3 e 5 della porta di Input siano anch'essi nello stato logico alto, RECS 101 alla ricezione del comando "0x75" risponderà come descritto nella **tabella 3**.
A questo punto l'interfaccia socket tra RECS 101 ed il PC utente si occuperà dell'interpretazione di questo valore visualizzandolo sul PC utente.

Anche se i dati relativi allo stato delle porte di I/O sono contenuti in 4 byte, RECS 101 invierà all'interfaccia socket una parola di 16 bytes.
Di conseguenza l'utente dovrà interpretare solamente i primi 4 bytes del pacchetto ricevuto.
Ciò è dovuto al fatto che, come detto in precedenza, la trasmissione di queste informazioni avviene mediante i socket che operano tramite il protocollo TCP/IP che a sua volta opera sullo standard Ethernet.
Poiché lo standard Ethernet impone una lunghezza minima del pacchetto di 64 bytes (inclusi i gli headers IP e TCP) [1], e considerando il fatto che nel caso in cui venga generato un pacchetto la cui lunghezza minima è inferiore ai 64 bytes questi viene scartato, bisogna arrivare alla conclusione che anche se RECS ne avrebbe di bisogno solamente 4 si è costretti ad usarne16, di conseguenza, l'utente dovrà interpretare solamente i primi 4 bytes del pacchetto ricevuto (**tabella 4**).

Controllo dei comandi di Output

Questo tipo di comando viene utilizzato in tutti quei casi in cui si vuole modificare il valore di un bit della porta di Output di RECS 101 senza che venga generato un messaggio di conferma.
La **tabella 5** riporta il formato del relativo comando "0x76" che si compone di 4 bytes di cui il primo contiene il comando vero e proprio e gli altri due rappresentano il nuovo stato che la porta d'Output dovrà assumere.
Per esempio, supponiamo il caso in cui si voglia modificare lo stato della porta d'Output di RECS 101 settando allo stato logico "alto" i bit 0,1,2 e 3 lasciando tutti gli altri nello stato logico "basso". Allora poiché il corrispon-

Tipo di operazione	Comando	Direzione	
		PC Utente	RECS 101
Monitor Stato I/O	I/O Get Command	⟶	
	I/O Get Command Response	⟵	
Controllo dell'Output	Output Set Command	⟶	

Tabella 1: *Comandi relativi alla comunicazione e i tipi di messaggi che vengono scambiati tra l'interfaccia socket ed il dispositivo RECS 101*

Tipo	Numero di Byte			
	1	2	3	4
Comando per monitorare lo stato delle porte di I/O	0x75			
Risposta contenete lo stato delle porte di I/O	Stato della porta di Input		Stato della porta di Output	
	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**
	MSB	LSB	MSB	LSB
	Stato della porta di Input		**Stato della porta di Output**	

Tabella 2: *Controllo dello stato delle porte di I/O di RECS 101*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
1111	1011	1110	1010 = 0xFBEA																0000	0000	0010	1010 = 0x002A															
Stato della porta di Input																**Stato della porta di Output**															

Tabella 3: *Esempio di codifica dello stato della porta di I/O*

dente valore in esadecimale è 0x000F, occorrerà inviare a RECS 101 il valore esadecimale 76:00:0F come mostrato nella **tabella 6**.

COMUNICARE CON RECS 101: L'INTERFACCIA SOCKET IN C

Si riporta, di seguito, un esempio di codice sorgente scritto nel linguaggio C, il quale rappresenta l'implementazione di un'interfaccia socket basata sulle API dei socket di Berkely. I frammenti di codice riportati di seguito, si occupano di gestire rispettivamente il "Monitor Stato I/O " e il "Controllo dell'Output" descritti precedentemente.

Prendendo spunto da questi esempi l'utente oltre a capire i meccanismi di funzionamento descritti potrà essere capace di costruire una propria interfaccia personalizzata che funzionerà come applicazione, ovvero permetterà di gestire RECS 101 attraverso il protocollo TCP/IP ma senza il supporto di un Web Browser.

Un'applicazione di questo tipo potrebbe essere utili per tutte quelle esigenze di protezione e di riservatezza che escludano l'utilizzo di una tale interfaccia.

Come primo esempio si riporta la procedura IOMonitor che si occupa di monitorare lo stato delle porte di Input e di Output di RECS 101.

Per poter gestire tale operazione occorre per prima cosa definire due buffer rispettivamente commandBuf che conterrà il codice relativo al comando da inviare a RECS 101 e ResponseBuf che conterrà il valore letto nella porta di I/O. Occorrerà inoltre definire delle variabili di ausilio quali:

- **commandLen:** è un intero che contiene la lunghezza del comando relativo a commandBuf.
- **lenReceived:** intero che conterrà la lunghezza del buffer di ricezione ResponseBuf.
- **i:** variabile intera da utilizzare per i cicli iterativi.

2 bytes	2 bytes	60 bytes
Stato della porta di Input	**Stato della porta d'Output**	Dati non utilizzati

Tabella 4: *Formato del pacchetto ricevuto dall'interfaccia socket.*

Definite le variabili occorre inizializzare il Socket TCP mediante la chiamata alla procedura TCPSocketInit() che per brevità non viene riportata. Si passa quindi ad inizializzare il buffer che conterà il comando utilizzando la costante IOGet, che definita altrove, è uguale a 0x75 che rappresenta il codice esadecimale del comando Monitor Stato I/O.

A questo punto utilizzando l'istruzione sendto s'invia l'istruzione Monitor Stato I/O a RECS 101, inviando come parametri il valore del buffer e altre informazioni riguardanti l'indirizzo IP di RECS 101.

Poiché la funzione sendto restituisce un valore che è uguale a -1 in caso d'errore, al verificarsi di quest'evento sarà visualizzato un opportuno messaggio d'errore indicante un problema riscontrato durante la comunicazione con il dispositivo.

Inviato il comando Monitor Stato I/O bisogna predisporre l'interfaccia socket a ricevere le informazioni che scaturiscono dall'interrogazione fatta.

Per prima cosa bisogna allocare i buffer di ricezione ResponseBuf, dopodiché mediante l'istruzione recvfrom (che è la corrispondente dell'istruzione sendto nel caso della ricezione) si riceveranno

Tipo	Numero di Bytes		
Comando di controllo della porta d'Output	1	2	3
	0x76	Stato della porta di Output	

Tabella 5: *Formato del commando di controllo della porta di Output*

Comando di controllo della porta d'Output	MSB Porta Output								LSB Porta Output							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0x76	0x00								0x0F							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	0000|0000|0010|1111 = 0x000F															
	Stato della porta d'output															

Tabella 6: *Esempio di modifica dello stato della porta d'uscita di RECS 101*

le informazioni relative allo stato della porta di I/O di RECS 101. Poiché l'istruzione recvform restituisce un valore uguale a -1, in caso d'errore, è possibile implementare delle istruzioni che avvertano nel caso in cui ci siano stati degli errori di comunicazione. Supponendo che non ci sono stati errori durante la comunicazione, si può passare alla visualizzazione dello stato delle porte di I/O mediante la lettura del buffer di ricezione Response Buf.
La procedura può terminare chiudendo il Socket TCP e rilasciando le locazioni di memoria allocate per la gestione dei buffer (vedi **listato 1**) Il secondo esempio che si riporta serve a variare lo stato della porta di Otuput di RECS 101.
In particolare si riporta come esempio la procedura per modificare un solo bit della porta di Output che una volta selezionato verrà portato a livello logico alto. Per tale scopo adopereremo la procedura SetOutput(). Come nel caso precedente iniziamo con le dichiarazioni delle variabili locali:

- **commandBuf:** è un array di caratteri che conterrà i tre byte che compongono il comando Output Set Command.
- **commandLen:** è un intero che contiene la lunghezza in byte dell'istruzione Output Set Command.
- **outbit:** è un intero inizializzato al valore zero che conterrà la posizione del bit della porta di Output che si vuole modificare.
- **outdata:** è un intero inizializzato al valore 0x0001 che viene utilizzato come maschera per la modifica del singolo bit che compone la parola relativa alla porta di Output.

La prima operazione da svolgere è quella di richiedere quale bit all'interno della parole che compone la porta di Output si vuole modificare.
Tale valore è quindi memorizzato nella variabile outbit. Richiamiamo la procedura IOMonitor() per leggere lo stato della porta di I/O che verrà memorizzato all'interno dell'array IOStatus.
Da notare che IOStatus [2] conterrà la parte MSB della porta di Output e IOStatus [3] conterrà la parte LSB.

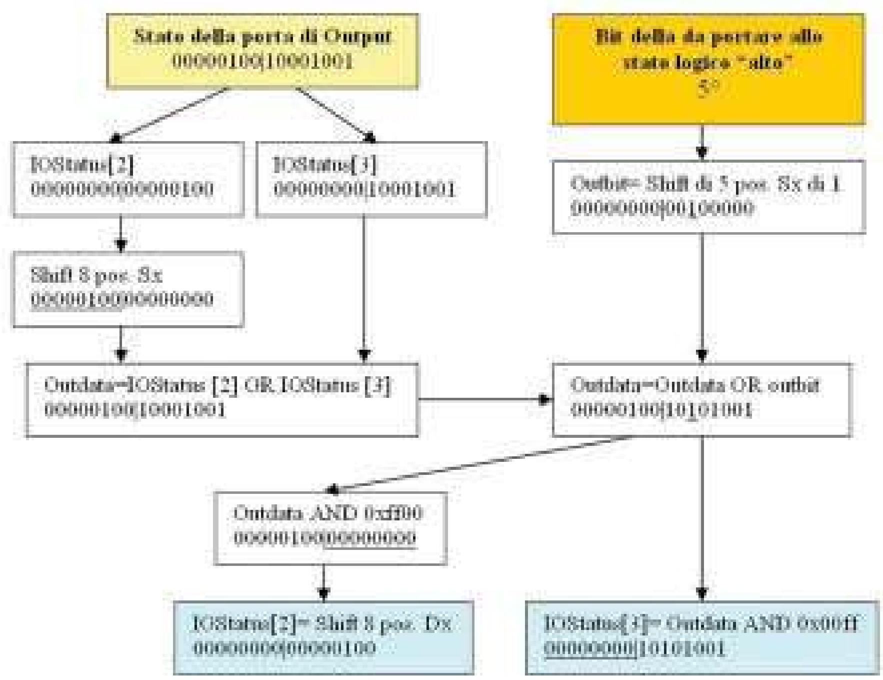


Figura 5: *Esempio che illustra l'algoritmo per settare a livello logico "alto" il 5° bit della porta di Output di RECS 101*

A questo punto occorre ristabilire una connessione con RECS 101 pertanto reinizializziamo il Socket tramite la procedura TCPSocketInit(). Poiché in C quando si definisce una variabile di tipo int questa viene allocata all'interno di una cella di memoria di 16 bit sia IOStatus [2] che IOStatus [3] saranno contenuti in due celle da 16 bit. Occorre quindi fare in modo che queste siano compattate come unico valore a 16 bit, tale operazione viene svolta eseguendo l'operazione logica sui bit di IOStatus [2] e IOStauts [3]:

outdata=(Shift di 8 posizioni verso sinistra di IOStatus[2]) OR (IOStatus [3])

Quanto appena detto viene espletato da un unica istruzione riportata nel listato:

outdata|=((IOStatus[2]<<8) | IOStatus[3]);

Essendo il nostro obiettivo portare a livello logico alto solamente il bit selezionato tramite la variabile outbit, l'operazione necessaria da fare è quella di utilizzare la variabile outdata precedentemente inizializzata ad 1 e farla shiftare (a livello di bit) di tante posizioni verso la sinistra rispetto al valore di outdata, in questo modo il bit posto inizialmente uguale ad i in outdata si posizionerà alla relativa posizione lasciando tutti gli altri bit uguali a zero. Quanto detto si riassume nella seguente pseudo istruzione:

outdata= outdata OR (1 Shift di outbit posizioni verso sinistra)

Che si traduce nella seguente istruzione C:

outdata |= (int) (1 << outbit);

A questo punto la variabile outdata conterrà il nuovo valore dello stato della porta di Out con il bit selezionato portato a livello logico alto. Occorre

LISTATO 1

```
//--------------------------------------------
// RECS 101: Esempio di programmazione di
//            un interfaccia socket in C
// Procedura IOMonitor
// By Intellisystem Technologies
// http://www.intellisystem.it
//--------------------------------------------
void IOMonitor()
 {
  char commandBuf, *ResponseBuf ;
  int commandLen, lenReceived ;
  int i ;

  // Inizializzazione del Socket TCP
  TCPSocketInit() ;

  // Esegui I comandi TCP
  commandBuf = IOGet ;
  commandLen = 1 ;

  // Invia I comandi a RECS 101
  err = sendto (sock,&commandBuf,commandLen,0,
       (struct sockaddr*)&clientAddr,sizeof(clientAddr));
  if (err == -1)
   {
    perror("\n Errore nell'invio dei dati !! \n");
    exit (1);
   }

  // Allocazione di un buffer per I pacchetti
  // in ingresso, 1 pacchetto = 16 bytes
  ResponseBuf = calloc(0x10, sizeof(char)) ;

  // Ricezione dei pacchetti entranti
  lenReceived = recvfrom (sock,ResponseBuf,0x10,0,
             (struct sockaddr*)&clientAddr,&clientLen);
  if (lenReceived < 0)
  {
   perror("\n Errore nella ricezione dei dati??? \n") ;
   exit(0) ;
  }

 // Visualizza la dimensione dei pacchetti entranti
 printf("\n%d N. di bytes ricevuti -> \n", lenReceived);

 // Memorizza lo stato delle porte di I/O
 // per usi futuri nell'array IOStatus
 for (i=0; i<4; i++)
  IOStatus[ i]  = ResponseBuf[ i] ;

 // Visualizza lo stato delle porte di I/O
 printf("\n\n* Stato delle porte di I/O di RECS 101 *\n") ;
 printf("Porta di Input : %x:%x\t\t Porta di Output : %x:%x",
         IOStatus[ 0] , IOStatus[ 1] , IOStatus[ 2] , IOStatus[ 3] ) ;
 printf("***** Intellisystem Technologies *****\n") ;

 // Rilascia le allocazioni di memoria allocate per il buffer
 free(ResponseBuf) ;

 // Chiude il Socket TCP
 TCPSocketClose() ;
}
```

adesso prepararsi per eseguire il comando Output Set Command. Per fare ciò dobbiamo riempire il buffer commandBuf di tre byte rispettivamente, uno per il codice istruzione 0x76 e i rimanenti che conterranno la parte MSB e LSB del nuovo stato della porta di Output. Adoperando le seguenti istruzioni:

IOStatus[2]=(outdata&0xff00)>> 8;
IOStatus[3] = (outdata & 0x00ff) ;

Facciamo in modo che IOStatus [2] contenga la parte MSB del nuovo stato della porta di Output, il ché si ottiene eseguendo la seguente operazione logica sui bit di outdata:

IOStatus[2]= Shift di 8 posizioni verso destra (outdata AND 11111111|00000000)

Per il secondo caso sarà sufficiente eseguire solamente la seguente operazione logica sui bit di outdata:

IOStatus[3]= outdata AND 11111111|00000000

Riempito il buffer che conterrà il comando da inviare a RECS 101, non ci rimane che adoperare l'istruzione sendto per rendere tale comando operativo.
Si ricorda che tale istruzione restituisce un valore che nel caso sia -1 indica che l'informazione non è stata trasmessa correttamente.
Per concludere, l'ultima operazione da fare è quella di chiudere il socket mediante la chiamata alla procedura TCPSocketClose.(vedi **listato 2**)

Per maggiore chiarezza la **figura 5** riporta un esempio pratico di quanto descritto precedentemente, pertanto si supporrà quanto segue:

- Lo stato della porta di Output di RECS 101 è uguale a 00000000|10001001.
- Si vuole portare a livello logico "alto"

LISTATO 2

```
//--------------------------------------------
// RECS 101: Esempio di programmazione di
//              un interfaccia socket in C
// Procedura SetOutput
// By Intellisystem Technologies
// http://www.intellisystem.it
//--------------------------------------------
void SetOutput ()
{
 char commandBuf[ 3] ;
 int commandLen ;
 int outbit=0, outdata=0x0000 ;
 int err ;

 // Richiede quale bit si vuole portare a livello logico
 // alto della porta di Output
 printf(" Prego selezionare il bit della porta d'Output
  di cui si vuole invertire lo stato logico "alto"(0-15) :");
 scanf("%d", &outbit) ;

 // Legge lo stato corrente delle porte di I/O
 IOMonitor() ;

 // Re-Initializza il Socket TCP
 TCPSocketInit() ;

 // Determina il nuovo valore della porta d'Output a partire
 // dallo stato attuale delle uscite
 outdata = ((IOStatus[ 2]<<8) | IOStatus[ 3]) ;
 outdata |= (int) (1 << outbit);
 // Or operation with currentle selected Bit
 // Memorizza il nuovo stato della porta di Output
 IOStatus[ 2] = (outdata & 0xff00)>> 8 ;
 IOStatus[ 3] = (outdata & 0x00ff) ;

 // Costruisci il buffer che conterrà Il comando
 // Output Set Command
 // 1) Command ID IOSet=0x76
 commandBuf[ 0] = IOSet ;

 // 2) Output status set
 commandBuf[ 1] = (BYTE) ((outdata & 0xff00) >> 8) ;
 commandBuf[ 2] = (BYTE) (outdata & 0x00ff) ;
 commandLen = 3 ;

 // Invia I comandi a RECS 101
 err = sendto (sock,&commandBuf,commandLen,0,
    (struct sockaddr*)&clientAddr,sizeof(clientAddr)) ;
 if (err == -1 )
 {
  perror("\n Errore nell'invio dei dati !! \n");
  exit (1);
 }

 // Chiude il Socket TCP
 TCPSocketClose() ;
}
```

LISTATO 3

```
//------------------------------------------
// RECS 101: Esempio di programmazione di un interfaccia
// socket in Java
// Procedura readIOport
// By Intellisystem Technologies
//------------------------------------------

public int readIOport()
{
  Socket socketTCP = null;
  int tmp = 0;
  int inputData = 0;
  byte rxData[] = new byte[16];
  byte data[] = {COMMAND_GET};

  try {
    socketTCP=new Socket(InetAddress.getByName(m_host), m_port);
    socketTCP.setTcpNoDelay(true);
    socketTCP.getOutputStream().write(data, 0, data.length);
    instream=new DataInputStream(socketTCP.getInputStream());
    tmp = instream.read(rxData, 0, rxData.length);
    if (tmp != -1)
    {
      inputData = (int) (rxData[2] << 8 | (rxData[3] & 0x00ff));
      inputData &= 0xffff;
    }
    socketTCP.close();
    instream.close();
  }
  catch (Exception e)
  {
    System.out.println("Err : " + e);
  }
  return inputData;
}

//------------------------------------------
// Procedura writeOutputPort
//------------------------------------------

public void writeOutputPort(int outdata)
{

  Socket socketTCP = null;
  byte[] data = new byte[4];
  data[0] = COMMAND_SET;
  data[1] = (byte) ((outdata >> 8) & 0x000000ff);
  data[2] = (byte) (outdata & 0x000000ff);
  // Initialize socket
  try {
   socketTCP=new Socket(InetAddress.getByName(m_host), m_port);
   socketTCP.setTcpNoDelay(trujhe);
   socketTCP.getOutputStream().write(data, 0, data.length);
    socketTCP.close();
  }
  catch (Exception e)
  {
    System.out.println("Err: " + e);
  }
}
```

il quinto bit della porta di Output a partire dalla destra.

COMUNICARE CON RECS 101: L'INTERFACCIA SOCKET IN JAVA

Come detto in precedenza per superare tutte le limitazioni dovute alla gestione di RECS 101 mediante un software applicativo la soluzione proposta da Intellisystem Technologies utilizza la tecnologia Java che prevede la creazione di un'Applet di controllo, gestita mediante un interfaccia browser. Come ben noto, le Applet sono dei programmi autonomi, scritti in Java, eseguibili mediante un comune browser. La potenzialità di un software scritto in Java, consente di essere totalmente indipendenti dalla piattaforma HW su cui si esegue l'applicazione. Senza entrare troppo nei dettagli della programmazione in Java riportiamo di seguito un frammento di codice Java riguardante un esempio d'implementazione dell'interfaccia socket basata su Applet che permette la ricezione e trasmissione di degnali di I/O, attraverso il protocollo TCP. Il lettore più attento può paragonare i codici seguenti con quelli scritti in C ed evidenziare quindi le analogie in termini di funzionalità. (**listato 3**)

BIBLIOGRAFIA

[1] Introduzione allo stack TCP/IP,IntellisystemTechnologie, http://www.intellisystem.it/download.htm

[2] Netid Managed Services, Information technology, Northwestern Technology, http://gradeswww.acns.nwu.edu/ist/snap/doc/sniffing.html.

[3] Internet spoofing reference page, http:// www.brd.ie / paper/ sslpaper / hyperlin.html.

REALIZZAZIONI PRATICHE • TUTORIALS • RADIANTISTICA • COMPUTER HARDWARE • ROBOTICA

FareELETTRONICA

N° 217/218 - LUGLIO/AGOSTO 2003 - ANNO 19 € 6,00 - Frs 12,00 NUMERO DOPPIO

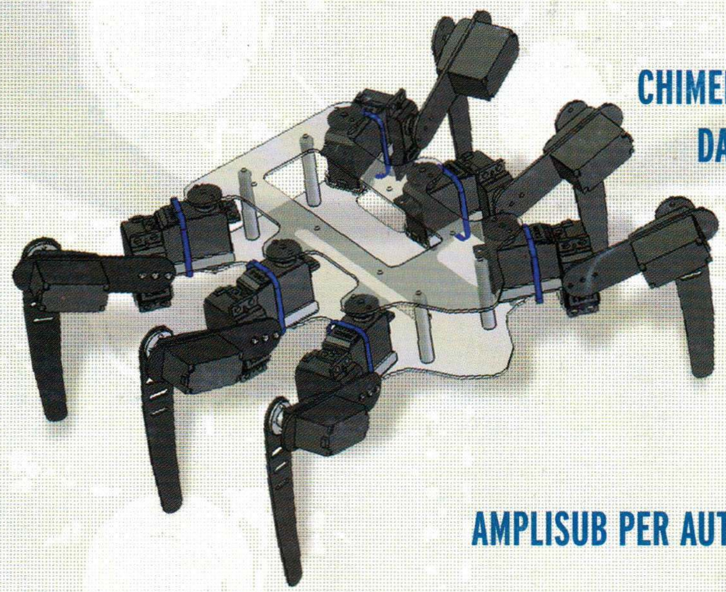
ALL'INTERNO LE PAGINE DI:

MHz FareELETTRONICARADIO

ROBOMANIA FareELETTRONICAROBOTICA

VITAMINA C

LA PRIMA PARTE DEL NUOVO E DIVERTENTE CORSO SUL LINGUAGGIO C



CHIMERA, UN ISOPODE DALLO SCHELETRO DI METALLO

TUTORIAL

- LA PORTA PARALLELLA EPP

HARDWARE

- USB PER TUTTI!
- ACCENSIONE ELETTRONICA PER AUTO
- COMMUTATORE PER HARD DISK

TECNOLOGIE SPERIMENTALI

- CARILLON (2ª PARTE)
- PROGETTIAMO UN RAZZO (2ª PARTE)

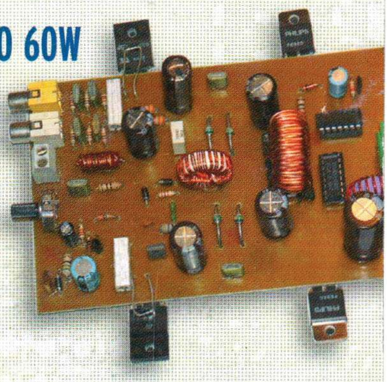
STRUMENTAZIONE

- ALIMENTATORE 0÷30 V 10 A

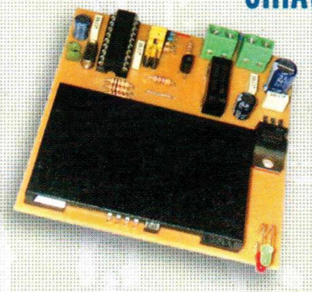
ROBOMANIA

- REALIZZIAMO UNA PIATTAFORMA MOTORIZZATA SU DUE RUOTE

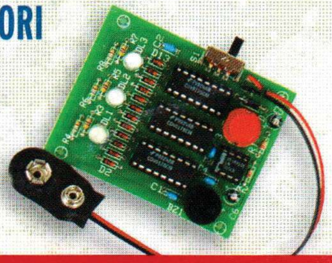
AMPLISUB PER AUTO 60W



CHIAVE ELETTRONICA CON CHIP CARD



UNA SLOT MACHINE A COLORI



Spedizione in A.P. - 45% - ART. 2 COMMA 20/B LEGGE 662/96 - Filiale di Milano. In caso di mancata consegna restituire all'editore che si impegna a pagare la relativa tassa presso il CMP di Roserio - Milano

ISSN 1591-2272

30218

9 771591 227008

DTP studio editrice

www.farelettronica.com

RECS 101: UN WEB SERVER EMBEDDED PER APPLICAZIONI DI CONTROLLO REMOTO TRAMITE TCP/IP

quarta parte

di Cristian Randieri
randieri@intellisystem.it

Con questa quarta parte, si conclude la trattazione del dispositivo RECS 101 con un argomento di rilevante importanza: "il proble della sicurezza per i web server embedded".

La sicurezza è un aspetto molto importante e da non trascurare nei sistemi di controllo specie se sono gestiti tramite la rete Internet. Se da un lato la rete Internet offre grandi flessibilità a livello di condivisione di risorse e di gestione da remoto, dall'altro è sicuramente un ambiente non sicuro, poiché chiunque può connettersi ad essa. Il web ha il potere di aumentare la produttività di chiunque, tuttavia come per ogni tecnologia o attività di gruppo oltre alle straordinarie attività occorre considerarne i rischi. Generalmente gli attacchi ad un sistema di controllo remoto si possono classificare mediante l'individuazione dei punti deboli del sistema che s'intende esaminare. In generale si possono individuare quattro categorie:

- Vulnerabilità dei dati.
- Vulnerabilità del software.
- Vulnerabilità del sistema fisico.
- Vulnerabilità delle trasmissioni.

Per difendersi da questi attacchi, ci si deve attendere che ogni potenziale intruso possa sfruttare queste vulnerabilità per accedere ai dati e quindi potenzialmente danneggiare il sistema. Ad esempio, la connessione di un sistema di controllo su internet può aprire delle falle nei sistemi di sicurezza e tali falle possono essere utilizzate da utenti non autorizzati per accedere o manipolarne le funzionalità. Gli intrusi potrebbero anche invalidare il server Internet del sistema di controllo, modificandone i file in esso memorizzati (ad esempio i file che contengono le informazioni sulle User-ID e Password degli utenti del sistema). I potenziali Hacker potrebbero inserire nel sistema dei virus e altri programmi distruttivi auto-replicanti in grado di danneggiare o disabilitare completamente il sistema. Possiamo classificare gli attacchi provenienti da internet nelle seguenti categorie:

Attacchi da password: Gli intrusi cercano di entrare nel sistema immettendo un codice di Login ed una password, provando varie volte sino a trovarne una funzionante [1]. Normalmente vengono adoperati dei software in grado di utilizzare variegati dizionari che provano di continuo diverse combinazioni sino a quando non trovano quella vincente che permette di far accedere al sistema. Ad esempio i sistemi Unix sono particolarmente vulnerabili ad attacchi di questo tipo, poiché, UNIX non blocca l'accesso degli utenti dopo un determinato numero di tentativi falliti, cosa che normalmente avviene nella maggior parte degli altri sistemi operativi.

Attacchi alla sicurezza della rete e dei pacchetti: Poiché ogni pacchetto trasmesso in Internet può attraversare un gran numero di nodi prima di giungere a destinazione, gli hacker possono utilizzare appositi strumenti denominati "racket sniffer" per intercettare i pacchetti inoltrati nella rete (inclusi i pacchetti di login e trasmissione dei dati). I più comuni attacchi ai pacchetti sono precursori degli attacchi al protocollo IP. Per iniziare un attacco sniffing, un hacker per prima cosa va alla ricerca di una User

ID e di una password di un utente legittimo utilizzandola per accedere alla rete distribuita. Dopo essersi intruso nella rete l'hacker osserva e copia le trasmissioni dei pacchetti e tenta di raccogliere quante più informazioni possibili sulla rete.

Attacchi al protocollo IP: Si concentra sull'indirizzamento dei pacchetti che il protocollo IP utilizza per le trasmissioni. Un attacco di questo tipo prevede due fasi. Nella prima si cerca di determinare l'indirizzo IP del server, generalmente mettendosi in ascolto dei pacchetti Internet, provando a specificare in ordine vari numeri di host oppure connettendosi al sito mediante un browser web e osservando l'indirizzo IP nella barra di stato. Poiché l'hacker sa che gli altri computer della rete condividono una parte del dell'indirizzo IP del server, cercherà di simulare un indirizzo IP che gli consenta di scavalcare il router e di accedere al sistema, come se fosse un utente interno. Dopo che l'hacker avrà iniziato a trovare gli indirizzi della rete, inizierà anche a controllare i numeri di sequenza dei pacchetti che si trasmettono tali computer. In seguito, dopo aver controllato le trasmissioni della rete, l'hacker cercherà di prevedere il prossimo numero di sequenza che verrà generato dal server e quindi fornirà un proprio pacchetto con tale numero di sequenza inserendosi fra il server e l'utente. Poiché l'hacker ha già l'indirizzo IP del server, può in realtà generare pacchetti con i numeri di sequenza corretti e indirizzi IP che gli consentono di intercettare le trasmissioni con l'utente. Dopo che l'hacker ha avuto accesso al sistema tramite la previsione di un numero di sequenza, può accedere alle informazioni che il sistema di comunicazione trasmette al server, inclusi i files di password, nomi, login, dati riservati e ogni altra informazioni trasmessa in rete. In generale un hacker utilizza la previsione del numero di sequenza come preparativo per l'attacco vero e proprio al server oppure come base per l'attacco di un altro server della rete.

Hyperlink Spoofing: È un tipo d'attacco che gli hacker sferrato contro computer che comunicano utilizzando il protocollo HTTP [2]. Gli hacker possono dunque sferrare attacchi anche al protocollo di autenticazione di server SSL (Secure Socket Layer) utilizzato per la creazione di browser e server Web sicuri, come i prodotti Microsoft e Netscape. Un attacco di questo tipo prevede che un hacker fungendo da intermediario convinca il browser a connettersi a un server fittizio presentando al browser l'aspetto di una sessione sicura. Un hacker intermediario è un hacker che s'inserisce nel flusso dei pacchetti che scorrono fra un client ed un server. In questo modo l'hacker convince l'utente a rilevare determinate informazioni quali ad esempio User ID e Password o altre informazioni riservate che saranno memorizzate nel server fittizio. Un alto rischio di Hyperlink spoofing accade se l'utente preleva ed esegue dal server fittizio applet Java pericolosi, credendo che tali applet siano forniti da un server sicuro e che debbano pertanto essere considerati sicuri. L'attacco Hyperlink spoofing rende palese un difetto nel modo in cui, la maggior parte dei browser, impiega i certificati digitali per rendere sicure le sessioni. L'attacco spoofing tramite collegamenti ipertestuali non attacca la crittografia a basso livello o il funzionamento del protocollo SSL. Di conseguenza l'attacco può essere sferrato anche ad altre applicazioni garantite da un certificato, a seconda del modo in cui tali applicazioni impieghino i propri certificati. Il problema principale è che gli attacchi Hyperlink spoofing si basano sul fatto che il certificato SSL fornito contiene informazioni errate (ad esempio il nome del DNS). Quindi, nonostante gli indirizzi URL sembrino corretti e riflettendo l'attività dell'azienda che possiede il sito Web cui si fa riferimento, non sempre questo accade. Quando è registrato un dominio, le autorità Internet assicurano che il DNS non sia già stato registrato da altri ma non assicurano che non violi le leggi di copyright.

Web Spoofing: È un tipo d'attacco che prevede di creare una copia falsa ma convincente dell'intero sito Web [3]. Il sito Web ha tutto l'aspetto del sito vero e proprio, ovvero contiene le stesse pagine e gli stessi link del vero sito WEB, ma è completamente sotto il controllo dell'hacker. In un attacco di questo tipo, l'hacker può osservare o modificare tutti i dati che vanno dalla vittima al server del sito Web. Inoltre, l'hacker può controllare tutto il traffico di ritorno dal server Web alla sua vittima. In seguito l'hacker può impiegare vari tipi di attacco tra cui ad esempio lo sniffing e lo spoofing. Con lo sniffing l'hacker osserva passivamente il traffico della rete. Lo spoofing invece prevede un'attività di manipolazione in quanto l'hacker convince un host di essere un altro computer fidato e pertanto si prepara a ricevere varie informazioni. Ad esempio l'hacker può registrare i contenuti e le risposte che il server invia al client (User ID, password ecc.). L'hacker può eseguire un'attività di sorveglianza, anche se la vittima ritiene di trovarsi in una connessione sicura.
Indipendentemente dal fatto che la connessione impieghi i metodi SSL o S-http, l'hacker sarà comunque in grado di ingannare l'utente. Si potrebbe pensare che sia difficile per l'hacker sostituirsi all'intero Web, ma sfortunatamente non è così. L'hacker non deve memorizzare l'intero contenuto del Web, poiché il Web è, per definizione, disponibile on-line.

Quando il server dell'hacker deve fornire una falsa pagina, gli basta prelevarla e modificarla dal Web stesso.

POSSIBILI CONTROMISURE

Sebbene il mondo dell'informatica sia in continua evoluzione trovare dei rimedi che eliminino definitivamente tali problemi è molto difficile, tuttavia nel seguente paragrafo vogliamo presentare alcune soluzioni che adottate potrebbero essere un modo per fronteggiare queste problematiche. Rispecchiando lo schema precedente riportiamo di seguito le soluzioni possibili:

Attacchi da password: Nelle reti, l'intercettazione delle transazioni, rappresenta uno dei rischi più gravi che attualmente affligge i singoli utenti e le organizzazioni. Per proteggersi dall'intercettazione dei pacchetti è opportuno crittografare tutte le trasmissioni. I due tipi principali di crittografia sono: la crittografia a chiave semplice (o a chiave simmetrica) e quella a chiave pubblica (o a chiave asimmetrica). La crittografia a chiave semplice, utilizza un'unica chiave nota ai due capi della comunicazione che la usano per crittografare e decrittografare le informazioni.
La crittografia a chiave pubblica, usa una chiave disponibile pubblicamente e una segreta conosciuta dall'utente. La maggior parte dei programmi normalmente utilizzati per eseguire la crittografia dei messaggi, seguono lo standard PEM (Privacy Enanched Mail) definito in dettaglio nelle RFC 1421, 1422,1423 e 1424. Gli algoritmi di crittografia più utilizzati sono l'algoritmo RSA (Rivest-Shamir-Adleman) e l'algoritmo Diffide-Hellman. Tali algoritmi possono quindi essere utilizzati per marcare in modo digitale le trasmissioni. Questa tecnica consente ai destinatari dei messaggi di verificare l'identità del mittente. Studi recenti hanno dimostrato che misurando accuratamente il tempo necessario per eseguire le operazioni sulla chiave privata, un hacker può dedurre gli esponenti fissi di Diffide-Hellman, i fattori delle chiavi RSA e dunque violare questi sistemi di crittografia. In termini realistici, il pericolo che qualcuno possa decodificare una trasmissione criptata, utilizzando un attacco di questo tipo, è solo leggermente inferiore rispetto al pericolo che qualcuno possa rubare la chiave privata dal disco fisso.

Attacchi alla sicurezza della rete e dei pacchetti: Gli attacchi sniffer su reti distribuite possono essere evitati utilizzando degli schemi di identificazione come il sistema delle password monouso o il sistema di autenticazione a ticket (come Kerberos [4]). Alcuni sistemi monouso forniscono agli utenti la prossima password nel momento in cui l'utente si connette dal sistema. Anche se sia le password monouso che gli schemi Kerberos possono rendere molto più difficile lo sniffing delle password su una rete non sicura, entrambi i metodi espongono al rischio di attacchi attivi se il canale dati non è criptato o codificato. Un attacco attivo al protocollo TCP/IP consente all'hacker di ridirezionare il canale TCP verso la propria macchina. Dopodiché l'hacker può by-passare la protezione che offre un sistema di password monouso o di autenticazione a ticket. La connessione TCP, diviene vulnerabile, a chiunque sia in possesso di uno sniffer di pacchetti TCP e di un generatore di pacchetti TCP posizionati sul percorso della connessione.

Attacchi al protocollo IP: Il modo più semplice per prevenire il sistema contro questo tipo di attacchi a previsione di numero di sequenza consiste nell'assicurarsi che il router, il firewall e ogni server del sistema abbiano attivato la protezione audit-trail. Un audit-trail è una registrazione cronologica delle attività di sistema, sufficiente per consentire la ricostruzione, la revisione e l'esame della sequenza di situazioni e di attività che hanno riguardato o che hanno condotto a un'operazione, una procedura o un evento in una transazione dal suo inizio ai suoi risultati finali. Utilizzando gli audit-trail, si può osservare quando un hacker tenta di attraversare il router e il firewall e quando tenta di accedere al server. Utilizzando uno dei programmi di servizio disponibili nel sistema operativo, si può richiedere che a seguito di un determinato numero di richieste di accesso negate venga prodotto un avvertimento. Si deve riconoscere che l'auditing e la manutenzione e l'osservazione degli audit-trail non offrono una protezione " a prova d'errore" contro gli attacchi al sistema. Se qualcuno esegue lo "spoofing" del sistema, ad esempio, l'operazione non potrà essere individuata dall'auditing. Se qualcuno ascolterà il sistema con uno sniffer, l'auditing probabilmente non si accorgerà di nulla poiché l'hacker non accede ai dati del server ma semplicemente osserva i dati in passaggio.
Come tutti gli altri strumenti di prevenzione degli attacchi, l'auditing-trail, se utilizzato correttamente, è solo uno degli strumenti per un piano organico di sicurezza. L'auditing non può sostituire un firewall o uno screening router o una politica di sicurezza. Analogamente gli altri sistemi difensivi non possono sostituire l'auditing.

Hyperlink Spoofing: Se s'impiegano già applicazioni Web che fanno affidamento sull'autenticazione del server (ad esempio per il prelevamento di applet Java), l'unica soluzione praticabile consiste nel far partire il browser da una pagina sicura in modo che gli utenti possano fidarsi dei link iniziali e che un hacker non possa mai inviarli in luoghi sospetti.

Una pagina sicura è quella di cui si può verificare l'integrità e questo, in genere, significa che tale pagina deve essere un file HTML locale o una pagina su un server SSL. Se si desidera che il browser di un utente parta aprendo una pagina SSL, si deve inviare l'indirizzo URL di tale pagina tramite mezzi difficili o impossibili da intercettare (ad esempio un floppy o una lettera), altrimenti la pagina potrebbe diventare il punto di partenza per l'attacco che s'intende prevenire. Tutti i link contenuti in questa pagina dovrebbero inviare gli utenti su siti di provata affidabilità e preferibilmente tutti i link dovrebbero essere di tipo SSL. L'affidabilità può basarsi sui seguenti criteri:

- Il sito deve essere condotto con criteri di sicurezza. Ovvero l'intero sito deve essere reso sicuro contro gli attacchi e l'intercettazione delle pagine.
- Il sito deve contenere link che conducono solo ad altri siti sicuri.

Web Spoofing: Questo genere d'attacchi è veramente pericoloso e in sostanza non è rilevabile. Le misure preventive che possono essere adottate si riassumono nei seguenti punti:

- Disabilitare nel browser gli script in modo che l'hacker non possa nascondere l'evidenza dell'attacco;
- Assicurarsi che la riga degli indirizzi del browser sia sempre visibile.
- Fare attenzione all'indirizzo URL visualizzato dal browser, assicurandosi che punti sempre al server a cui si pensa di essere connessi.

Questa strategia riduce fortemente i rischi di attacco anche se un hacker può comunque colpire un utente dalla rete, specialmente coloro che non si preoccupano di osservare strani comportamenti sulla riga degli indirizzi o della barra di stato. Con la disattivazione degli script si perderà qualche utile funzionalità, si potrà in ogni caso riattivarne l'uso, all'interno di siti fidati, per disattivarli, nuovamente, quando si lascia il sito fidato.
La creazione di una soluzione a lungo termine è molto più difficile, poiché occorrerebbe modificare il codice del browser in modo tale che programma visualizzi sempre la riga dell'indirizzo offrendo una maggiore sicurezza così come la possibilità di rendere sicuro il browser contro modifiche esterne, ovvero fare in modo che i programmi Web non possano creare false barre di menù, false barre di stato ecc.
Per le pagine che il browser preleva utilizzando una connessione sicura, una migliore indicazione di attivazione della connessione sicura potrebbe aiutare a garantire un'effettiva sicurezza dell'utente.
Invece di indicare semplicemente l'attivazione di una connessione sicura, i browser potrebbero visualizzare con chiarezza il nome del server che ha completato tale connessione.
Fondamentalmente ogni approccio al problema del Web-spoofing sembra essere affidato alla vigilanza dell'utente. Il fatto che un amministratore di sistema possa realisticamente attendersi questo tipo di vigilanza da tutti gli utenti della rete, pone seri dubbi.

I PRINCIPALI PROBLEMI DI SICUREZZA DEGLI APPLET JAVA

Anche se Java rappresenta un ambiente di programmazione relativamente sicuro, occorre considerare vari argomenti che aiutino a difendersi contro i problemi di sicurezza derivanti dall'impiego di Java. Poiché la JVM interpreta gli applet Java localmente, in genere gli applet consumano grandi quantità di risorse di sistema. Gli applet ostili o mal programmati possono consumare troppe risorse di sistema utilizzando la maggior parte della CPU o della memoria de computer.
Quando un applet consuma troppe risorse, il computer può rallentare sino quasi a bloccarsi.
Questo stato di blocco è il risultato di un attacco. Nelle prime implementazioni di Java (JDK 1.1.2) esisteva un bug nel verificatore di applet che consentiva a un applet prelevato su un client che si trova all'interno di un firewall di collegarsi a un determinato host al di là del firewall. Dopo la connessione, l'applet poteva trasmettere informazioni relative alla macchina client invece che informazioni relative al server proxy così come dovrebbe fare l'applet, aprendo la rete a un attacco spoofing.

In generale possiamo dire che il Java può soffrire di quattro tipi possibili di attacchi [5÷13]:

- **Leakage** (unauthorized attempts to obtain information belonging to or intended for someone else).
- **Tampering** (unauthorized changing/including deleting/of information).
- **Resource stealing** (unauthorized use of resources or facilities such as memory or disk space).
- **Antagonism** (interactions not resulting in a gain for the intruder

Installazione Cache	SAND BOX	Accesso Completo Trusted
	Applet java Standard *	Applet java Trusted *
Installazione permanente	Libreria Java standard *	Libreria Java Trusted *
	* Utilizzo dell'autenticazione	

Tabella 1: Sisitema di sicurezza Sandbox di Java

but annoying for the attacked party).

Gli Applet Java utilizzano un sistema di sicurezza, noto con il nome di Sandbox, che protegge il computer contro l'intrusione di applet ostili. Il modello Sandbox limita l'accesso al sistema da parte del applet restringendolo a determinate aree del client.
La **tabella 1** si riferisce ad un applet Java di tipo standard chiamato applet sandbox. L'applet sandbox ha un accesso limitato alle risorse del sistema. Un applet sandbox non può ad esempio accedere al disco fisso dell'utente, aprire nuovi canali di trasmissione o restituire informazioni approfondite, relative al client che esegue l'applet stessa.
Gli applet e la libreria standard java, sono applet sandbox. Il tipo trusted è una nuova variante del modello java, un applet trusted ha accesso a tutte le risorse di sistema e opera all'esterno della sandbox.
In genere, gli applet java trusted, sono applet creati da un'organizzazione o all'interno di un'intranet aziendale, oppure applet che l'autore firma prima della trasmissione via internet. In generale non è possibile garantire la sicurezza degli applet trusted in quanto l'applet ha un accesso completo alle risorse del sistema.

ARCHITETTURA DI SICUREZZA JAVA

In accordo a quanto riportato in letteratura [8], l'applet Verifier [9], è una parte del sistema run-time di java, che assicura che l'applet segua determinate regole di sicurezza.
Per iniziare, l'applet verifier conferma che il file della classe segua le specifiche del linguaggio java. L'applet verifier non presume che il file della classe sia stato prodotto da un compilatore sicuro.
Al contrario controlla nel file della classe l'esistenza di violazioni alle regole del linguaggio e altre restrizioni riguardanti lo spazio dei nomi e chiudere altre via di fuga, impiegabili per uscire dal file della classe. In particolare l'applet verifier assicura che:

- Il programma non provochi l'overflow o l'underflow dello stack.
- Il programma esegua accessi validi alla memoria e ai registri.
- I parametri di tutte le istruzioni bytecode siano corretti.
- Il programma non converta illegalmente i dati.

L'applet verifier svolge queste funzioni critiche analizzando le istruzioni contenute nel file dell'applet. Un browser Web utilizza un solo class loader che il browser attiva all'avvio, dopo questa fase il browser non può estendere, modificare o sostituire il caricatore di class. Gli applet non possono creare o far riferimento a un proprio class loader.
L'applet verifier è indipendente dall'implementazione di riferimento Sun del compilatore java e dalle specifiche di alto livello del linguaggio Java.
L'applet verifier esamina il bytecode generato dal compilatore java. La JVM si fida (e pertanto esegue) del byte code importato da internet solo dopo che tale bytecode ha passato l'analisi del verifier. Per passare al verifier il bytecode deve rispondere alla sintassi, alle firme degli oggetti, al formato del file della classe ed altre prevedibilità dello stack run-time definiti dall'implementazione.
Gli applet sono eseguiti in condizioni di sicurezza relativamente stringenti. L'applet security manager è il meccanismo java che si occupa delle restrizioni sugli applet. Un browser ha un solo manager della sicurezza. L'applet security manager si inizializza all'avvio del browser e in seguito non può essere sostituito, modificato o esteso.
Gli applet non possono creare o far riferimento a un proprio gestore della sicurezza. Per una descrizione più dettagliata dell'architettura di sicurezza Java si rimanda alle note bibliografiche [5÷13].

RECS 101 SECURITY

Come evidenziato in precedenza, RECS101, rappresenta un implementazione realistica di un web server integrato capace di gestire al suo interno la JVM.
Il problema della sicurezza nel nostro caso presenta diversi vincoli non indifferenti che riguardano le scarse capacità di calcolo del dispositivo realizzato.
Sicuramente è quasi impensabile poter implementare tutti i sistemi anti-intrusione presentati nei paragrafi precedenti. Nonostante ciò gioca a nostro favore il fatto che essendo un dispositivo non standard che non ha al suo interno un sistema operativo standard ciò permette di sfruttare le proprietà hardware del dispositivo.
Per essere più chiaro, i problemi per cui il dispositivo potrebbe essere vulnerabile, sono principalmente dovuti a:

- Possibile attacco alla password d'accesso al sistema.
- Attacchi al protocollo IP e alla sicurezza dei pacchetti.
- Hyperlink Spoofing.
- Web Spoofing.

Poiché la nostra applicazione si basa sulla JVM, abbiamo un livello di protezione base che comunque ci viene fornito dalla gestione delle Applet (come esposto precedentemente). Le contromisure che abbiamo adottato per far fronte alle problematiche sopra esposte, sono le seguenti:

Possibile attacco alla password d'accesso al sistema. RECS 101 non integra al suo interno alcun sistema

di gestione delle chiavi sia esse pubbliche che private, trattandosi di un sistema embedded dedicato al controllo remoto di apparecchiature elettronico il suo utilizzo sarà sicuramente riservato ad una cerchia molto ristretta di utenti di conseguenza si può ipotizzare che gli utenti del sistema possano essere definiti a priori.
Ciò implica che il firmware del sistema deve essere programmato in modo tale da inserire tutti i possibili utenti del sistema.
La gestione del database delle password ed user ID è effettuata mediante l'applet Java Stessa che andrà a leggere un file crittografato posto all'interno del file system di RECS 101.

Attacchi al protocollo IP e alla sicurezza dei pacchetti. Un attacco di questo tipo viene in qualche moto ridotto mediante la crittografia del pacchetto contente lo stato delle porte di I/O.
Poiché come si è visto precedentemente le porte di I/O di RECS 101 sono codificate in un dato a 32 bit è possibile inserire un algoritmo di crittografia che possa proteggere il contenuto dei dati. Nel caso in cui RECS 101 venga utilizzato con una propria logica di controllo con operazioni di sniffing è pressoché impossibile risalire all'algoritmo di controllo del sistema poiché questo è contenuto all'interno dell'applet.

Hyperlink Spoofing & Web Spoofing. L'implementazione di un supporto SSL all'interno di RECS 101, per la sua complessità è pressoché impensabile. Di conseguenza un attacco Hyperlink Spoofing sarebbe possibile. Per evitare ciò si può pensare di adoperare due RECS 101 che lavorando in parallelo uno controlli gli stati dell'altro, in questo modo la probabilità che entrambi i sistemi vengano attaccati simultaneamente decresce di molto. Per quanto riguarda il Web Spoofing, si può pensare di disattivare il supporto Java in RECS 101, però il prezzo da pagare è la portabilità del dispositivo, nel senso che il dispositivo perderebbe tutte le proprietà inerenti l'accesso alle porte di I/O tramite interfaccia Web. Poiché RECS 101 supporta anche i Socket C è possibile scrivere delle applicazioni Client/Server in C che eseguite localmente in un PC possono attivare una connessione con quest'ultimo. In questo modo si risolvono tutti i possibili problemi di Hyperlink Spoofing e Web Spoofing.

Electronic shop 28

BIBLIOGRAFIA

[1] Netid Managed Services, Information technology, Northwestern Technology: *http://gradeswww.acns.nwu.edu/ist/snap/doc/sniffing.html*

[2] Internet spoofing reference page: *http://www.brd.ie/paper/sslpaper/hyperlin.html*

[3] Web Spoofing: An Internet Con Game: *http://www.cs.pronceton.edu/sip/pub/spoofing.html*

[4] B. C. Neuman and T. Ts'o. Kerberos: An Authentication Service for Computer Networks. In IEEE Communications, volume 39, pages 33–38.

[5] S.Fritzinger and M. Mueller. Java security, 1996. Sun Microsystems Incorporated, White Paper: *http://java.sun.com/security/whitepaper.txt*

[6] L. Gong. Secure Java Classloading. IEEE Internet Computing, 2(6):56{61, November/December 1998.

[7] C. Kerer. A exible and extensible security framework for Java code. Master's thesis, Distributed Systems Group, Technical University of Vienna, Austria, October 1999.

[8] G. McGraw and E. Felten. Java security and type safety. Byte, 22(1):63{64, January 1997.

[9] G. McGraw and E. W. Felten. Java security: hostile applets, holes, and antidotes. John Wiley, New York, 1997.

[10] G. McGraw and E. W. Felten. Securing Java: getting down to business with mobile code. John Wiley, New York, 1999.

[11] A. Rubin and D. E. Geer. Mobile Code Security. IEEE Internet Computing, 2(6):30{34, November/December 1998.

[12] Sun Microsystems, Incorporated. Secure computing with Java: now and the future, September 1998. White Paper: *http://java.sun.com/marketing/collateral/security.html*

[13] F. Yellin. Low level security in Java. In Proceedings of the Fourth International World Wide Web Conference, Boston, Massachusetts, USA, December 11{14, 1995, volume 1 of World Wide Web Journal. O'Reilly & Associates, Incorporated, November 1995. *http://www.w3.org/pub/Conferences/WWW4/Papers/197/40.html*

Il Giornale dell'Installatore Elettrico
ANNO 25 n. 9
Reed Business Information
SPEDIZIONE IN A.P. - 45% - ART. 2 COMMA 20/B LEGGE 662/96 - FILIALE DI MILANO - ISSN 0392-3630 - € 3,10
2003 25 MAGGIO

Master universitario di primo livello

"I sistemi di automazione e la sicurezza (security e safety) negli edifici: tecnologie e servizi per gli immobili residenziali, terziari, industriali e per il territorio": è questo il titolo del master universitario di primo livello in fase di avanzata progettazione presso il Politecnico di Milano (per informazioni: Politecnico di Milano - Dipartimento Best - Professor Oliviero Tronconi - tel. 02 23995896-7 - fax 02/23995897 - oliviero.tronconi@polimi.it.).
Il master, alla sua prima edizione e destinato a laureati in corsi di laurea triennali e specialistici, è organizzato in collaborazione con Anciss (Associazione italiana sicurezza ed automazione) - Anie e prevede il coinvolgimento di alcune tra le più importanti aziende di settore.
Il master (a pagamento) si svolgerà da ottobre 2003 a ottobre 2004, per una durata complessiva di 1500 ore suddivise in una parte teorica (600 ore), una di studio individuale (150 ore) e una di stage (750 ore).
La frequenza è obbligatoria e prevede una presenza giornaliera in aula di 5 ore di lezione e un impegno di studio ed esercitazioni individuali a casa per una durata di altre 3 ore (dal lunedì al venerdì). Lo stage presso le aziende sponsor avrà una durata di 8 ore giornaliere per un arco temporale da aprile 2004 a ottobre 2004.
La figura professionale prodotta sarà quella di un responsabile-manager nel settore della sicurezza e dell'automazione: un elevato profilo professionale con competenze di progettazione, organizzazione, gestione e commercializzazione dei sistemi di sicurezza e di automazione.
Tra gli argomenti trattati: l'evoluzione e la trasformazione degli edifici nella società terziaria; l'automazione degli edifici: verso l'edificio intelligente; la sicurezza: normative, principi e sistemi; area manageriale/commerciale.

Un modello di ambientazione

Salone del mobile: un'altra occasione, per BTicino, per proporre un modello di ambientazione domestica equipaggiata con il sistema domotico e teledomotico My Home e per farne conoscere le potenzialità tecnologiche.
Durante la manifestazione "Design Italia Show", infatti, ha presentato un modello lineare di casa domotica che ne riproduce alcuni sistemi integrati: un prototipo che ha consentito ai visitatori di interagire direttamente con le funzionalità e la vasta gamma di servizi di automazione domestica offerti da My Home.

Attenti alla provenienza!

30.000 lampade a marchio Philips, perfettamente contraffatte, sono state trovate – insieme ad altri prodotti "taroccati" - dalla Guardia di Finanza di Roma. Tutto il materiale - scoperto in due grandi capannoni, veri e propri ipermercati illegali "Made in China" - risulta perfettamente falsificato oppure illegalmente importato e privo delle autorizzazioni necessarie per essere messo in commercio nel nostro Paese.

Alcune lampade contraffatte sequestrate a Roma

I prodotti a marchio Philips sequestrati sono lampade a risparmio energetico prismatiche serie SL di vecchia generazione e lampade ai vapori di mercurio ad alta pressione Hpl: sottoposte all'analisi dei tecnici specialisti di Philips Lighting, è stata riscontrata un'eccellente riproduzione del packaging e del design della lampada (in particolare quelle a risparmio energetico), mentre sotto il profilo delle prestazioni a livello elettrico e illuminotecnico è stato riscontrato un bassissimo livello qualitativo e di sicurezza per l'impianto e per l'utente.
"Quando vengono offerte lampade Philips da intermediari non ufficiali a prezzi più bassi di quelli che Philips direttamente offre alla distribuzione, si deve essere consapevoli dei rischi a cui ci si espone acquistandole" ha affermato in una dichiarazione Aldo Bigatti, chairman di Philips Lighting Italia.

Il Giornale si è spostato

Siamo in tanti, siamo sempre di più, per offrire ai lettori sempre di più. E per affrontare nuove sfide e proporre iniziative interessanti e innovative, Reed Business Information si è dotata di una nuova, grande e prestigiosa sede, in viale Richard 1 a Milano (tel. 02818301): un rinnovamento logistico e stilistico creato per migliorare, sotto tutti i profili, la qualità dei servizi offerti e l'espressione delle nostre competenze.

Un **web** server integrabile per applicazioni **"Home Building Automation"** basate sul protocollo **TCP/IP**

La casa va in Internet

Cristian Randieri
Intellisystem Technologies

Un web server embedded è un web server progettato per lavorare all'interno di un sistema a microprocessore caratterizzato da risorse di calcolo limitate. Aggiungendo ad un tale dispositivo la programmazione tipica del Web unitamente alle proprietà di un linguaggio di programmazione ad alto livello quale il Java si ottengono interfacce di qualità, amichevoli (user friendly), a basso costo, cross platform (multi piattaforma), e network ready (pronte per lavorare in rete).

> Il browser web è diventato uno standard per lo sviluppo di interfacce utente di numerose applicazioni

Intellisystem Technologies presenta un nuovo dispositivo Recs 101, nato per far fronte alle esigenze di sviluppatori che intendono gestire applicazioni professionali per la Home Building Automation in ambiente TCP/IP in maniera veloce, facile e sicura.
Il comfort nell'abitazione è parte intrinseca determinante del benessere psichico di ognuno.
Il continuo adeguamento del modo di vivere alle nuove forme di comportamento sociale impone il trasferimento delle nuove abitudini alla propria abitazione come elemento indispensabile di continuità dell'aspetto comportamentale.
La continua evoluzione delle tecnologie basate sui sistemi digitali ha fortemente modificato le tecniche e metodologie usate nei sistemi di controllo dedicati alla Home Building Automation.
In particolare oggi la richiesta di processi distribuiti richiede sistemi intelligenti, dispositivi di controllo e sistemi di misura capaci di comunicare attraverso la rete. Un importante requisito di questi sistemi è l'esigenza di ridurre le connessioni, il che si traduce nel semplificare la gestione dei sistemi riducendone le problematiche inerenti alla manutenzione.

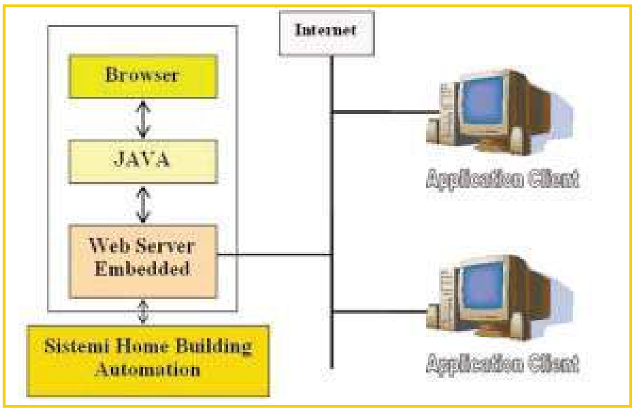


Fig. 1 - Architettura di un web server embedded

D'altro canto poiché il World Wide Web (la "ragnatela mondiale", o Web) è in continua evoluzione, per molte applicazioni commerciali e scientifiche il browser web è diventato uno standard per lo sviluppo di interfacce utente di numerose applicazioni. Questo perché i browser web sono capaci di fornire interfacce GUI a varie applicazioni client/server senza il bisogno di andare ad implementare dei software per il lato client. La soluzione migliore a questo tipo di esigenze è sicuramente data dall'utilizzo di un web server embedded connesso ad una infra-

TABELLA 1 - SPECIFICHE DEL DISPOSITIVO RECS 101

Specifica	Recs 101
CPU	Ubicom SX52BD (8 bit microprocessor, 50 MIPS)
Memoria	512 Kb flash memory (utilizzata per contenere le pagine web dell'utente)
Connessione di Rete	Interfaccia Ethernet 10 Base-T (IEEE802-3)
Connessione Utente	16 Ingressi digitali/16 Uscite digitali
Protocolli Internet Supportati	HTTP / BOOTP / TCP / UDP / IP ICMP / ARP Ethernet 802.3
Software di Utilità	Recs Utility (Piattaforma Windows) Web page uploader e cambio indirizzo IP

TABELLA 2 - PARAMETRI DI CONFIGURAZIONE DELL'APPLET PER LA GESTIONE AVANZATA DI RECS 101

Parametro	Fusione	Esempio	Obbligatorio	Possibilità di modifica
PDFOOK	Stringa inizializzazione Applet		Si	No
host	Indirizzo IP di RECS	host=value*172.16.10.103"	Si	Si
poet	Porta TCP per la comunicazione RECS 101	port value=6001	Si	No
polling	Intervallo di polling	polling value=1	Si	Si
Title	Interstazione Applet	Title value="RECS VO DEMO"	No	Si
ColTit	Colore da associare alla stringa Title	CotTit value="green"	No	Si
CAPL	Colore background Applet	CAPL value="yellow"	No	Si
Num Led	Numero ingressi da monitorare	Num Led value=16	Si	Si
NumB	Numero di pulsanti per il controllo delle uscite	NumB value16	Si	Si
TBT*	Testo da associare al pulsante*	TBT1 value="Comando 10"	No	Si
CTBT*	Colore del testo associato al titolo pulsante*	CTBT10 value="red"	No	Si
CLBF*	Colore di stato dell'uscita* quando questa si trovi nello stato "OFF"	CLBT10 value="blu"	No	Si
TLD*	Testo da associare al LED* relativo all'ingresso*	TLD 1 value="Luce Camera"	No	Si
CTLD*	Colore del testo associato al titolo del LED* relativo all'ingresso	CTLD1 value="black"	No	Si
CLIF*	Colore del tessuto al LED di stato dell'ingresso* quando quest'ultimo è nello stato "OFF"	CLIF10 value="green"	No	Si
CLIT*	Colore associato al LED di stato dell'ingresso "quando" quest'ultimo è nello stato "ON"	CLIT10=value "red"	No	Si

Fig. 3 - Il sistema Recs 101

struttura di rete al fine di fornire una interfaccia utente basata sull'utilizzo dell'ormai noto linguaggio HTML unitamente ad altre caratteristiche comuni ai web browser.
Se si pensa di aggiungere alle funzionalità ormai consolidate di un web server embedded la capacità di poter gestire applicazioni Java ecco che questi sistemi aprono le frontiere a capacità inesplorate, che li rendono capaci di eseguire i più variegati compiti quali, ad esempio, quelli di controllo remoto, supervisione e gestione di sistemi elettronici (figura 1).
L'implementazione delle funzionalità Java all'interno di un tale dispositivo è particolarmente indicato per questo approccio permettendo l'uso di una strategia di controllo indipendente dalla piattaforma hardware del sistema in cui viene gestita. Questa metodologia è stata da tempo adoperata nelle applicazioni Internet dove non sono richiesti stringenti vincoli di real-time. L'uso del linguaggio di programmazione Java per le applicazioni di controllo remoto fornisce il vantaggio di integrare sistemi di uso generale con Internet permettendone la supervisione ed il controllo.
Il nuovo concetto che intendiamo introdurre si basa sull'esecuzione di Applet Java (piccoli programmi aggiuntivi) per eseguire operazioni di controllo o di monitoraggio di dispositivi remoti.
In questo tipo di sistemi il controllo distribuito si ottiene mediante il trasferimento di pagine HTML e l'esecuzione di applet Java (figura 2).
I vantaggi dell'utilizzo di Java possono essere brevemente riassunti nei seguenti punti:
- indipendenza dalla piattaforma: diversamente dai comuni compilatori che producono codice per CPU specifiche, il Java produce un codice per una CPU virtuale. Al fine di rimanere indipendente da specifiche piattaforme hardware il sistema runtime di Java fornisce un'interfaccia universale per qualsiasi applicazione che si desidera sviluppare denominata JVM (Java Virtual Machine)
- Potenza: Java racchiude in sé nuove caratteristiche che includono la gestione dei database, l'invocazione dei metodi remoti ed altre caratteristiche inerenti la gestione della sicurezza
- Networking: Java nasce come linguaggio di programmazione distribuito, il che si traduce nel fatto che la sua progettazione includeva sin dall'inizio la gestione di particolari funzioni inerenti il networking quali, ad esempio, il TCP/IP, l'HTTP, l'FTP
- Efficienza: le moderne JVM, grazie all'utilizzo della tecnologia Just in Time (JIT) compiler, le performance d'esecuzione delle applet sono state fortemente migliorate.

Frutto di anni di ricerca e sperimentazione da parte di Intellisystem Technologies hanno portato alla realizzazione di Recs 101 (figura 3). Recs 101 è un dispositivo di facile utilizzo a prestazioni elevate, ideale per applicazioni di controllo remoto professionale. Una volta collegato ad una rete Ethernet, Recs 101 mette a disposizione dell'utente 32 canali digitali di cui 16 di Input e 16 di Output.

→

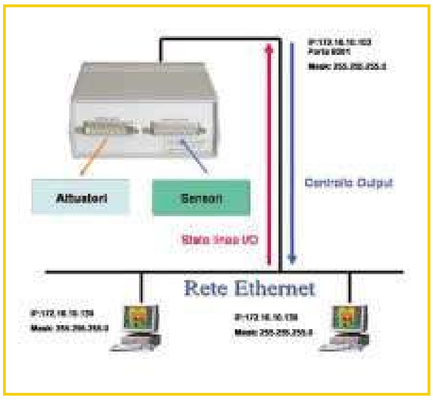


Fig. 4 - Scenario d'applicazione del dispositivo Recs 101

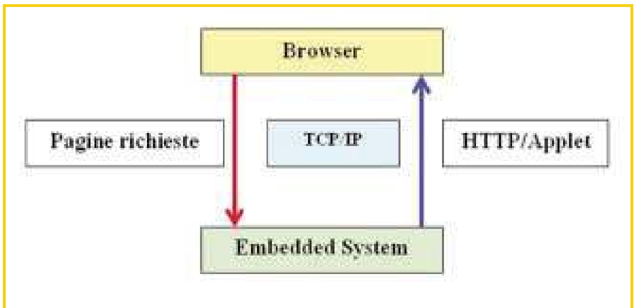


Fig. 2 - Applet Java per eseguire operazioni di controllo o di monitoraggio di dispositivi remoti

UN SOFTWARE SUL SITO

Per chi volesse dilettarsi a sperimentare la personalizzazione delle interfacce, Intellisystem Technologies mette a disposizione nel proprio sito tutto il software necessario (http://www.intellisystem.it/recs/Interfaccia.htm).

Figura 5 - Esempio di una possibile interfaccia GUI implementata in Recs 101

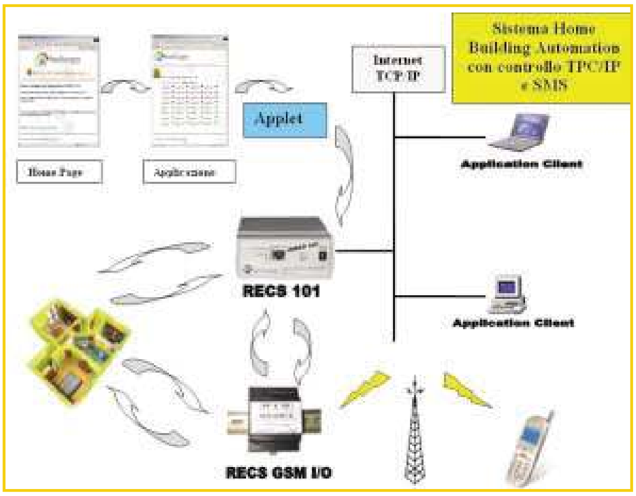


Figura 6 - Esempio di una possibile integrazione di Recs 101 con Recs GSM I/O in un sistema Home Building Automation

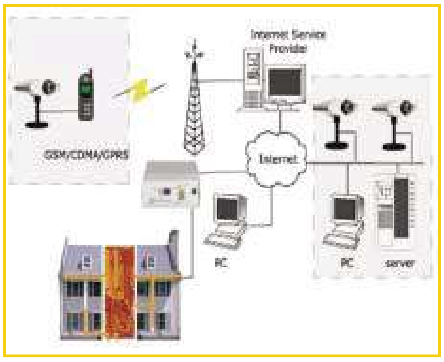


Figura 7 - Esempio di una possibile integrazione di Recs 101 con un sistema di videosorveglianza

Facile da installare e configurare, permette di sviluppare un'applicazione di controllo remoto in pochi e semplici passaggi. Supportato da qualsiasi browser internet quale Internet Explorer o Netscape permette di gestire totalmente da remoto qualsiasi dispositivo da controllare.

La figura 4 mostra lo schema architetturale semplificato di un possibile scenario d'applicazione di Recs 101.

L'architettura presentata permette la gestione di problematiche tipiche dei sistemi di controllo quali, ad esempio: acquisizione di segnali, azioni di controllo per mezzo di attuatori, l'elaborazione e la presentazione delle informazioni acquisite o manipolate.

La tabella 1 riporta le principali caratteristiche e specifiche del sistema proposto.

Recs 101 integra al suo interno un network processor dotato di interfaccia di rete Ethernet per connettersi direttamente a qualsiasi rete locale sia essa Internet sia Intranet. Ciò permette agli integratori/sviluppatori di sistemi e alle aziende produttrici di connettere i loro dispositivi direttamente ad Internet attraverso una rete Lan e, di conseguenza, di gestire da remoto il controllo totale dei loro dispositivi attraverso interfacce grafiche utente personalizzabili, accessibili mediante i comuni browser. Il web server integrato in Recs 101 è capace di gestire fino a 512k di documenti ed applicazioni web: tali risorse sono precaricate all'interno della memoria flash del dispositivo.

La fig. 7 è un esempio di una pagina web gestita da Recs 101 che può essere personalizzata per fornire informazioni statiche sul dispositivo quali, ad esempio, immagini, video, testi, file eccetera.

La caratteristica che rende unico tale dispositivo consiste nella capacità di poter usufruire del codice Java per la gestione dell'interfaccia relativa al controllo delle porte di I/O.

Tale caratteristica permette di poter gestire l'interfaccia utente tramite un'Applet Java parametrica: in questo modo l'utente finale può sviluppare la propria applicazione di controllo in modo molto veloce e sicuro senza dover essere in grado di programmare in Java.

La figura 5 riassume quanto detto in precedenza, ovvero partendo dalla home page del sito web contenuto in Recs 101 si accede all'interfaccia utente personalizzata che tramite un'applet di controllo interviene sulle porte d'input e di output per la gestione dell'hardware che si intende controllare.

Recs 101 è un dispositivo totalmente personalizzabile.

Viene fornito con tutto il software necessario allo sviluppatore per poter sviluppare rapidissimamente la propria applicazione in maniera facile e sicura.

Il software in dotazione contiene un'Applet di controllo che può essere personalizzata mediante i parametri riportati in tabella 2.

Facilmente integrabile con qualsiasi altro sistema offre una soluzione funzionale ed efficiente, per il telecontrollo distribuito.

Ad esempio, integrando Recs 101 con Recs GSM I/O (modulo GSM prodotto da Intellisystem Technologies provvisto di due ingressi e due uscite digitali gestibili tramite SMS) è possibile integrare tutte le tipiche funzionalità di gestione di sistemi per la Home Building Automation tramite Internet ed al tempo stesso tramite SMS, fornendo all'utente una piattaforma di controllo remoto multifunzionale non necessariamente legata ad un'infrastruttura di rete (figura 6).

Recs 101 trova ampio spazio nelle applicazioni di Home Building Automation legate ai moderni sistemi di videosorveglianza rappresentando un valido strumento per integrare tutte le funzionalità tipiche di un sistema di controllo remoto ai normali sistemi di monitoraggio video specie quelli che si basano sulla tecnologia TCP/IP (figura 7).

Con particolare riferimento al mondo del videocontrollo over IP, si intuisce facilmente che le soluzioni proposte da Intellisystem Technologies non hanno limiti in termini di funzionalità ed applicazioni ad hoc per tutte le esigenze dei più disparati utenti.

Ad esempio integrando i sistemi

L'utente finale può sviluppare la propria applicazione di controllo in modo molto veloce e sicuro

Recs con le ben note telecamere AXIS (di cui Intellisystem Technologies è partner tecnologico) si aprono le frontiere per un controllo totale di sistemi remoti, che nel caso della Home Building Automation si traducono in una presenza virtuale dell'individuo all'interno della propria dimora.

Sfruttando la combinazione vincente di tali sistemi si ottiene uno strumento completo capace di gestire immagini e di rilevare lo stato di dispositivi esterni, quali sensori e di manovrarne altri quali ad esempio attuatori.

In conclusione Recs 101, essendo un dispositivo totalmente flessibile nelle sue applicazioni, si presta come valido strumento per la reingegnerizzazione di macchinari a controllo semi-automatico, fornendo la possibilità di telecontrollare a distanza tramite Internet sistemi che sino ad oggi non prevedevano tale funzionalità.

servizio lettori 118

BIBLIOGRAFIA

1. McCombie, B.,"Embedded Web server now and in the future", Real-Time Magazine, no.1 March 1998, pp. 82-83.
2. Wilson, A.,"The Challenge of embedded Internet", Electronic Product Design, January 1998, pp. 31-2,34.
3. J. Gosling, B. Joy, G. Steele,"The Java Laguage Specification", http://java.sun.com
4. T. Lindholm, F. Yellin "The Java Virtual Machine Specification", 1996. http://java.sun.com
5. Intellisystem Technologies "Recs 101 Manuale Utente", http://www.intellisystem.it
6. Intellisystem Technologies "Recs 101 GSM I/O", http://www.intellisystem.it/prodotti.htm
7. Intellisystem Technologies "Reenginnering", http://www.intellisystem.it/servizi/reengineering.htm

Giugno 2003 www.ilb2b.it

Fieldbus & Networks

Hardware e software per l'automazione distribuita

DOSSIER
CONTROLLO E ASSISTENZA DA REMOTO

BUILDING AUTOMATION
FIELDBUS PER L'ENTERTAINMENT

vnu business publications italia

Supplemento ad Automazione Oggi n° 259 - Giugno 2003 - Spedizione in A... art. 2 comma 20/B legge ... Filiale di Milano - In caso di mancata consegna resti... ...ore che si impegna a pagare la relativa tassa presso il CMP di Roserio-Milano

viene creata da OPC Drive Server e costruisce la base dei link di comunicazione per le singole applicazioni. Ogni applicazione compatibile con l'interfaccia OPC può accedere alle funzioni dell'inverter. I vari server bus inglobano le specifiche dei differenti fieldbus, mentre Drive Server fornisce le caratteristiche dell'apparecchio, fungendo al contempo da server e da client OPC. Grazie alla sua architettura aperta è possibile impiegare anche server bus di altri produttori. La disponibilità di bus server Lenze comprende Lenze Lecom A/B/LI (RS-232/485 e fibra ottica), Lenze systembus CANopen e Lenze S7-MPI (interfaccia Siemens per PC), mentre Lenze S7-TCP è in preparazione.

Teleassistenza: Drive Server in azione

Manutenzione in remoto, registrazione di dati operativi, visualizzazione dei processi e messa in servizio semplificata sono solo alcune delle possibilità offerte dal sistema Lenze. La prima risulta importante soprattutto quando gli impianti vengono venduti in Paesi extraeuropei. Ad esempio, per la manutenzione in remoto dei propri frantoi da pietra la società austriaca PMT utilizza Drive Server con bus server S7-MPI. Questo tipo di impianti impiegano numerosi inverter e servoinverter, nonché un PLC S7-315 completo di adattatore per la teleassistenza e un modem. L'attività di manutenzione e diagnostica è svolta dalla stessa PMT, che si collega alle macchine direttamente dalla propria sede. Se ne ottiene un significativo risparmio di tempo e denaro. E' sufficiente disporre di un PC con il software di parametrizzazione Global Drive Control e del sistema Drive Server forniti da Lenze. Infine, la funzionalità di messa in servizio semplificata è stata particolarmente apprezzata dalla società ATZ di Mülsen St. Jacob, in Germania, che produce controlli per magazzini a sviluppo verticale. Normalmente, i controlli di posizionamento non vengono installati in un armadio di controllo, ma montati direttamente sul carrello; occorre quindi che un tecnico viaggi sull'apparecchiatura, a volte a grandi altezze, per impostare i parametri di controllo dell'azionamento. Grazie a Drive Server è possibile eseguire le regolazioni da terra, in modo semplice, tramite Profibus.

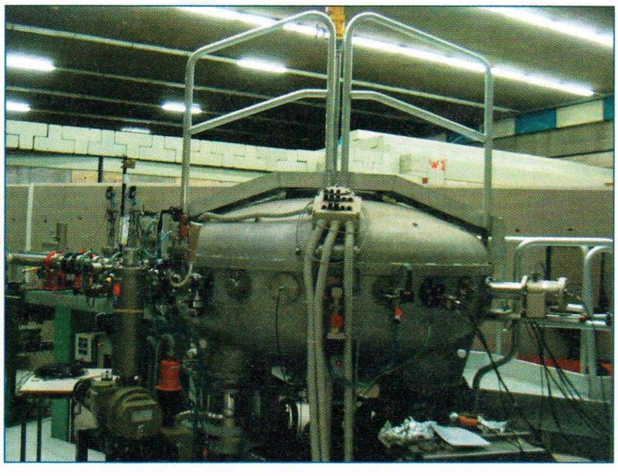
Esperimento Diamante

Intellisystem Technologies

Con il telecontrollo si possono ridurre i rischi di contaminazione da radiazione durante gli esperimenti di fisica nucleare

Situati a Catania, i Laboratori nazionali del sud dell'Istituto nazionale di fisica nucleare rappresentano da anni un centro mondiale di ricerca per la fisica nucleare sperimentale. Recentemente qui è stato condotto l'esperimento di fisica nucleare denominato 'Esperimento Diamante', che prevedeva il test di un rivelatore sperimentale al diamante (composto da un film di diamante depositato mediante la tecnica 'Microwave plasma enhanced chemical vapor deposition' su un substrato al silicio) per lo studio sistematico di un fascio di protoni ottenuto mediante un acceleratore di tipo tandem da 15 MV. Nelle applicazioni in cui è previsto un elevato livello di radiazioni l'utilizzo di un rivelatore al diamante ha dimostrato, meglio di ogni altro materiale, la capacità di caratterizzare la regione interna di un fascio di particelle emesse da un acceleratore. Misure di questo tipo rivestono particolare importanza per le applicazioni derivabili in ambito medico che utilizzino fasci di radiazioni. Solitamente tali misure vengono effettuate mediante dei rivelatori denominati 'tazze di Faraday', il cui limite sta nel fatto che il segnale da essi emesso è troppo basso perché se ne possa ottenere una risoluzione spaziale, il che limita la misura del profilo del fascio in esame.

Scopo dell'esperimento era porre a confronto le misure ottenute con tre diversi tipi di rivelatori, ossia al silicio, al diamante e basati sulle tazze di Faraday. Spesso durante gli esperimenti di fisica nucleare si pone maggiore attenzione ai sistemi di acquisizione dei dati che non agli apparati di gestione dell'esperimento, trascurando a volte tutta una serie di complicazioni puramente operazionali che possono limitare la corretta esecu-



Sala sperimentale dei Laboratori nazionali del sud dove si è svolto l'esperimento Diamante

zione dell'esperimento stesso. L'insorgere di questo tipo di problematiche, infatti, può ridurre drasticamente l'effettivo tempo di acquisizione dei dati sperimentali, poiché ogni qualvolta occorre accedere alla sala sperimentale, è necessario sospendere il fascio ed eseguire un nuovo set-up.

Facciamo qualche osservazione

Attualmente la maggior parte dei sistemi di supporto agli esperimenti di fisica nucleare sono attivati manualmente all'interno delle sale sperimentali. Durante ogni esperimento la presenza di radiazioni, pericolose sia per gli essere umani, sia per la strumentazione, richiede di porre le sale sperimentali in un cosiddetto stato di 'Safety control'; perciò occorre sospendere il fascio ogni qualvolta un operatore deve intervenire all'interno delle sale. Tra le sale sperimentali e quelle di acquisizione vengono approntate connessioni dedicate di tipo punto-punto; ciò significa che un gran numero di conduttori collegano le due zone, rappresentando di fatto un potenziale punto di diffusione delle radiazioni. I sistemi di telecontrollo e teleassistenza di ultima generazione proposti da Intellisystem Technologies, utilizzando la suite di protocolli TCP/IP permettono di ottimizzare le risorse logistiche, riducendo il numero delle connessioni necessarie, e riducono i costi di gestione. Consentono inoltre agli addetti di concentrarsi con maggiore libertà sul lavoro primario di acquisizione di dati e misure sperimentali.

A prova di radiazione

Prendendo parte all'esperimento Diamante, Intellisystem Technologies ha testato un valido strumento per il monitoraggio delle sale sperimentali utilizzando le moderne tecnologie di telecontrollo e teleassistenza. Il micro embedded Web server Recs 101 prodotto dall'azienda, in grado di controllare 16 ingessi e 16 uscite, è stato impiegato in combinazione con una videocamera Axis 2120, a sua volta collegata a un modulo audio Axis 2191. E' stato possibile disporre di una postazione di telecontrollo capace non solo di gestire immagini, ma anche di controllare lo stato di dispositivi esterni quali i sensori, e di manovrarne altri, quali gli attuatori. In particolare, il dispositivo Recs 101 è stato testato per la prima volta come sistema di supervisione e controllo per il posizionamento di un bersaglio mobile oggetto del test dell'esperimento, posto all'interno della camera a vuoto sperimentale ove ha avuto luogo il test. I risultati hanno mostrato l'ottima immunità della soluzione ai disturbi dovuti a radiazioni di tipo nucleare, nella fattispecie neutroni. L'integrazione del sistema Recs 101 con i dispositivi Axis e il sistema di movimentazione hanno dimostrato come sia possibile controllare da remoto tramite Internet un impianto sperimentale per la ricerca nel campo della fisica nucleare, ottimizzando costi e tempi d'esecuzione. Sono stati anche ridotti i tempi d'intervento degli operatori durante le varie fasi dell'esperimento, con conseguente diminuzione dei rischi da esposizione a radiazioni nucleari per gli addetti. Inoltre, per la prima volta più ricercatori da ogni parte del mondo hanno potuto connettersi alla sala sperimentale via Internet, tramite un canale video e uno audio bidirezionale, e controllare la movimentazione del bersaglio. ■

Cristian Randieri

Arriva la 'tele-illuminazione'

Merloni Progetti Energy Saving

Presso l'aeroporto di Bologna e il porto di Genova i punti luce vengono regolati in modo automatico in telegestione

In impianti di grande estensione assume particolare importanza la possibilità di gestire in modo centralizzato le utenze, ovvero di tenere sotto controllo lo stato degli impianti d'illuminazione e verificarne la funzionalità e l'efficienza.
Questa funzionalità viene chiamata 'telegestione' perché si fonda sulla trasmissione di dati da/verso gli apparecchi installati a/dal centro di controllo, nonché dal centro ai cellulari degli addetti al pronto intervento. Il controllo viene effettuato tramite un modem telefonico, un radio-modem o GSM.
I regolatori di Merloni Progetti Energy Saving comunicano con un PC in posizione centrale e inviano dati sullo stato degli impianti e sui parametri di funzionamento, ossia tensioni, correnti, sfasamenti, mancata accensione, intervento degli interruttori di protezione, ecc... Tutti questi dati sono memorizzati in una memoria centrale. Il software di telegestione 'Energy' di

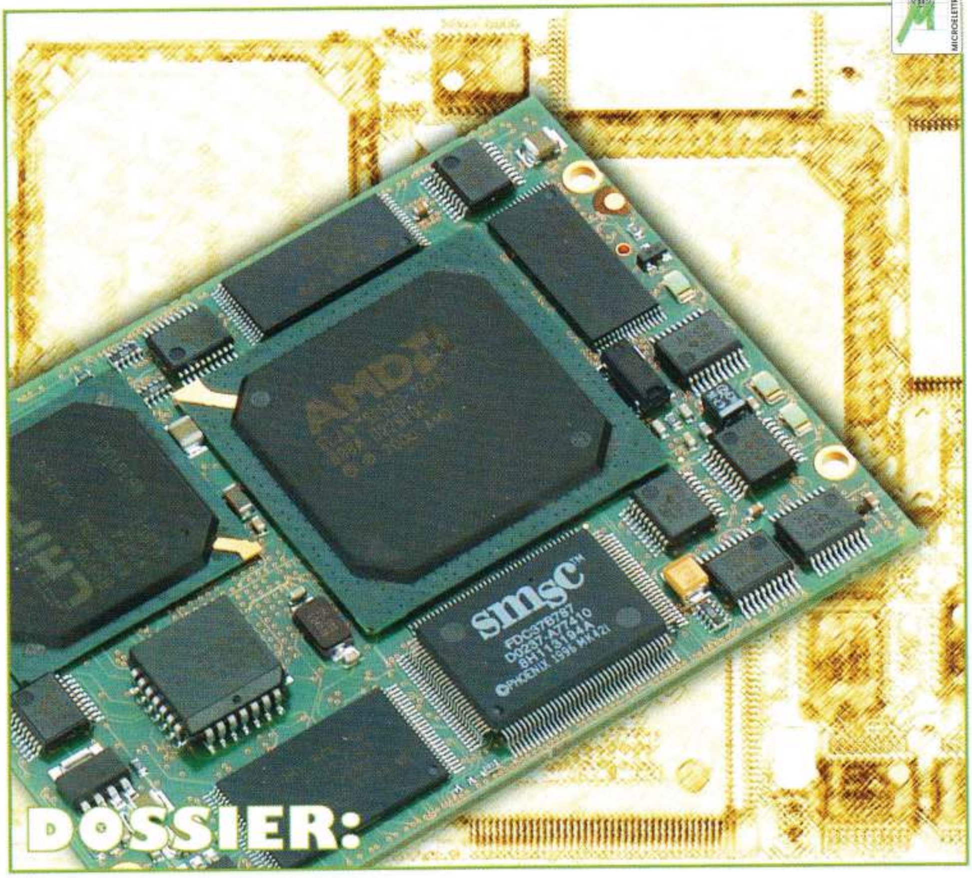
L'aeroporto di Bologna ha adottato i sistemi Merloni

www.ilb2b.it

elettronica OGGI

Mensile di elettronica professionale, componenti, strumentazione e tecnologie

Giugno 2003



DOSSIER: EMBEDDED

Spedizione in A.P. - 45% art.2 comma 20/B legge 662/96 - filiale di Milano Supplemento a Elettronica Oggi n° 324

vnu business publications italia

PER ULTERIORI INFORMAZIONI

Società	Tel	Fax	web
Adelsy	051763069	051763073	www.adelsy.it
Advantech Italia	029544961	0295449650	www.advantech.it
Automata	029639970	0296399731	www.automataweb.com
Astromed	0226411909	0226412828	www.astromed.com
Axiom Italia	02614299.1	0266400279	www.axiomitalia.it
Cimtech	0432853679	0432853736	www.cimtech.it
C.J.B. Computer Job	0303531883	030349557	www.cjb.it
Contradata Milano	0392301492	0392301489	www.contradata.it
Delo Systems	0290722441	0290722742	www.delo.it
Ele.Si.A.	017436531	01743653300	www.elesia.it
Eurolink Systems	06612401	0661240200	www.eurolinksystems.com
Eurotech	0433485411	0433485499	www.eurotech.it
Ganymed Computer	0119040816	0119006655	www.ganymed.com
Goma Elettronica	0117725024	011712298	www.gruppogoma.it
Grifo	051892052	051893661	www.grifo.it
I.M.A.	010593077	0105956925	www.imaweb.it
Intelligent Instrumentation	015980096	015980668	www.instrument.com
Intelligentsystem Technologies	0931703312	0931703312	www.intelligentsystem.it
Koan	035255235	035255235	www.koansoftware.com
Kontron Italia	0331827895	0331828703	www.kontron.com
National Instruments Italy	02413091	0241309215	www.ni.com/italy
N.C.S. Computer Italia	0331263975	0331263978	www.ncs-computer.com
Rafi Elettronica	024779181	02428880	www.rafi.it
Seco	057526979	0575350210	www.seco.it
Sistemi Avanzati Elettronici	015983206	015980668	www.sisav.it
Softing Industrial Solution Italia	0303733919	03032416014	www.softing.com
STMicroelectronics	0396031	0396035700	www.st.it
Vsystems	0119661319	0119662368	www.vsystems.it
Wind River	0117501511	011748247	www.windriver.com
Zelco Sistemi	0248011211	0248011247	www.zelco.it

Produttore	Innovative Integration	Innovative Integration	Intellisystem Technologies
Distributore	Vsystems	Vsystems	Intellisystem Technologies
Modello	Matador	Solomente	RECS 101
Tipo di bus	PCI		proprietario
Formato	Card	Stand alone board	-
Modalità trasf. Dati	DMA	USB	sincrona
Counter/Timer	-	-	-
Velocità di campionamento	-	-	20-100ms
Tipo di trigger	-	-	-
Ingressi analogici	sì	sì	nessuno
Numero	da 4 a 32	da 2 a 16	-
Risoluzione (bit)	da 14 a 24 bit	da 12 a 24 bit	-
Range	-	-	-
Guadagno	-	-	-
Modalità di campionamento	SigmaDelta su alcuni modelli	SigmaDelta su alcuni modelli	polling
Uscite analogiche	sì	sì	nessuna
Numero	da 4 a 16	da 1 a 16	
Risoluzione (bit)	da 14 a 24	da 12 a 24	16
Range	-	-	-
Ingressi digitali	sì	sì	sì
Numero	32	32	16
Compatibilità	-	-	-
Uscite digitali	sì	sì	sì
Numero	32	32	16
Compatibilità	-	-	-
Software	Pismo Development Environment	Zuma Development Environment	RECS Utility
Sistemi operativi supportati	MS Win, Linux	MS Win, Linux	Sistema operativo propietario
Linguaggi supportati	C, C++	C, C++	c/java
Protocollo comunicazione	-	-	TCP/IP
Funzione di analisi	-	-	Acquisizione dati su porta I/O 16 bit
Note	Schede di acquisizione ed elaborazione dotate di DSP TI67xx	A/D sino a 40MHz, 96KHz per conversione SigmaDelta a 24 bit	RECS 101 è un web server embedded con supporto funzionalità per la gestione delle Applet Java e soket c

Produttore	Axis Comunication	CJB Computer Job	Concurrent Technologies	Concurrent Technologies
Distributore	**Intellisystem**	**CJB Computer Job**	**Vsystems**	**Vsystems**
Modello	-	CJ370I/P	PP220	PP120
Microprocessore	Etrax 100 LX	Tutti i Celeron e Pentium!!! Anche Tualatin e VIA-C3, socket 370	Dual Xeon	Dual Pentium 3
Tipo di bus	proprietario AXIS	ISA e PISA	cPCI	cPCI
Formato	-	HalfSize operativo anche standalone (con o senza backplane)	Card	Card
Memoria di massa	esterna	2 interfacce EIDE UDMA	EIDE/Ultra SCSI 160	EIDE/Ultra SCSI 160
RAM	2MB Flash, 8MBSDRAM	fino a 512MB	sino a 4 Gbyte	sino a 1 Gbyte
Porte di comunicazione	62GPIO, 4 porte IDE, 2 USB	10	2 Gbit Ethernet	4 Gbit Ethernet
Seriale	4 porte seriali, 2 porte seriali sincrone	4 (RS232, una anche RS422/485)	2	2
Parallela	2 porte parallele	1	ECC/ECP	ECC/ECP
Altre	2 SCSI	1 CANbus integrata onboard, 2 USB, 2 LAN Ethernet 10/100	USB, PMC, IPMI, PICMG 2.16	USB, PMC, IPMI, PICMG 2.16
Sist. Operativi supportati	linux kernel 2.4	Windows (98, NT, 2000, XP), QNX (4.25 e 6.2), Linux RTAI	MS Win, Linux, QNX, Solaris	MS Win, Linux, QNX, Solaris
Note	Il sistema proposto, distribuito da Intellisystem integra tutte le funzionalità di un linux single board computer	Scheda CPU embedded progettata per diventare il motore di un sistema softPLC e softMotion dalle elevate prestazioni e costo contenuto	Svga embedded	Svga embedded

Settembre 2003 www.ilb2b.it

Fieldbus & Networks

Hardware e software per l'automazione distribuita

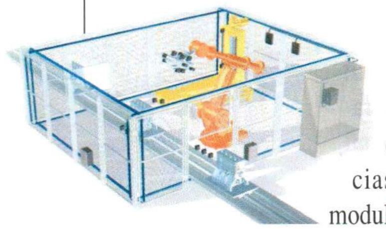
Primo piano
Il mondo wireless

Tavola rotonda
Internetworking: sì o no?

Dossier
Fieldbus a bordo macchina

Supplemento ad Automazione Oggi n° 261 - Settembre 2003 - Spedizione in A.P. - 45% - art. 2 comma 20/B legge 662/96 - filiale di Milano - In caso di mancata consegna restituire all'editore che si impegna a pagare la relativa tassa presso il CMP di Roserio-Milano

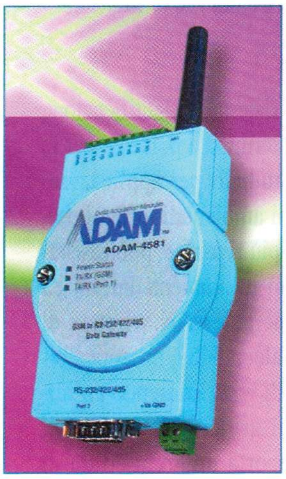
vnu business publications italia



attraverso il sistema FBP (FieldBusPlug) di ABB. La configurazione massima per ciascuna applicazione è di 5 moduli di I/O, per un totale di 300 WPS. L'alta affidabilità funzionale è garantita dalla continua verifica di 'presenza' di ogni sensore induttivo wireless; ciò permette la rapida rilevazione e segnalazione di eventuali malfunzionamenti. I messaggi di diagnostica provenienti dai sensori WPS sono aggiornati ogni 500 ms.

Raggiungibile ovunque, via GSM

Adam-4581 di **Advantech** è un modulo in grado di comunicare i dati sia con tecnologia GSM, sia via SMS tramite l'interfaccia GSM a 2 bande (900/1800). Può essere integrato in una vasta serie di applicazioni, nella gestione di impianti, nel monitoraggio delle acque, dei livelli e delle condutture, nel controllo strutturale di ponti e tunnel e all'interno di applicazioni di telecomunicazione in aree dove il cablaggio è critico o costoso, in applicazioni di sorveglianza. Come interfaccia Adam-4581 utilizza lo standard industriale RS-232/485 con controllo RTS auto-flow, mentre la funzione intelligente di trasmissione dati SMS permette una comunicazione trasparente tra i dati Ascii RS-232/485 e il segnale GSM AT. L'unità è in grado di convertire i comandi o i dati Ascii in un messaggio in modalità SMS o in un dato GSM e viceversa, anche verso altre unità uguali.



In caso d'allarme o urgenza, grazie a suoi 8 segnali d'ingresso digitale TTL, può inviare istantaneamente un messaggio a una stazione di controllo, verso un cellulare o in qualsiasi altro luogo.

Intelligenza nascosta nel quotidiano

Dall'esperienza Motorola nel campo dei chip e dal know-how **Cedac Software** è nata una scheda rivolta al comparto industriale destinata a ridurre il time to market. Basata sull'architettura Motorola 32 bit Mcore, essa presenta un set completo di funzioni già implementate on board, che consentono di abbattere i tempi di programmazione. La scheda madre è implementata utilizzando i microcontroller MMC2107 o MMC2114, dotati di differenti capacità di memoria. I microcontroller sono basati sulla famiglia 32-bit Mcore M210 microRisc CPU capace di 31 Dhrystone 2.1 con una performance di 33 MHz. L'utilizzatore ha a propria disposizione fino a 24 pin I/O, una porta SPI, 2 external interrupt e 3 input analogici. Sono disponibili differenti frequenze (433, 868, 916 MHz e 2,4 GHz), due protocolli (collision detection e frequency hopping Tdma) e TCP/IP. La scheda è applicata in svariati campi, quali l'automotive e il ciclo del bianco (lavatrici e lavastoviglie). Oltre che nella domotica, dove il 'dialogo' fra elettrodomestici consente un notevole risparmio energetico, la tecnologia RF e il protocollo wireless trovano applicazione in altri campi, ad esempio nella gestione dei parchi auto delle compagnie che forniscono servizi di leasing. A tal fine, la scheda è disponibile anche in versione dotata di CAN control. Posizionata nel cruscotto della macchina è in grado di comunicare con la centralina di bordo trasmettendo i dati di manutenzione dell'auto e il suo stato generale; da qui può inoltre inviare le informazioni in RF verso le 'base station', centrali di raccolta dati da dove si può intervenire sulle automobili affittate.

Sistemi di telecontrollo satellitare

Intellisystem Technologies in collaborazione con Elsacom ha realizzato un sistema satellitare per il telecontrollo e la trasmissione di immagini mediante micro embedded Web server avvalendosi della tecnologia Globalstar. La soluzione proposta, basandosi sulla tecnologia packet PPP di Elsacom/ Globalstar, oltre ad acquisire e trasmettere immagini in formato Jpeg, permette la gestione software di attuatori per il telecontrollo dell'apparato in uso. Grazie alle potenzialità del micro Web server integrato nel sistema l'utente interagisce con il dispositivo remoto da controllare semplicemente attraverso interfacce Web personalizzabili. Le caratteristiche del sistema permettono di trasferire immagini in video-live supportando tutte le funzionalità della suite di protocolli TCP/IP, quali http, email ed ftp, e al contempo rappresenta un apparato di accesso al sistema che permette all'utente di interagire con esso in remoto agendo su opportuni attuatori telecontrollati via Web.
La facilità di programmazione della soluzione permette la gestione dell'invio dei fotogrammi, risolvendo molte delle problematiche inerenti il videocontrollo e il telecontrollo 'over IP' di macchinari o siti remoti. Il sistema trova ampie applicazioni nel telesoccorso, il monitoraggio ambientale, la protezione civile e militare.

Il controllo oltrepassa i limiti del filo

Vi sono 50 bilioni di dispositivi sul pianeta e ognuno dev'essere sorvegliato, monitorato e controllato attraverso una connessione. In molti casi basta stabilire un collegamento fisico, ma questo diventa un problema quando vi sono sistemi collocati in

Fieldbus & Networks

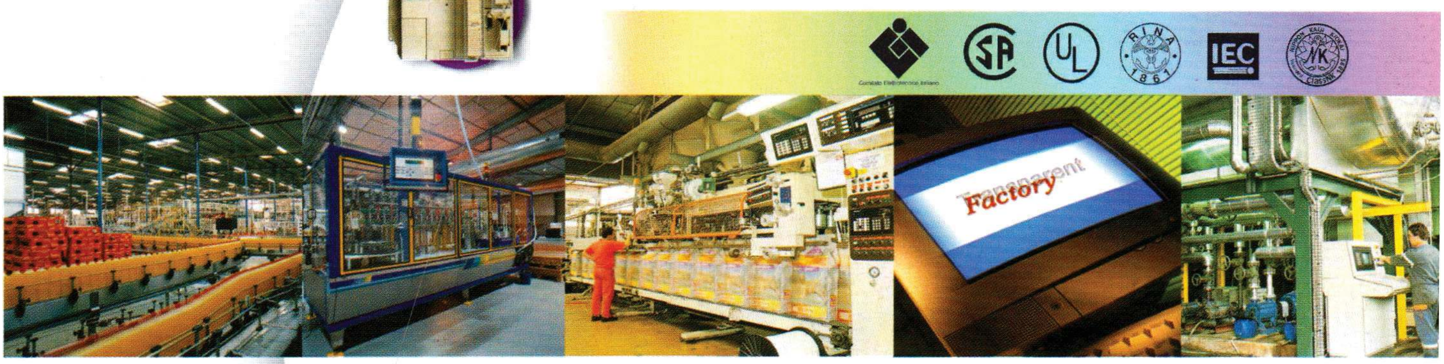
SETTEMBRE 2003 www.ilb2b.it

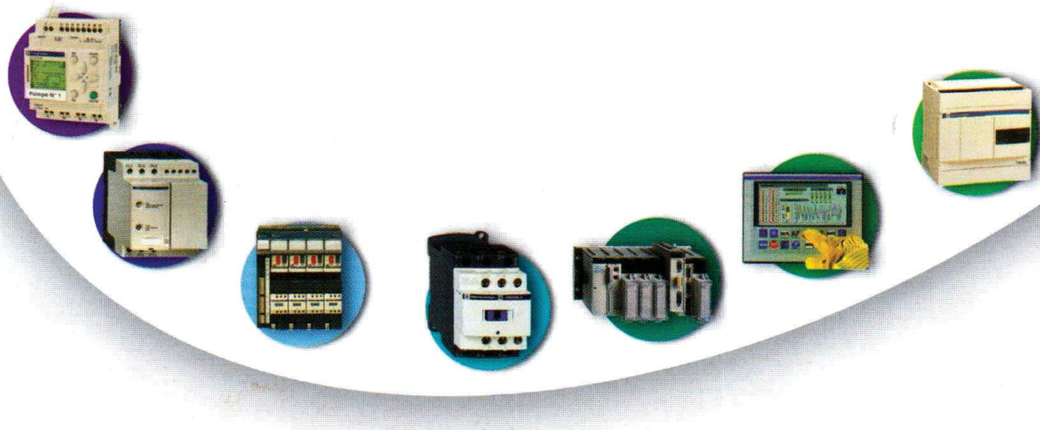
HARDWARE E SOFTWARE PER L'AUTOMAZIONE DISTRIBUITA

PRIMO PIANO
IL MONDO WIRELESS

TAVOLA ROTONDA
INTERNETWORKING: SÌ O NO?

DOSSIER
FIELDBUS A BORDO MACCHINA



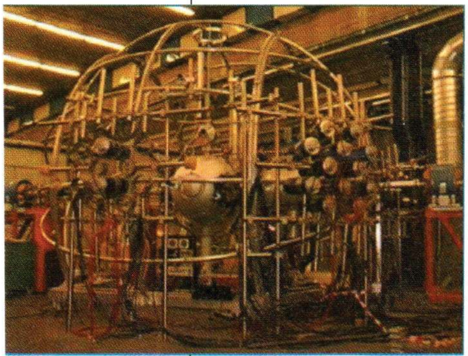


Supplemento ad Automazione Oggi n° 261 - Settembre 2003 - Spedizione in A.P. - 45% - art. 2 comma 20/B legge 662/96 - filiale di Milano - In caso di mancata consegna restituire all'editore che si impegna a pagare la relativa tassa presso il CMP di Roserio-Milano

vnu business publications italia

Profibus per la fisica nucleare

Intellisystem Technologies



La sala sperimentale chiamata 'Neutron Hall' dei Laboratori Nazionali del Sud di Catania

I fieldbus rappresentano una valida soluzione per il controllo remoto e la misura nei sistemi d'acquisizione usati nel campo della ricerca sperimentale

La continua evoluzione delle tecnologie basate su sistemi digitali ha fortemente modificato le tecniche e metodologie usate nei sistemi di controllo, specie negli esperimenti di fisica nucleare. In particolare, oggi la richiesta di processi distribuiti richiede sistemi intelligenti, dispositivi di controllo e sistemi di misura capaci di comunicare attraverso la rete. Tali sistemi vengono richiesti per ridurre le connessioni, il che si traduce in una semplificazione della gestione dei sistemi poiché diminuiscono le problematiche inerenti alla manutenzione.

I sistemi fieldbus rappresentano una valida soluzione per il controllo remoto e per la misura nei sistemi d'acquisizione normalmente applicati nella ricerca sperimentale oggetto della fisica nucleare. Questo campo di ricerca richiede apparati di controllo distribuiti capaci di lavorare in condizioni particolarmente difficili (campi elettromagnetici elevati, presenza di particelle radioattive, bassa temperatura, ecc...); al tempo stesso bisogna soddisfare i fondamentali requisiti di sicurezza, portatilità e semplicità richiesti. La presenza di un gran numero di dispositivi complica ulteriormente la progettazione e realizzazione di tali sistemi di controllo. Molti dispositivi devono essere controllati localmente, ciò richiede frequenti accessi alle sale sperimentali. Sfortunatamente, l'ambiente tipico degli esperimenti nucleari non è sicuro; di conseguenza, è molto difficile e pericoloso per un operatore umano recarsi all'interno di tali sale sperimentali per modificare i parametri dei sistemi quando l'esperimento è in esecuzione.

Misurare gli ioni pesanti

Il sistema sperimentale a cui ci si riferisce viene utilizzato per studiare le reazioni e interazioni degli ioni pesanti. La struttura è alloggiata in una sala sperimentale dei Laboratori Nazionali del Sud di Catania che prende il nome di 'Neutron Hall'.

Tale esperimento è gestito dal gruppo di ricerca Chic (Collaboration for Heavy Ion Collision). La struttura sperimentale si compone di una serie di camere a vuoto, con finestre d'osservazione, nelle quali vengono effettuate gli esperimenti di interferometria nucleare. Un fascio di ioni ad energia intermedia (10-50 MeV/A) viene deflesso nel vuoto e quindi convogliato all'interno della camera di reazione attraverso opportuni magneti. Il fascio urta il bersaglio nucleare; durante la collisione genera delle particelle subatomiche le cui caratteristiche vengono rilevate da vari rivelatori; i dati generati vengono acquisiti e successivamente analizzati dagli operatori.

Una prima applicazione del sistema Profibus si può trovare nel controllo remoto di un multidetector (costituito da un array bidimensionale di scintillatori allo Ioduro di Cesio) usato per rilevare particelle nucleari come protoni e/o ioni leggeri. L'apparato meccanico deve poter essere movimentato lungo i tre assi x, y e z in realtime, senza che il fascio prodotto dagli acceleratori di particelle venga interrotto. Un'altra esigenza stringente è costituita dalla precisione della meccanica e quindi dell'elettronica di controllo. Tramite l'utilizzo dei sistemi Profibus è stato implementato il controllo remoto dell'apparato meccanico integrato con un sistema di monitoraggio online della posizione di ogni singolo rivelatore. Tale applicazione ha messo in evidenza i vantaggi derivanti dall'introduzione dei sistemi Profibus in termini di portabilità e semplicità, grazie all'uso di un sistema a singolo bus di comunicazione.

Inoltre, si sono registrate buone performance in termini di velocità della rete durante la sofisticata interazione dinamica tra la console di comando e il dispositivo di controllo remoto.

Una seconda applicazione presenta l'implementazione di un apparato fieldbus su un sistema di controllo del vuoto utilizzato per una camera a vuoto speciale usata per sviluppare e testare sistemi di rivelazione di particelle operanti in condizioni di vuoto spinto. Il sistema nel suo complesso include due pompe da vuoto, due sensori per il vuoto e un PLC. Le due pompe lavorano in sequenza, poiché ognuna di esse opera in un determinato range di pressione. Il PLC viene impiegato per monitorare la pressione nella camera e contiene una logica capace di far commutare il funzionamento di ogni pompa. In questo caso è stato dimostrato come sia possibile combinare le caratteristiche dell'intelligenza decentrata, in accordo con la filosofia dei sistemi fieldbus, con stazioni di supervisione, sistemi di controllo (regolatori, PLC, ecc...), attuatori e trasduttori che condividono il bus e interagiscono in modi differenti. ■

Cristian Randieri

Born from the scientific research love, apply the modern technology for the well-being of all.

Research & Development

Intellisystem Technologies

Intellisystem Technologies S.r.l.
Via Augusto Murri, 1
96100 Siracusa - Italy

Tel: +39 (0)931-1756256 - +39 (0)2-87167549
Fax: +39 178 2286352 - +39 (0)931-1995470
Mobile +39 335-1880035

web: http://www.intellisystem.it

email: marketing@intellisystem.it - info@intellisystem.it