



# 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 scientifi-

che 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 cre-

sciuto 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.

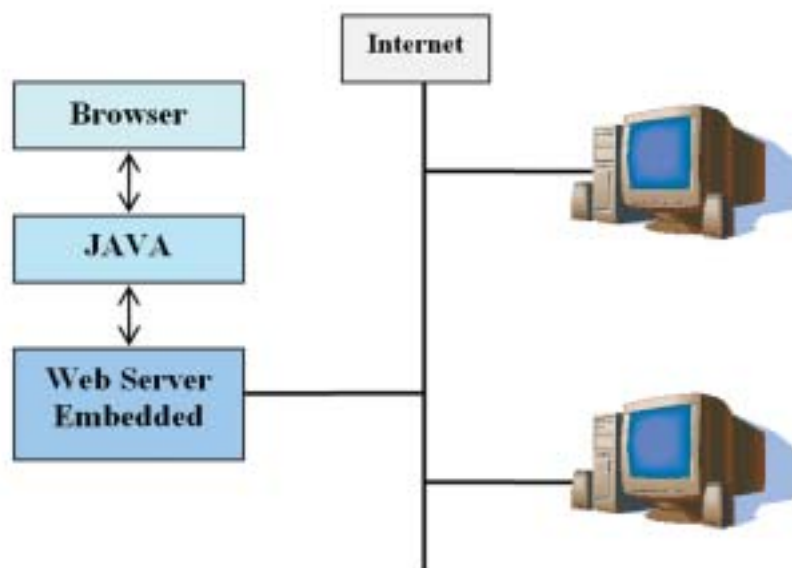
In realtà TCP e IP sono due protocolli utilizzati per interconnettere le reti. TCP sta per Transport Control

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).

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



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

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).

re. Aniché 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 pos-

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 tecnologia 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

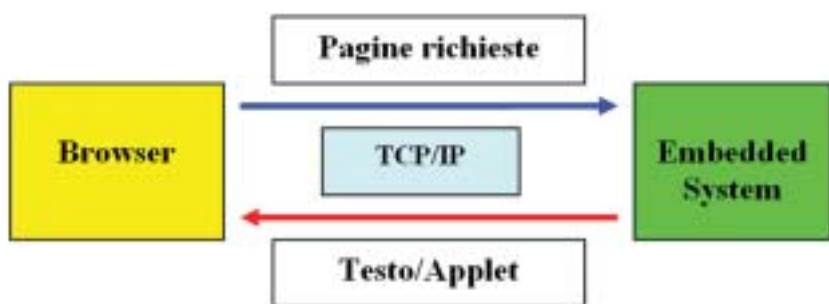


Figura 2 Esecuzione di Applet Java per eseguire operazioni di controllo o di monitoraggio di dispositivi remoti

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 softwa-

sono 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



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 conto 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,

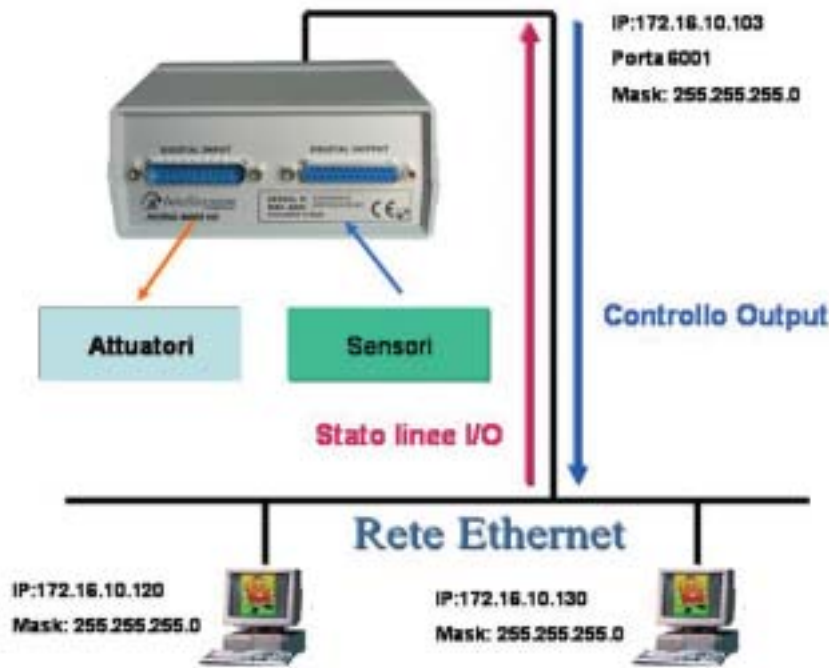


Figura 4 Scenario d'applicazione del dispositivo RECS 101

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 suo citato software di sviluppo [6]:

- Il file loader .class, che permette di fare il download 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 implementata in

accordo alle specifiche dettate dallo standard [7], fig. 5.

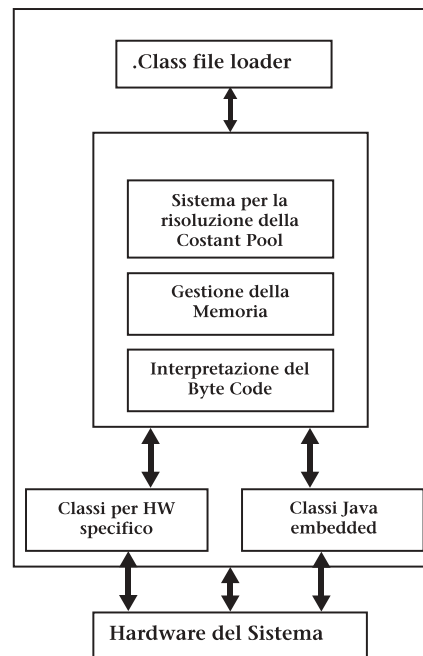


Figura 5 Architettura della JVM implementata

Il programma monitor che risiede all'interno del sistema embedded scarica il file .class che deve essere ese-

guito 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.

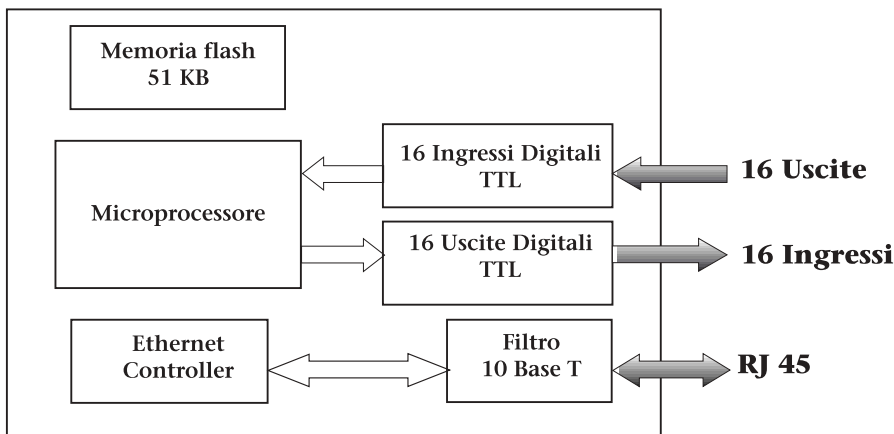
La JVM di per se stessa non comunica direttamente con l'hardware del sistema ma usa delle classi per fare ciò. Un fattore molto limitante di questi sistemi è dovuto alla scarsità della memoria che si ha a disposizione. Di conseguenza ciò porta a fare delle scelte su quali metodi e classi devono essere 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 immagi-

ni 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 pro-

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

grammare in Java.

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 rapidamente 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` rela-

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

**Tabella 2** Parametri di configurazione dell'Applet

tivo alla personalizzazione dell'Applet in cui si evidenzia il setup dei parametri di inizializzazione.

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 effet-



**Figura 9** Interfaccia GUI avanzata implementata in RECS 101

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

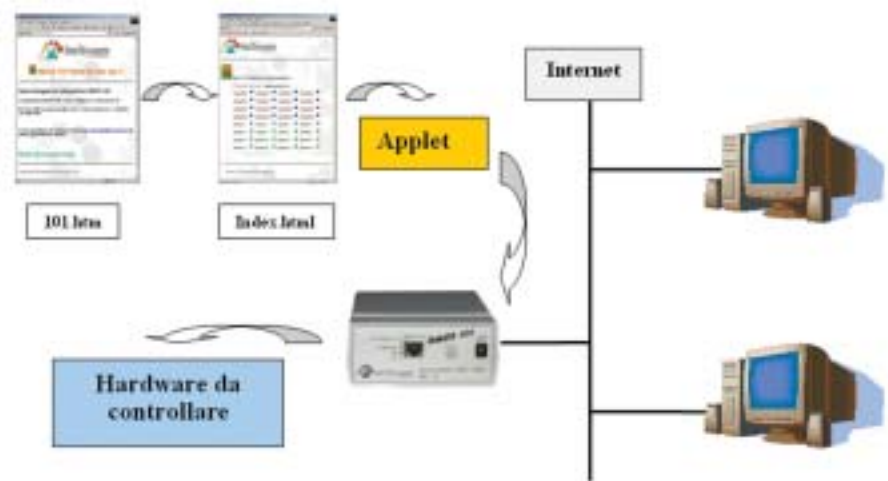
I files necessari per la personalizzazione dell'interfaccia utente di RECS 101 sono essenzialmente tre (a parte tutti quelli necessari per il contenuto grafico delle pagine web): `101.html`, `index.html`, `RECS.jar`.

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

*Electronic shop* 08

### 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] Apronix, "Bring Embedded System to the Internet", <http://www.aptronix.com>.
- [5] J. Gosling, B. Joy, G. Steele, "The Java Language 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>