



SDKTTS

Software Development Kit

16.05.2014
Rev. 0.4

TTS, STREAMTTS, TTSMAN, EMULTTS, sono marchi registrati della PHS srl.

INDICE:

| | |
|---|-----------|
| GENERALITA' | 3 |
| TTSMAN | 4 |
| Configuratore_Sinottico | 5 |
| Oggetto TTS | 9 |
| Oggetto Machine | 14 |
| Machine OPC client | 18 |
| Machine Portale RFID | 20 |
| Machine Pallettizzatore RFID | 21 |
| Oggetto Host Connection | 22 |
| Configuratore Generale | 24 |
| Configurazione tipo di master | 26 |
| Configurazione tipo di slave | 28 |
| Clonazione master | 31 |
| Clonazione slave | 31 |
| Generatore display formattato | 32 |
| TTSSIM – Simulatore logico di uno slave TTS | 35 |
| Display e Tastiera | 40 |
| Bottoni | 45 |
| Oggetti Hardware | 47 |
| RS 232C | 48 |
| RS 232C Porta Ausiliaria | 51 |
| Barcode | 52 |
| Badge | 54 |
| Relè | 55 |
| Buzzer | 55 |
| Calendario | 56 |
| Digital I/O | 57 |
| Global | 59 |
| UserSlave – plugin per il TTSSIM | 60 |
| RS232SIM – Simulatore fisico dei protocolli TTS per RS 232C | 62 |
| Configurazione | 63 |
| Attivazione e disattivazione | 64 |
| Utilizzo del bottone Setup Memory | 65 |
| UserRS232C – plugin per lo RS232SIM | 67 |
| VCSIM – Simulatore fisico dei canali virtuali VC in ethernet | 68 |
| Configurazione | 69 |
| Attivazione e disattivazione | 71 |
| Utilizzo del bottone Setup Memory | 72 |
| UserVC – plugin per il VCSIM | 75 |

GENERALITA'

Lo SDK del Sistema di Raccolta Dati TTS è composto da:

- **TTSMAN** primo strato software di una applicazione;
- **TTSSIM** simulatore logico di uno slave TTS;
- **RS232SIM** simulatore fisico dei protocolli TTS per la porta RS 232C;
- **VCSIM** simulatore fisico dei canali VC in ethernet.

TTSMAN

Il TTSMAN è il primo strato di un'applicazione di raccolta dati basata **principalmente** sul sistema TTS, ed è fornito:

- a) per le piattaforme windows, unix, ecc. in Java o in .NET (esso espone dei metodi e delle funzioni che permettono di gestire tutti gli eventi del campo);
- b) per il sistema operativo IBM I5/OS (o OS/400), sia in una soluzione completamente RPG, che RPG e Java (in questo caso la denominazione è TTSMANI5); le due soluzioni sono compatibili perché, con modalità diverse, richiamano due programmi RPG (TTSUSR e TTSSPL).

Il TTSMAN è composto da due moduli:

- a) uno operativo per la ricezione, trasmissione e decodifica delle stringhe TTS che permette di sviluppare l'applicazione, esponendo metodi, ecc.;
- b) l'altro per la configurazione di tutti i dispositivi TTS componenti il sistema TTS che si vuole realizzare.

Per il primo modulo si rimanda alle documentazioni:

- a) **JTTSMAN.pdf** per le piattaforme Java e .NET;
- b) **TTSMANAS.pdf** per quella IBM.

Il secondo modulo, configuratore, è descritto di seguito.

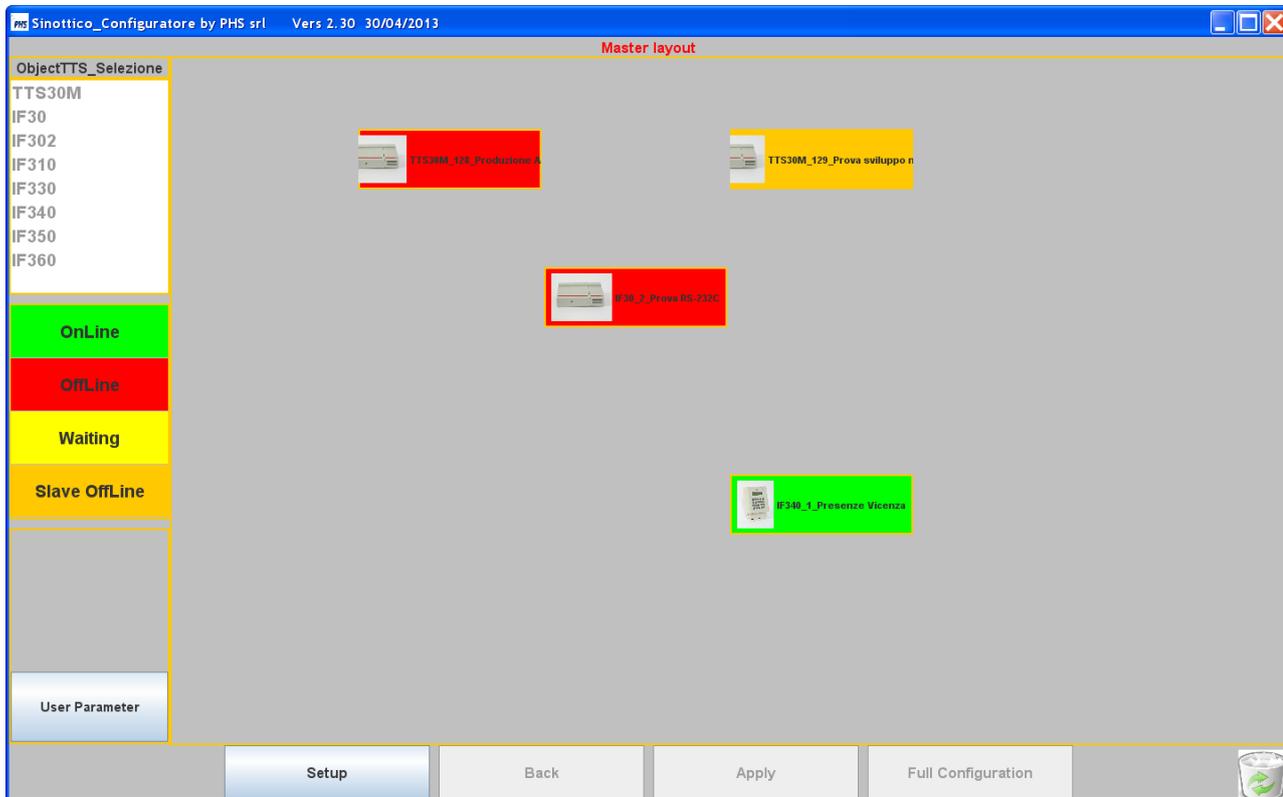
Configuratore_Sinottico

Lo scopo del Sinottico_Configuratore è di presentare lo stato dei dispositivi e di permettere una configurazione intuitiva e semplice che può essere affinata, per i dispositivi TTS, tramite il **configuratore generale** illustrato nel capitolo successivo.

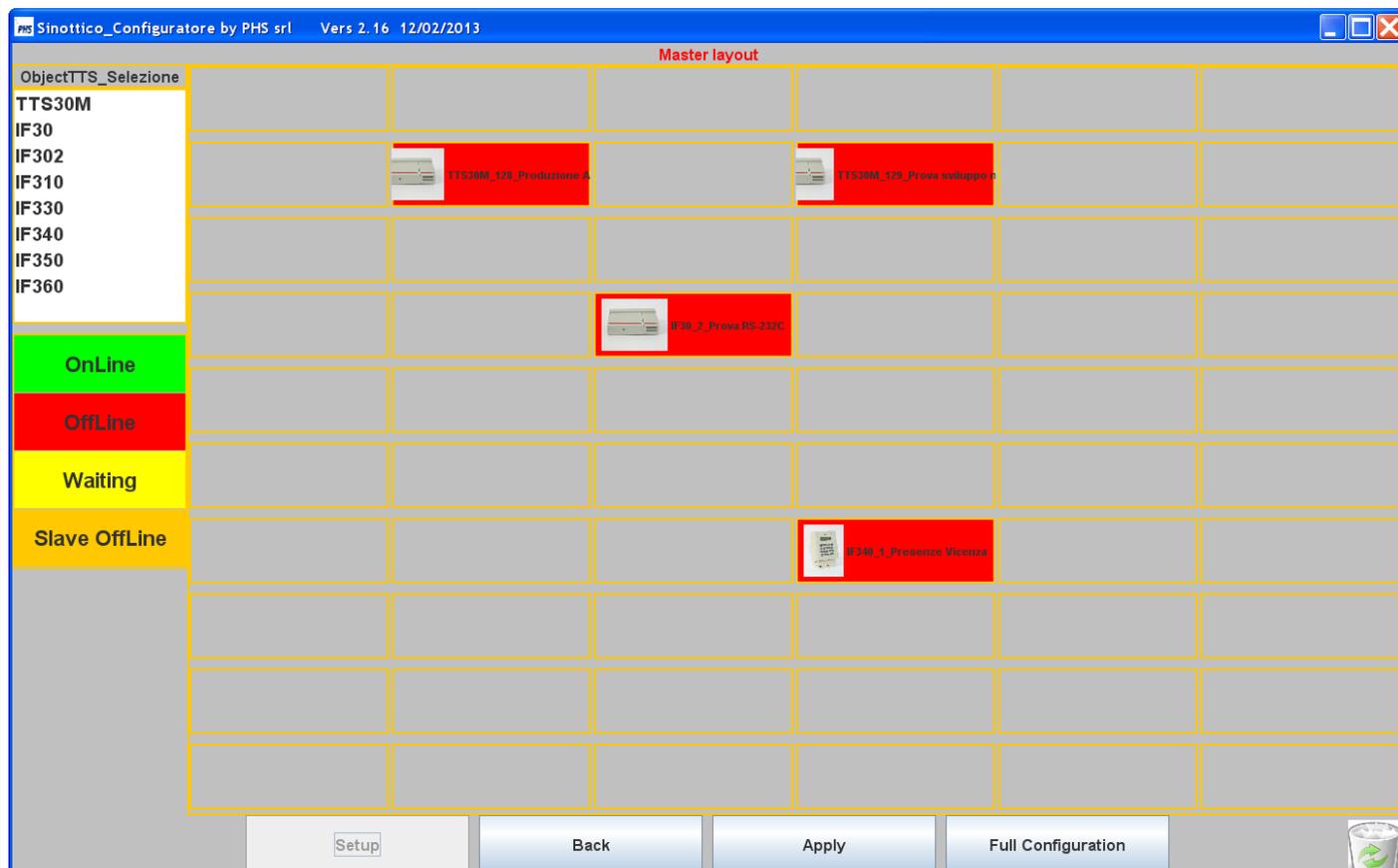
Nella figura seguente è riportato il pannello nello stato di sinottico e di visualizzazione, ogni oggetto assume il colore relativo al proprio stato.



Il bottone **User Parameter** permette di lanciare un' applicazione costruita esternamente, tramite il metodo **displayUser()** della classe StrTTS (vedi il documento JTTSMAN.pdf).

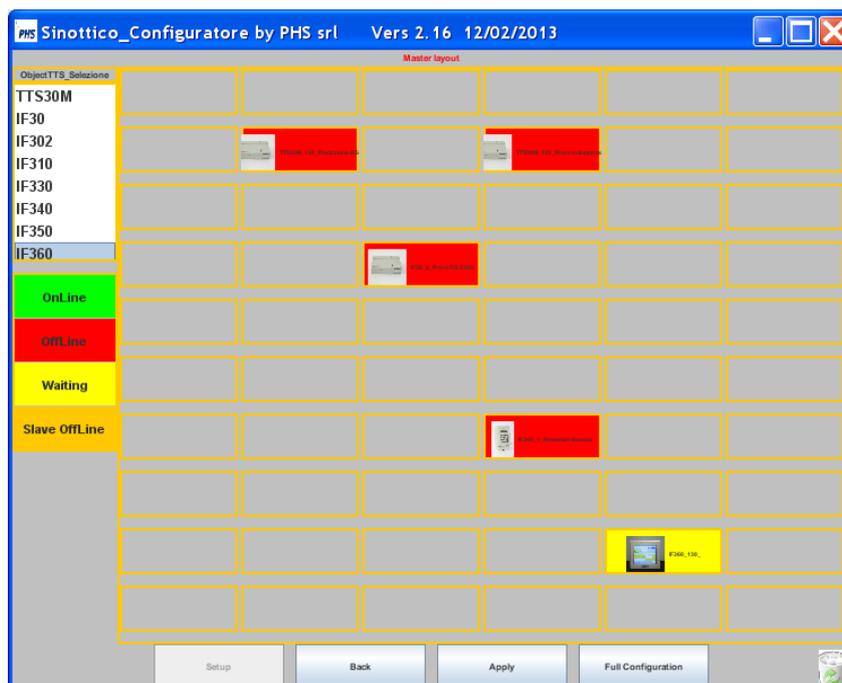


Il bottone **Setup** cambia lo stato del pannello portandolo in quello di configurazione.

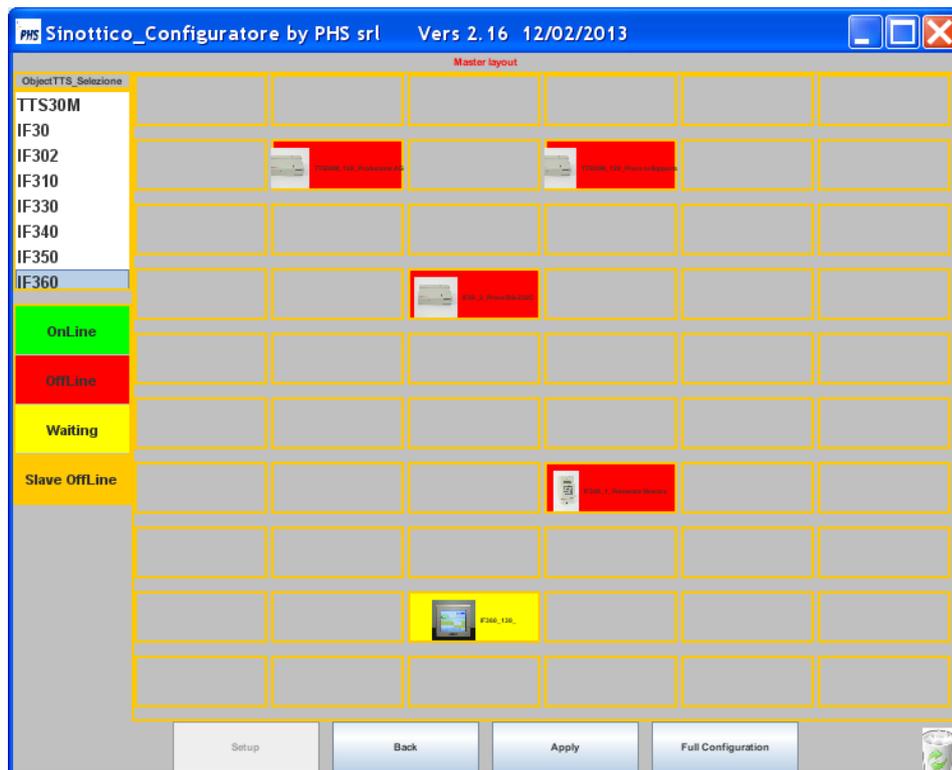


In questa modalità è attivo il drag & drop che permette:

- 1) di scegliere e posizionare un nuovo oggetto dal pannello **ObjectTTS_Selezione**;

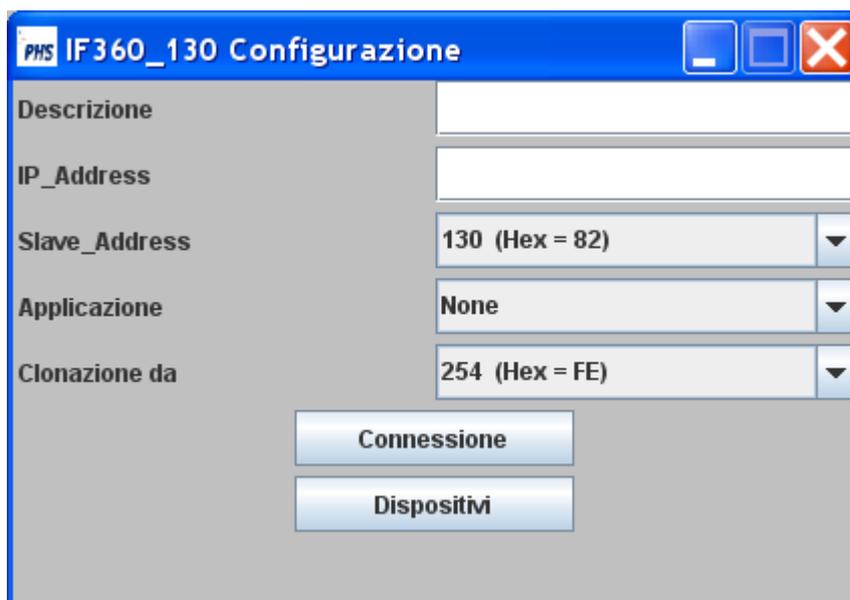


2) di spostare un oggetto;



3) di cancellare un oggetto spostandolo nel cestino.

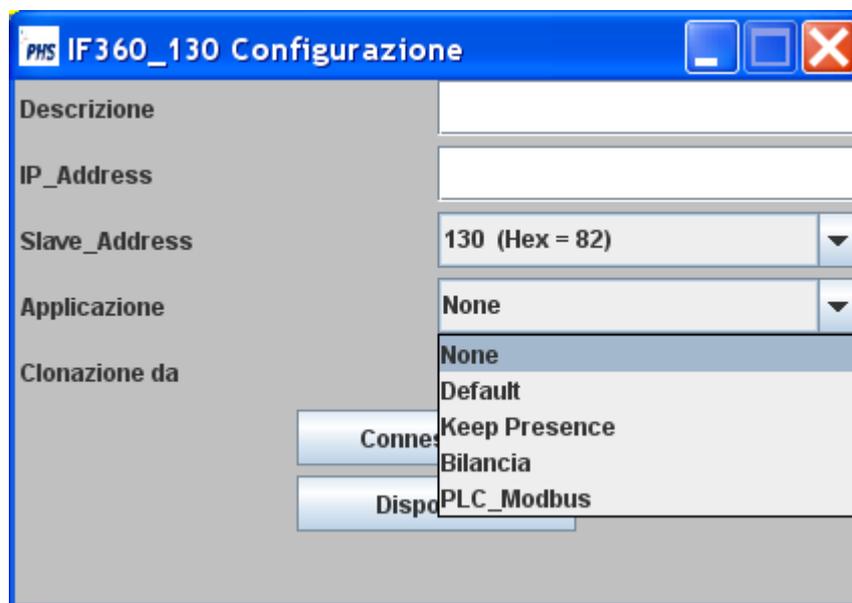
Il **doppio click** su un'oggetto permette di di configurarlo.



Oggetto TTS

Nello stato di configurazione e' possibile per un oggetto TTS:

- 1)inserire una descrizione;
- 2)inserire un IP;
- 3)scegliere l'indirizzo fisico corrispondente al dispositivo previsto;
- 4)clonare i restanti parametri da un oggetto dello stesso tipo già configurato;
- 5)selezionare una configurazione precablata tramite il selettore **Applicazione**.



Le strutture precablate saranno rese disponibili progressivamente,ma ognuno può svilupparne delle proprie tramite un plugin di classe:

PlugAppObjectTTS.

Lo sviluppo del plugin prevede,in una struttura predefinita,la compilazione di tabelle;
vale l'esempio seguente:

```
public class TableConf
{
    // *****
    //***** Definizione Parametri Applicazione***** *
    // *****
    public static int numApp = 2;
    private static int logicUnit;

    public static String[] nameAPP = new String[numApp];
    public static String parmApp[][] = new String[numApp][ParmDefault.numHwo][ParmDefault.numParm];
    public static StructHwo[][] hwoStruct = new StructHwo[numApp][ParmDefault.numHwo];
    static { for (int i = 0;i<numApp;i++){ for (int j = 0;j<ParmDefault.numHwo;j++){ hwoStruct[i][j] = new StructHwo();} } }
```

```
static
{
    nameAPP[0] = "Bilancia";
    nameAPP[1] = "PLC_Modbus";

    // *****
    //***** Applicazione Bilancia ***** *
    // *****
    logicUnit = 0;          //RS-232-C

    //Modalità di configurazione
    hwoStruct[0][logicUnit].existHwo = "0";
    hwoStruct[0][logicUnit].confHwo = "1";
    hwoStruct[0][logicUnit].lockHwo = "0";
    hwoStruct[0][logicUnit].setupHwo = "0";

    //Parametri canale seriale
    parmApp[0][logicUnit]['A'-'A'] = "00";      //A_Protocollo
    parmApp[0][logicUnit]['B'-'A'] = "46";      //B_(Baud rate = 9600,no parity,8 bit)
    parmApp[0][logicUnit]['C'-'A'] = "1";       //C_Autoblocco
    parmApp[0][logicUnit]['K'-'A'] = "1";       //K_ricezione continua

    // *****
    //***** Applicazione PLC Modbus ***** *
    // *****
    logicUnit = 0;          //RS-232-C

    //Modalità di configurazione
    hwoStruct[1][logicUnit].existHwo = "0";
    hwoStruct[1][logicUnit].confHwo = "1";
    hwoStruct[1][logicUnit].lockHwo = "0";
    hwoStruct[1][logicUnit].setupHwo = "0";

    //Parametri canale seriale
    parmApp[1][logicUnit]['A'-'A'] = "05";      //A_Protocollo
    parmApp[1][logicUnit]['B'-'A'] = "46";      //B_(Baud rate = 9600,no parity,8 bit)
    parmApp[1][logicUnit]['J'-'A'] = "140";     //J_CRC polinomiale
}
}
```

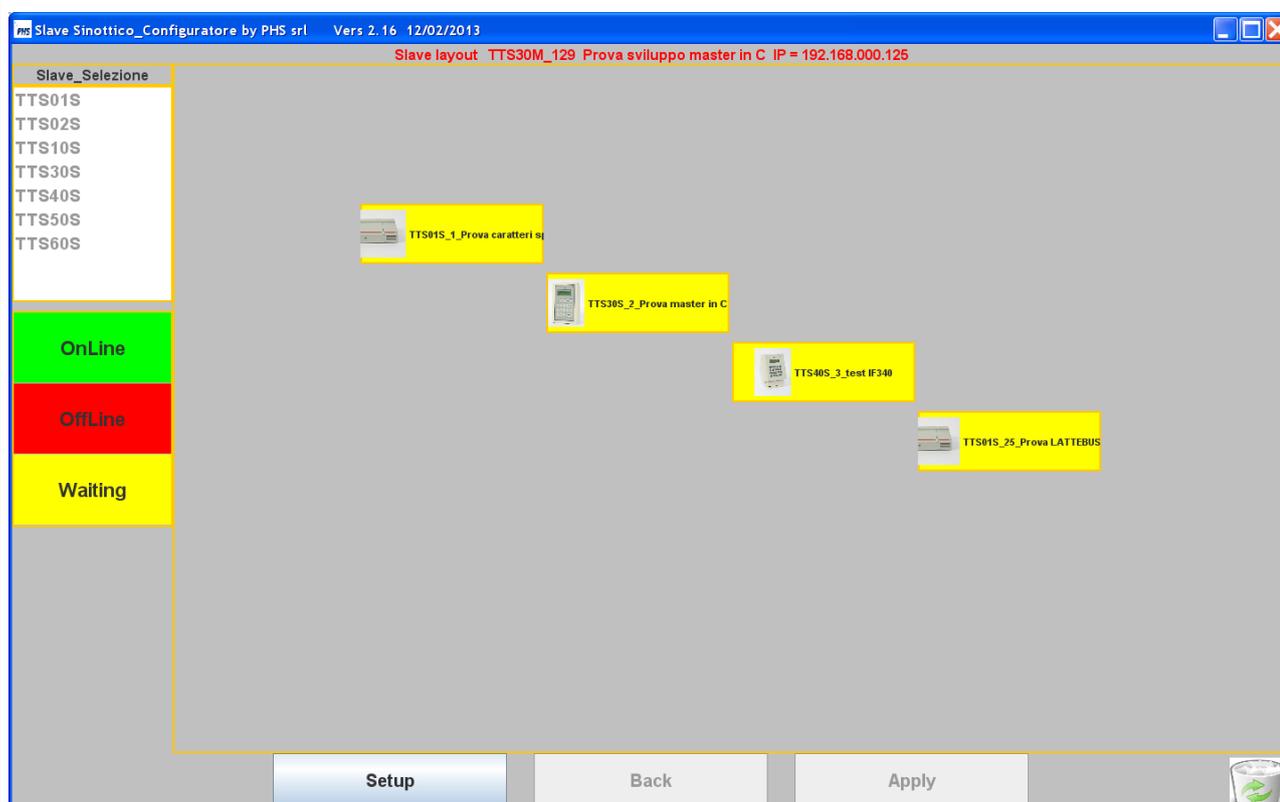
Un'applicazione prevede un nome (*nameAPP*) e i relativi parametri (*hwoStruct,parmApp*).

I bottoni **Connessione** e **Dispositivi** permettono di attivare i pannelli di configurazione descritti nel capitolo **configuratore generale**.

Sul pannello Sinottico_Configuratore i restanti bottoni sono:

- 1)**Backup**: permette di annullare tutte le operazioni effettuate;
- 2)**Apply**: applica quanto configurato e ripristina lo stato di visualizzazione;
- 3)**Full Configurator**: attiva il **configuratore generale**.

Nello stato di configurazione, per un master TTS30M, si può configurare la sua struttura di slave.



Il **doppio click** su un'oggetto TTS permette, nello stato sinottico di avere le informazioni previste per esso.

Se l'oggetto **non è un TTS30M**, otteniamo:



Il bottone **Altri dati** permette di richiamare un pannello aggiuntivo costruito tramite il metodo **displayParm()** della classe StrTTS (vedi il documento JTTSMAN.pdf).

| Macchina | Tempo ciclo | Numero pezzi | Scarti |
|----------|-------------|--------------|--------|
| Tornio | 60 sec | 1000 | 10 |
| Fresa | 90 sec | 2000 | 5 |
| Pressa | 2 sec | 10000 | 50 |

Next Previous

Se l'oggetto è un **TTS30M**, otteniamo:

TTS30M_129 Configurazione

Descrizione: Prova sviluppo master in C

IP_Address: 192.168.000.125

Master_Address: 129 (Hex = 81)

Slave_Struttura

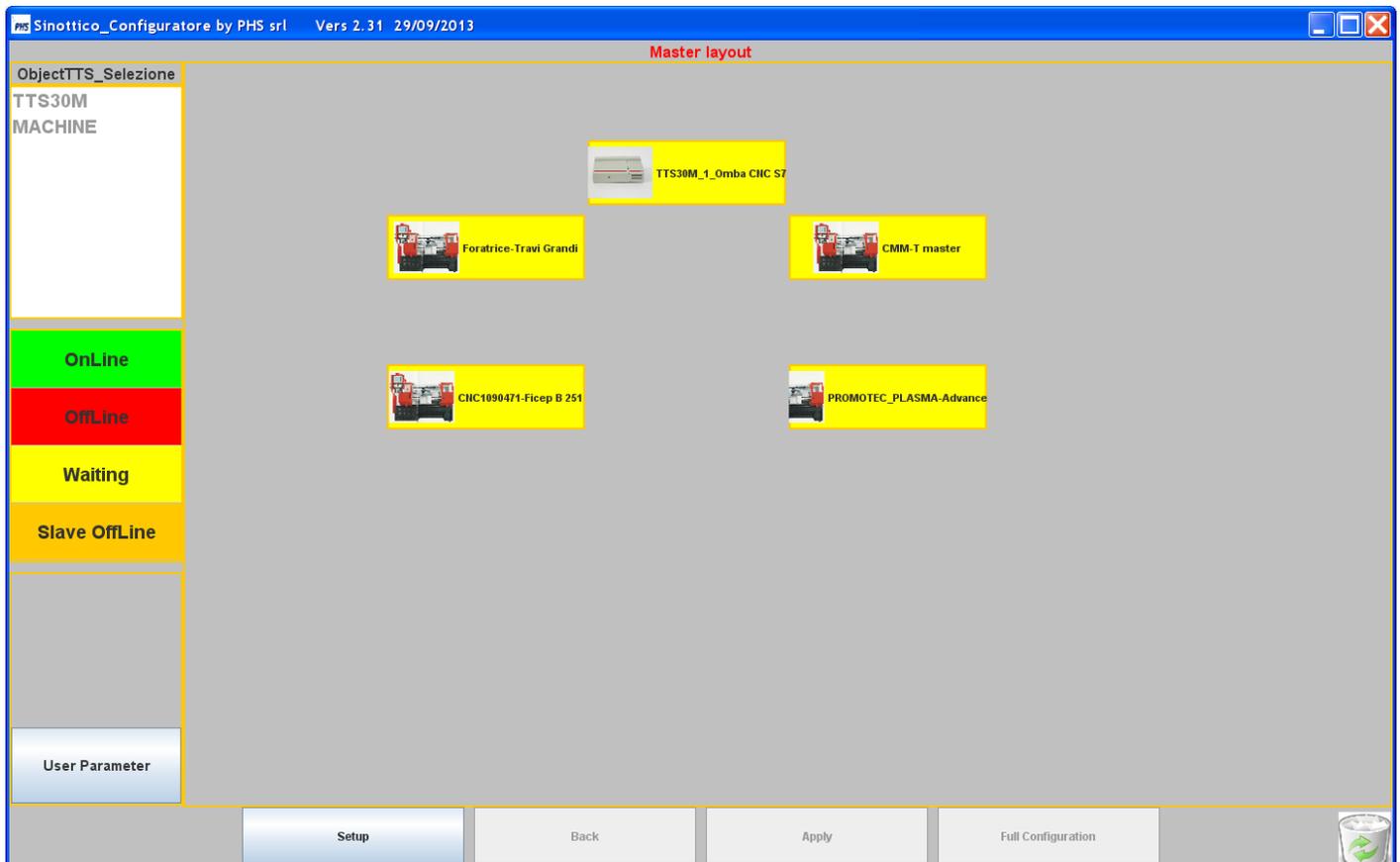
Altri dati

Il bottone **Slave_Struttura** permette di richiamare il pannello riportante la struttura degli slave associati al master.



Oggetto Machine

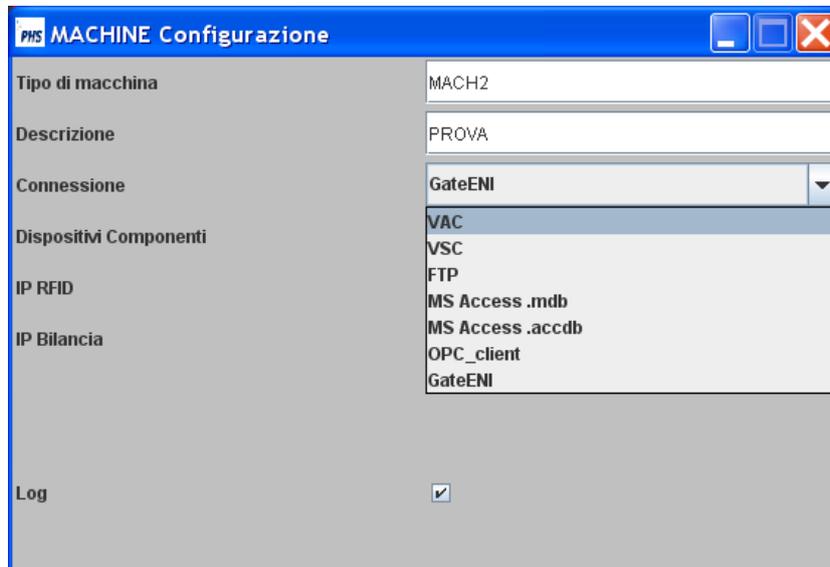
Se vale l'assegnazione `listType.machine = true`, è abilitato l'utilizzo dell'oggetto di classe Machine.



L'oggetto Machine permette di inserire, nel sistema di raccolta dati TTS, una macchina generica dotata di una connessione (porta di comunicazione ethernet e protocollo).

Nella configurazione il parametro **“Tipo di macchina”** è l’**identificativo dell’oggetto**.

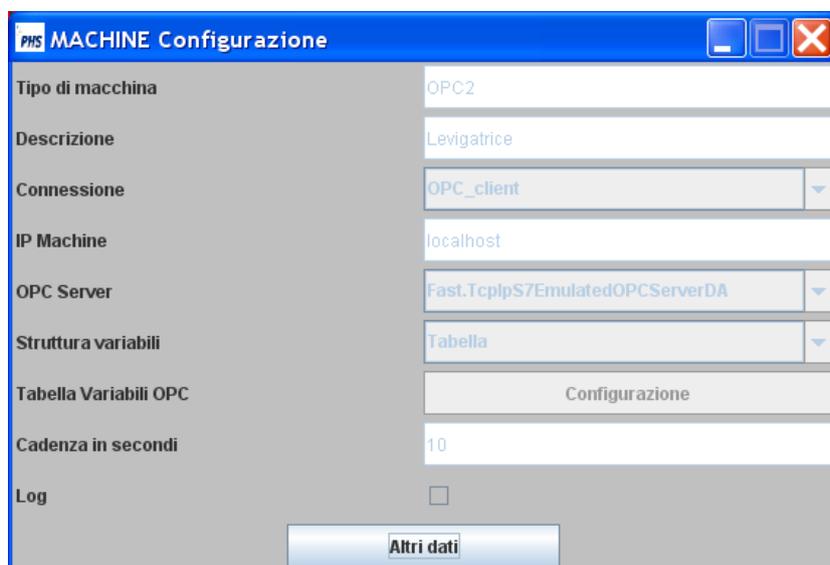
Per l’oggetto è possibile scegliere il tipo di connessione tra quelli previsti: VAC,VSC,FTP, MS Access,OPC Client,ecc.;o definirne uno custom (es. GateENI) tramite il plugin **PlugMachine**.



Per ogni connessione,secondo le caratteristiche specifiche,è possibile inviare e ricevere dati dalla macchina collegata.

Le connessioni VAC e VSC,prevedono il collegamento delle macchine,con i canali virtuali del master TTS30M.

Il bottone **Altri dati** permette di richiamare un pannello aggiuntivo costruito tramite il metodo **displayMach()** della classe StrTTS (vedi il documento JTTSMAN.pdf).



Il plugin **PlugMachine**, in una struttura definita, rende disponibili delle classi secondo lo schema seguente:

1) **PlugMachine** <SM,PA>

- Campi

- String **nameMach** : nome che apparirà nel selettore (es. GateENI);
- Image **img** : logo dell'oggetto;
- boolean **enaBut** : flag di abilitazione;
- StructMachine **structMach** : struttura parametri.

- Metodi:

- **initialize**, è il metodo che importa una classe Machine PA;
- **getParm**, è il metodo che restituisce una struttura parametri di classe SM;
- **conField**, è il metodo che può aggiungere fino a sei oggetti al pannello di configurazione, oltre i primi tre.
- **manEnaProt**, è il metodo che permette di gestire l'abilitazione del pannello.



2) **ManMachine** <SM,LO,PA,SO>

- Campi

- long **timeMach** : determina la cadenza (in msec) dell'attivazione del metodo **getMachine**.

- Metodi:

- **initialize**, è il metodo che importa una classe Machine PA, ed una classe Socket SO;
- **getMachine**, è il metodo che gestisce la ricezione dalla macchina e restituisce un' **ArrayList di ArrayList**, i cui elementi sono di classe **FieldColumn**.
- **putMachine**, è il metodo che gestisce la trasmissione alla macchina.
- **getStatus**, è il metodo che restituisce lo stato della connessione.
- **conSock**, è il metodo che segnala la connessione/sconnessione del socket.
- **killer**, è il metodo che permette di annullare tutti gli oggetti, timer, thread, ecc.

3) StructMachine, è la classe che definisce la struttura parametri inizializzata tramite una classe SM.

4) FieldColumn, è la classe che definisce la struttura di un record composto dai campi:

- String **row** :identifica la riga a cui associare il record;
- String **nameField** :identificativo del record;
- Object **objField** :è il valore del record.

Machine OPC client

L'oggetto Machine prevede, come connessione, l'**OPC client**, per esso vale la configurazione riportata.

The screenshot shows a configuration window titled 'MACHINE Configurazione'. The fields are as follows:

| | |
|---------------------|---------------------------------|
| Tipo di macchina | OPC1 |
| Descrizione | Pressa |
| Connessione | OPC_client |
| IP Machine | localhost |
| OPC Server | Fast.TcplpS7EmulatedOPCServerDA |
| Struttura variabili | File Esterno |
| File | OPCTcplpS7 |
| Cadenza in secondi | 10 |
| Log | <input type="checkbox"/> |

Per la struttura delle variabili è possibile utilizzare il file xml standard del server OPC, o costruire una propria tabella.

The screenshot shows the same configuration window, but with the 'Struttura variabili' dropdown set to 'Tabella'. A new field, 'Tabella Variabili OPC', has appeared with a button labeled 'Configurazione'.

| | |
|-----------------------|---------------------------------|
| Tipo di macchina | OPC1 |
| Descrizione | Pressa |
| Connessione | OPC_client |
| IP Machine | localhost |
| OPC Server | Fast.TcplpS7EmulatedOPCServerDA |
| Struttura variabili | Tabella |
| Tabella Variabili OPC | Configurazione |
| Cadenza in secondi | 10 |
| Log | <input type="checkbox"/> |

Tramite il bottone Configurazione si richiama la tabella delle variabili, che può essere opportunamente editata.

| Riga | Nome | Variant Type | Accesso | Commento |
|------|------------------------------|---------------|------------|---------------------------------|
| 1 | Plc Trenino.VAR.020_VEL_S... | VT_I2 (short) | Read_Write | Risultante velocità SEMI + A... |
| 2 | Plc Trenino.VAR.040_VEL_S... | VT_R4 (float) | Read_Write | Risultante velocità SEMI + A... |

Inoltre è previsto il plugin **PlugOPC** che, in una struttura definita, rende disponibili una classe secondo lo schema seguente:

1) PlugOPC

- Campo

- String **nameServer** : nome del Server OPC che apparirà nel selettore;

- Metodo:

- **getGroup**, il metodo restituisce un ArrayList, composto da tre ArrayList:

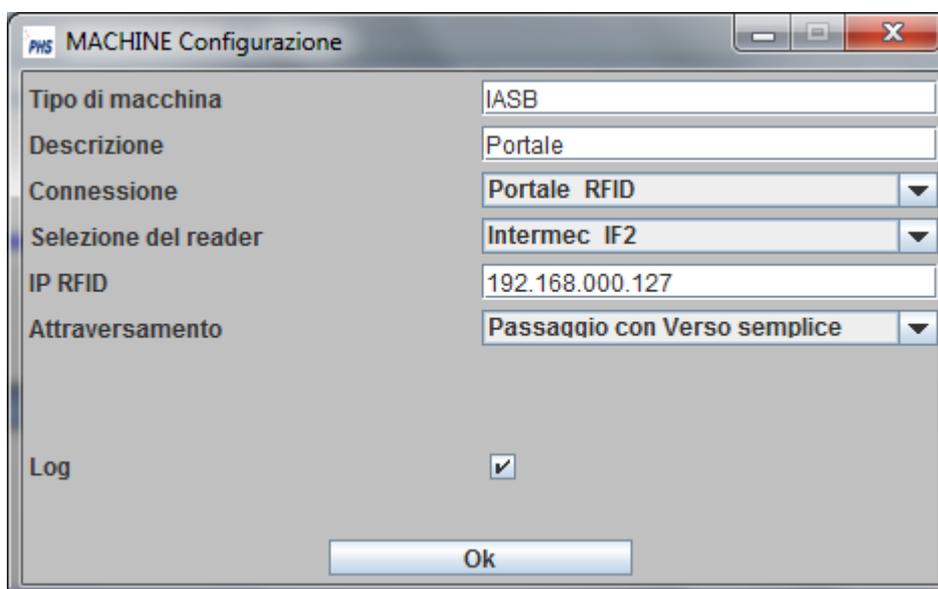
- lista delle variabili di input;

- lista delle variabili di output;

- lista dei Variant Type per le variabili di output.

Machine Portale RFID

L'oggetto Machine prevede, come connessione, il **Portale RFID**, per esso vale la configurazione riportata.

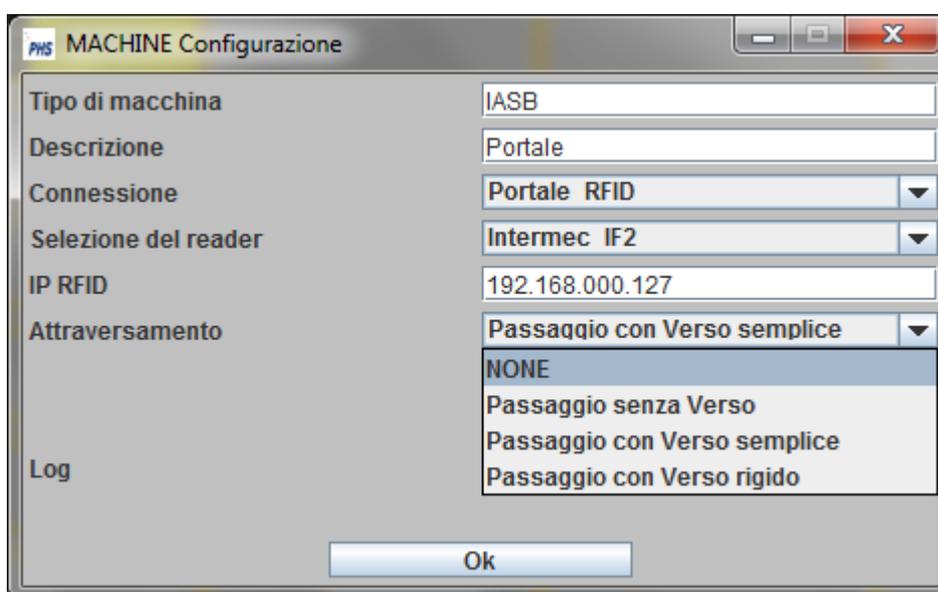


The screenshot shows a dialog box titled "MACHINE Configurazione" with the following fields and values:

| | |
|----------------------|-------------------------------------|
| Tipo di macchina | IASB |
| Descrizione | Portale |
| Connessione | Portale RFID |
| Selezione del reader | Intermec IF2 |
| IP RFID | 192.168.000.127 |
| Attraversamento | Passaggio con Verso semplice |
| Log | <input checked="" type="checkbox"/> |

An "Ok" button is located at the bottom center of the dialog.

Oltre al Reader è possibile scegliere il tipo di **Attraversamento**.



The screenshot shows the same dialog box as above, but with the "Attraversamento" dropdown menu open, displaying the following options:

- NONE
- Passaggio senza Verso
- Passaggio con Verso semplice
- Passaggio con Verso rigido

The "Ok" button is visible at the bottom.

Il **Passaggio con Verso rigido** prevede che l'attraversamento avvenga linearmente e senza soluzione di continuità; è trasmesso il TAG di ogni oggetto che attraversa il varco ed il verso.

Il **Passaggio senza Verso** prevede la presenza di un sensore come input, invece due negli altri casi.

Machine Pallettizzatore RFID

L'oggetto Machine prevede, come connessione, il **Pallettizzatore RFID**, per esso vale la configurazione riportata.

| | |
|--------------------------|-------------------------------------|
| Tipo di macchina | PA1 |
| Descrizione | Fascia Pallet |
| Connessione | Pallettizzatore RFID |
| Selezione del reader | Intermec IF2 |
| IP RFID | 192.168.000.130 |
| Bilancia Pallettizzatore | Dini Argeo |
| IP Bilancia | 192.168.000.131 |
| Log | <input checked="" type="checkbox"/> |

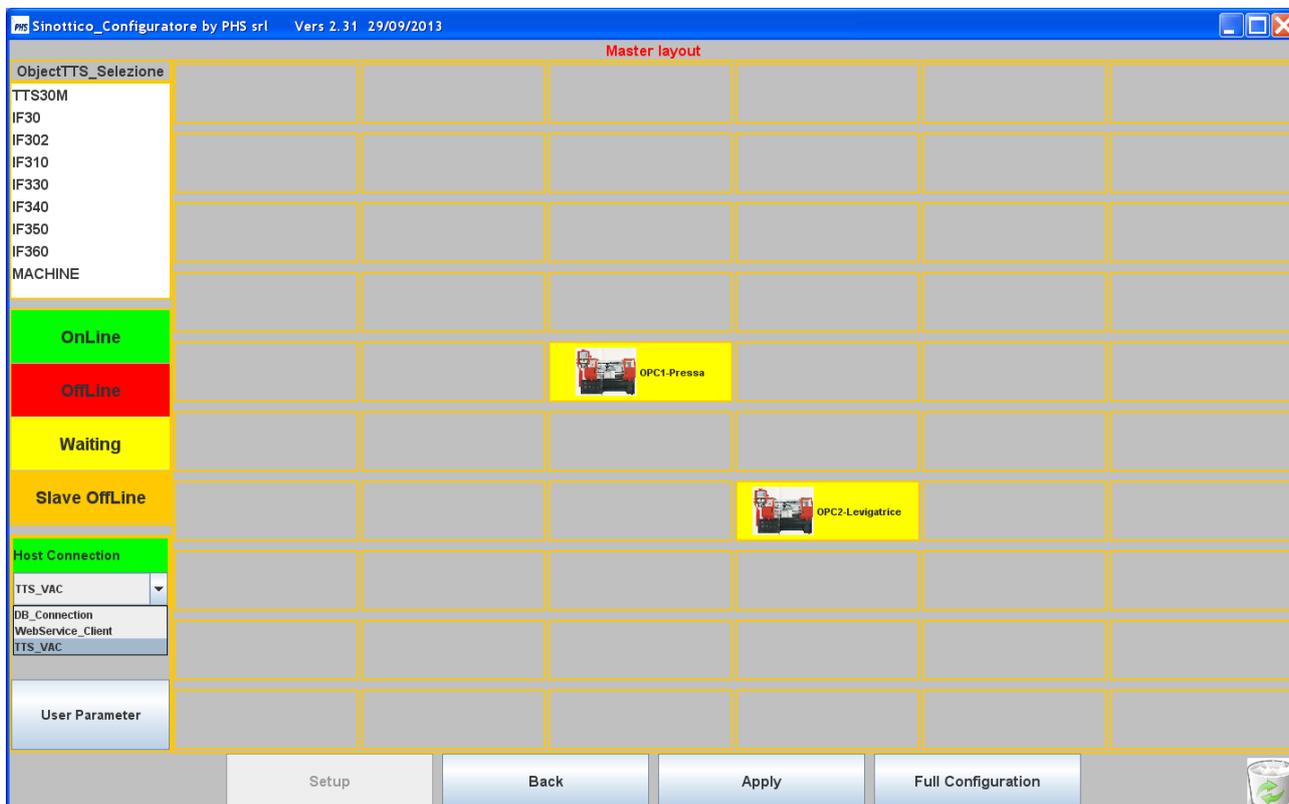
Ok

Il pallet è gestito sia in fase di montaggio, che smontaggio; è trasmesso il TAG ed il peso del singolo collo.

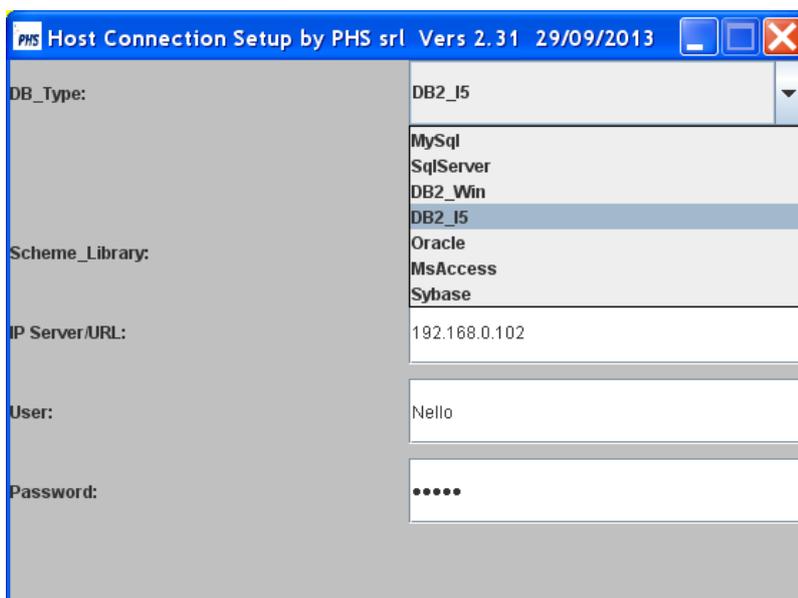
Oggetto Host Connection

Se vale l'assegnazione `listType.hostCon = true`, è abilitata la configurazione del collegamento ad un Host Server (DB, WebService, TTS VAC, ecc.).

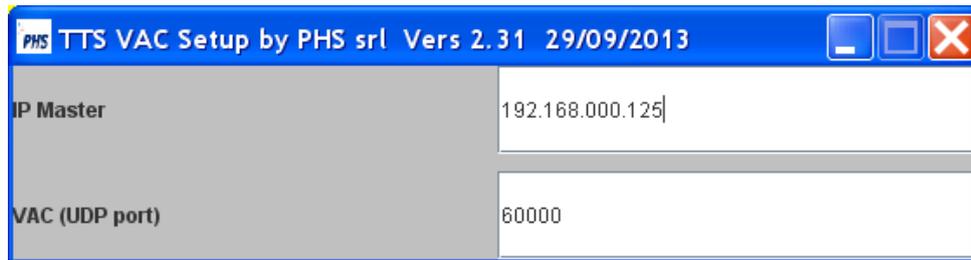
Il colore dell'oggetto indica lo stato della connessione verso l'Host.



Per un **Server DB** abbiamo:



Per il canale VAC di un master TTS30M abbiamo:



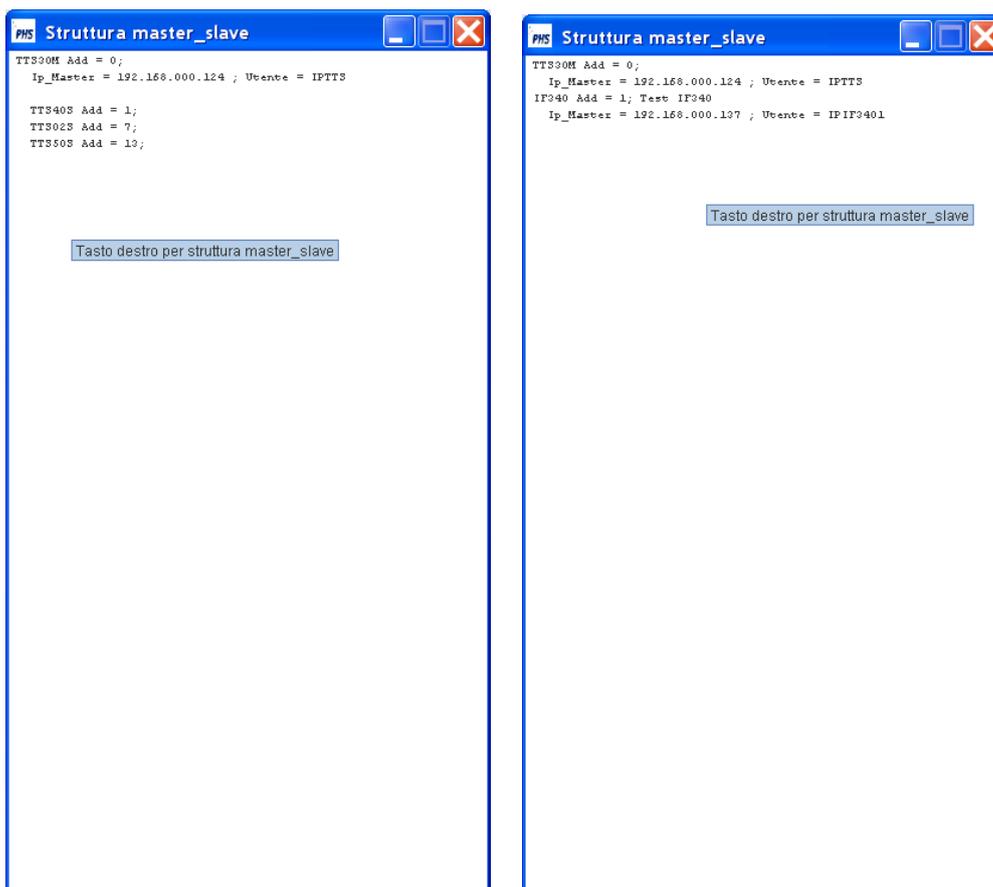
| PHS TTS VAC Setup by PHS srl Vers 2.31 29/09/2013 | |
|---|-----------------|
| IP Master | 192.168.000.125 |
| VAC (UDP port) | 60000 |

Configuratore Generale

Per utilizzare il configuratore bisogna far riferimento al documento **STREAM.pdf**, che descrive il formato (inbound/outbound) delle stringhe TTS.

L'attivazione del configuratore presenta due pannelli affiancati.

Il primo commuta stato tramite il tasto destro, in uno presenta la struttura corrente, nell'altro l'elenco delle strutture configurate.



Il secondo dispone delle funzioni necessarie per la configurazione delle strutture hardware (master e slave) volute.

| Label | Value 1 | Value 2 |
|-------------------------------|---------|---------|
| Configurazione tipo di master | 0 | TTS30M |
| Configurazione tipo di slave | 1 | NONE |
| Clonazione master | 0 | 0 |
| Clonazione slave | 1 | 1 |
| Generatore display formattato | 0 | TTS60S |

Save

Le funzioni sono:

- a) Configurazione tipo di master;
- b) Configurazione tipo di slave;
- c) Clonazione master;
- d) Clonazione slave;
- e) Generatore display formattato

Configurazione tipo di master

Preventivamente è necessario scegliere l'indice e il tipo di master,

Configuratore Sistema TTS by PHS srl Vers 1.23 11/11/2011

Configurazione tipo di master: 0 (dropdown), TTS30M (dropdown)

Configurazione tipo di slave: 0 (dropdown), NONE (dropdown)

Clonazione master: 0 (dropdown)

Clonazione slave: 1 (dropdown)

Generatore display formattato: 0 (dropdown), TTS60S (dropdown)

Save

Configuratore Sistema TTS by PHS srl Vers 1.23 11/11/2011

Configurazione tipo di master: 0 (dropdown), TTS30M (dropdown)

Configurazione tipo di slave: 1 (dropdown), IF30, IF302, IF303, IF310, IF330, IF340, IF350 (list)

Clonazione master: 0 (dropdown)

Clonazione slave: 1 (dropdown)

Generatore display formattato: 0 (dropdown), TTS60S (dropdown)

Save

dopo, utilizzando il bottone **Configurazione tipo di master**, si ottiene un pannello del tipo:

Configuratore MASTER by PHS srl

Master TTS30M Add = 0

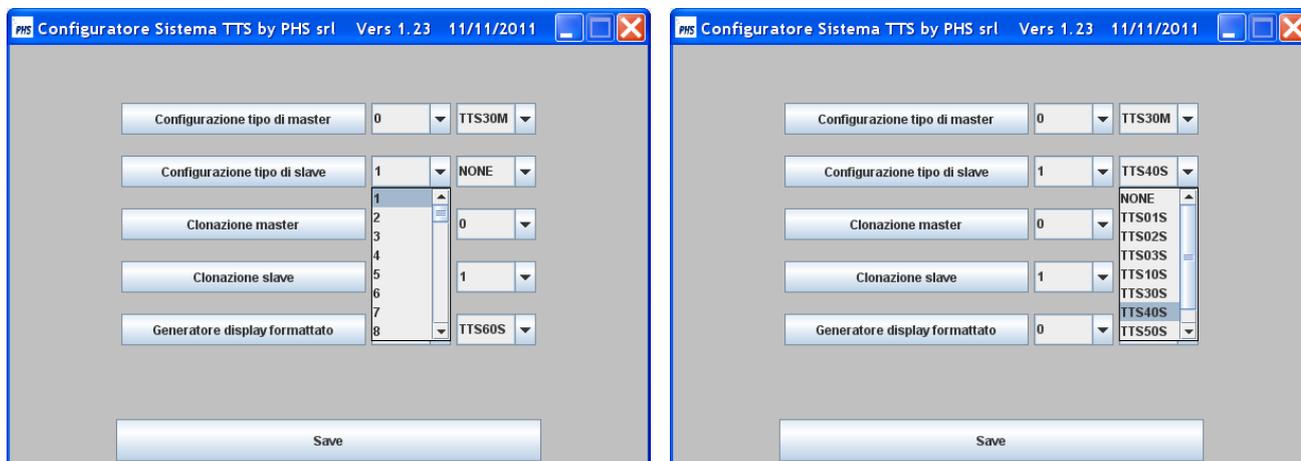
| STREAMTTS | Valore | Descrizione |
|--|--------------------------|--|
| C-IPmaster (xxx.yyy.www.zzz) | | Indirizzo IP del master |
| J-KEEP_ALIVE (step di 2 minuti) | 1 | Timeout di inattività TCP |
| E-Modalità Client/Server | Server | Tipo di collegamento all'Host |
| F-IPHost (xxx.yyy.www.zzz) | 000.000.000.000 | Indirizzo IP dell'Host |
| O-IP_Backup (xxx.yyy.www.zzz) | 000.000.000.000 | Indirizzo IP dell'Host di backup |
| I-Default Gateway (xxx.yyy.www.zzz) | 000.000.000.000 | Indirizzo IP Default Gateway |
| G-Porta TCP | Telnet | Servizio TCP richiesto |
| H-Cadenza HTTP (step di 0,5 secondi) | 120 | Timeout per la GET request se HTTP |
| K-Multisessione | Disabled | Abilitazione delle sessioni di servizio |
| L-IP_VAC (xxx.yyy.www.zzz) | 000.000.000.000 | Indirizzo IP di base del canale VAC |
| P-Porta VAC | 60000 | Porta UDP del canale VAC |
| Q-VAC abilitazione conversione HEX/ASCII | 000000000000000000000000 | Canali VAC abilitati |
| R-IP_VSC (xxx.yyy.www.zzz) | 000.000.000.000 | Indirizzo IP di base del canale VSC |
| S-VSC scelta protocollo | 00000000 | Protocollo dei canali VSC |
| M-Lingua AS/400 | Italian | Nazionalità dell'OS/400 |
| N-Record AS/400 | 256 | Dimensione del campo del file video |
| D-RF 433,33 Mhz roaming | 0000000000000000 | Slave componenti il roaming del sistema RF |
| Descrizione Master | | |

Abilitazione Configurazione allo start Abilitazione scrittura log

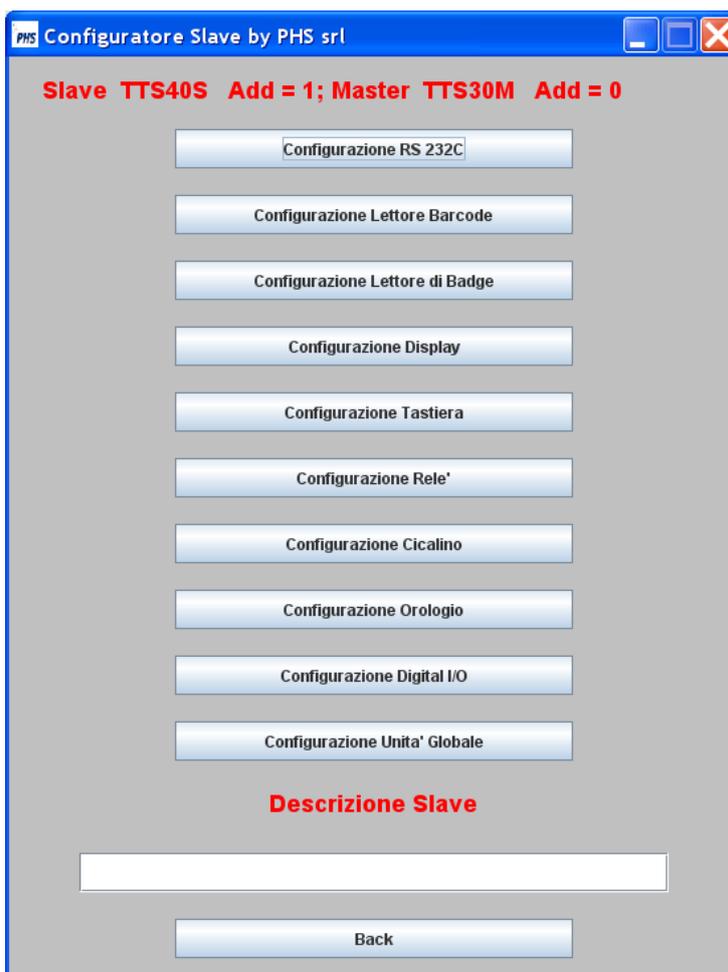
Apply Back

Configurazione tipo di slave

Preventivamente è necessario scegliere l'indice e il tipo di slave,

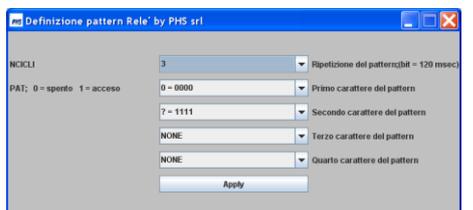
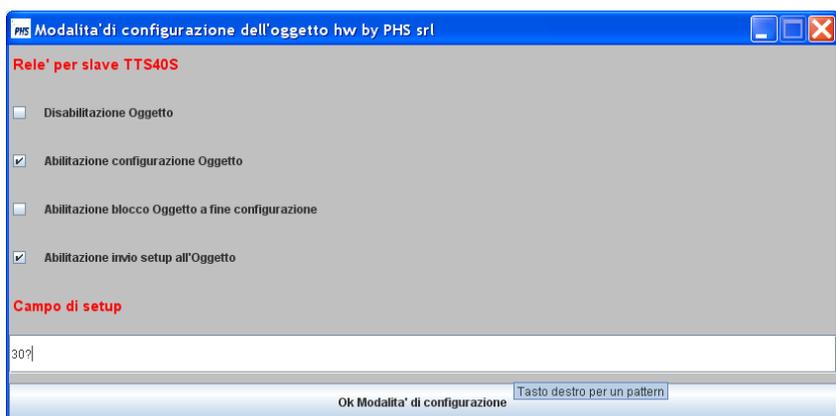
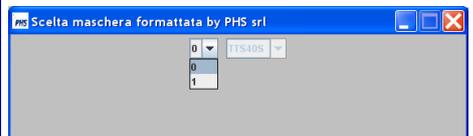
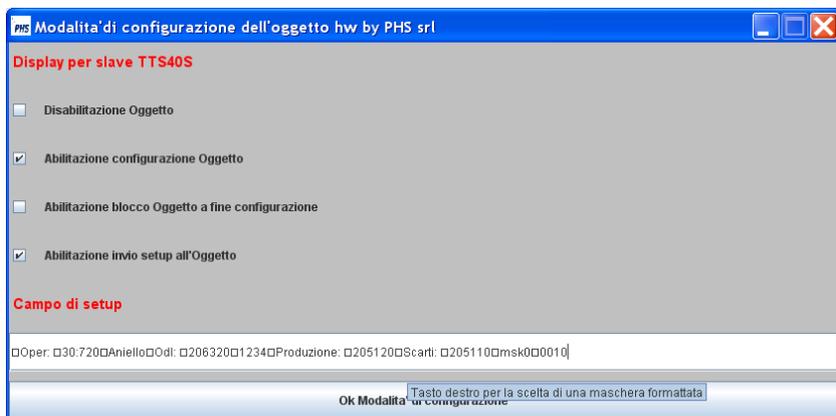
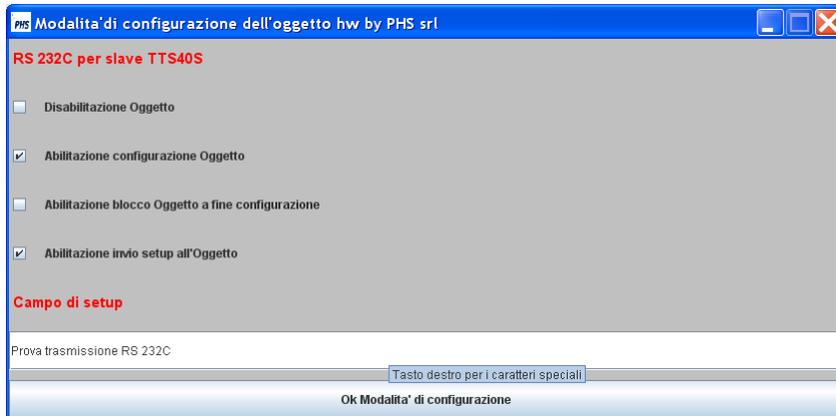


dopo, utilizzando il bottone **Configurazione tipo di slave**, si ottiene il pannello seguente,



che riporta gli oggetti hardware componenti lo slave selezionato, il campo per digitare la descrizione ed il bottone **Back** che riporta al pannello precedente.

Per ogni oggetto hardware (RS 232C, Barcode, ecc.) è disponibile un bottone che attiva due pannelli. Il primo permette di configurare la modalità con cui deve essere fatta la configurazione dell'oggetto.



Il pannello permette:

- a) **Disabilitazione oggetto**,esclude o include l'oggetto nella configurazione;
- b) **Abilitazione configurazione Oggetto**,disabilita o abilita l'invio delle stringhe di configurazione;
- c) **Abilitazione blocco Oggetto a fine configurazione**,per l'oggetto,a fine configurazione,è bloccata o sbloccata la porta di input;
- d) **Abilitazione invio setup all'Oggetto**,disabilita o abilita l'invio della stringa contenuta nel **Campo di setup**,che cambia a seconda dell'oggetto ed è ottenuta tramite la tastiera o un pannello attivato dal tasto destro, come indicato dal tooltip;
- e) **Ok Modalità di configurazione**,abilita il secondo pannello contenente i parametri dell'oggetto.

Il secondo pannello permette di configurare i parametri dell'oggetto, esso è del tipo seguente.

The screenshot shows a software window titled "Configuratore RS 232C by PHS srl" with a sub-header "RS 232C per Slave TTS40S Add = 1". The interface is divided into two main columns: "STREAMITS" on the left and "Descrizione" on the right. Each row in the "STREAMITS" column contains a checkbox, a label, and a dropdown menu. The "Descrizione" column provides a brief explanation for each parameter. At the bottom of the window, there are two buttons: "Apply" and "Back".

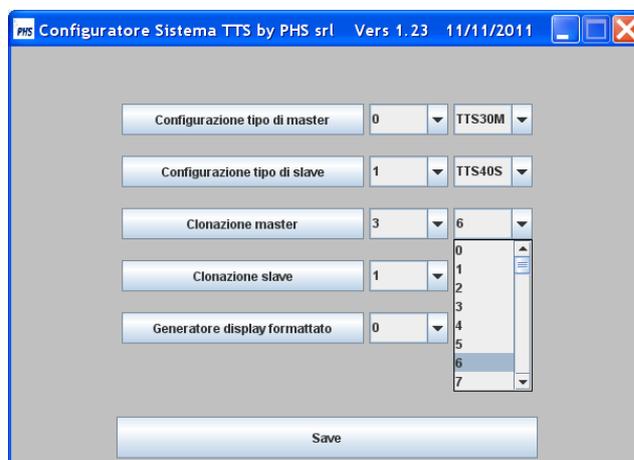
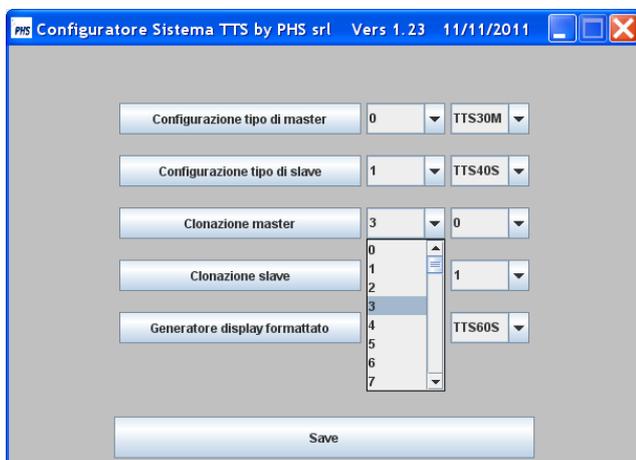
| STREAMITS | Valore | Descrizione |
|---|------------------------|---------------------------------------|
| <input type="checkbox"/> A-Protocollo in input configurabile | Stringa + CR | Protocollo del dispositivo collegato |
| <input type="checkbox"/> B-Frame_Baudrate configurabile | no parity,8 bit,1 stop | Formato del carattere |
| <input type="checkbox"/> C-Autoblocco configurabile | 9600 bit/sec | Velocità della porta RS 232C |
| <input type="checkbox"/> K-Ricezione continua configurabile | Disabled | Blocco della porta dopo una ricezione |
| <input type="checkbox"/> S-Filtro di input configurabile | Disabled | Dispositivo con trasmissione continua |
| <input type="checkbox"/> U-Conversione ASCIIHEX configurabile | z | Range dei caratteri ascii accettati |
| <input type="checkbox"/> J-Checksum configurabile | Disabled | Ogni carattere e' convertito |
| <input type="checkbox"/> M-Lunghezza di input configurabile | Disabled | Controllo integrità della ricezione |
| <input type="checkbox"/> Y-Ack_Nak configurabile | ?? - dummy | Algoritmo di calcolo |
| <input type="checkbox"/> X-Trasparenza configurabile | Disabled | Lunghezza record in input |
| <input type="checkbox"/> Z-Trigger configurabile | 255 | Accettazione dell'input |
| <input type="checkbox"/> D-Display RS 232C configurabile | Disabled | Trasparenza dell'input |
| <input type="checkbox"/> R-Good read configurabile | Disabled | Trigger di inattiva' in input |
| <input type="checkbox"/> H-XON_XOFF configurabile | Disabled | Presentazione dell'input sul display |
| <input type="checkbox"/> I-RTS_CTS configurabile | Disabled | Good read acustico sull'input |
| <input type="checkbox"/> N-Porta ausiliaria configurabile | Disabled | Controllo di flusso logico |
| <input type="checkbox"/> O-Range terminali RF configurabile | Disabled | Controllo di flusso hardware |
| <input type="checkbox"/> T-RF unidirezionale configurabile | 1 | Gestione sdoppiatore esterno |
| <input type="checkbox"/> V-RFID tag passivi configurabile | 1 | Limite superiore (1-19) |
| | tastiera | Limite inferiore (1-19) |
| | 0 | Risposta da slave al terminale RF |
| | | Dispositivi abilitati |
| | | Lunghezza SNR |

Impostati i parametri richiesti per l'applicazione si utilizzano i bottoni:

- a) **Apply**,valida le impostazioni introdotte e riporta al pannello precedente;
- b) **Back**, riporta al pannello precedente.

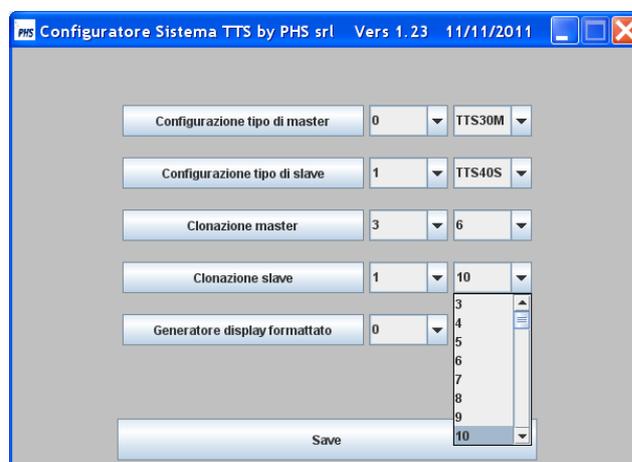
Clonazione master

La funzione permette di clonare il master di indice x (es. 3) sul master di indice y (es. 6); vengono clonati tutti i parametri tranne l'indirizzo IP (parametro C) e la descrizione; l'indirizzo IP poi va impostato obbligatoriamente.



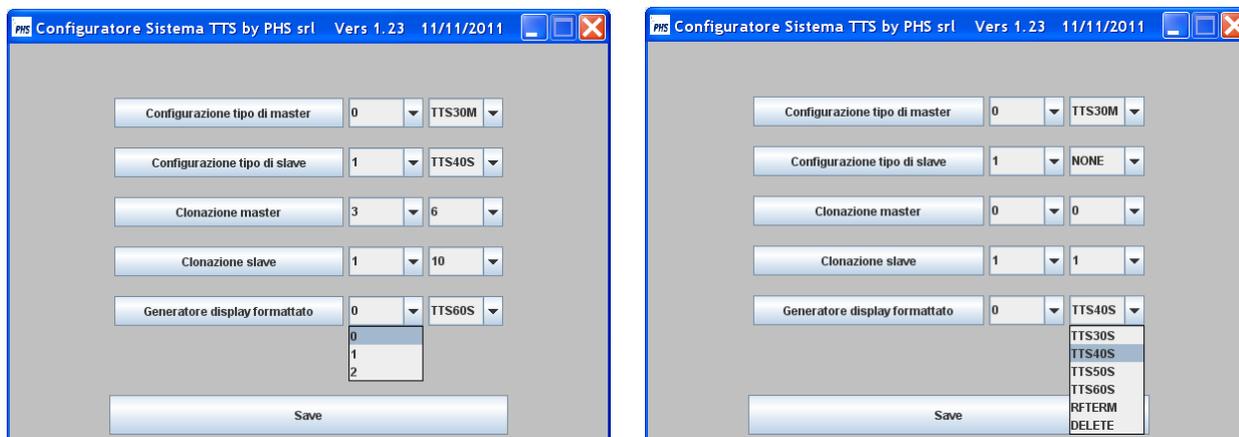
Clonazione slave

La funzione permette di clonare lo slave x (es. 1) sullo slave y (es. 10); vengono clonati tutti i parametri tranne la descrizione.

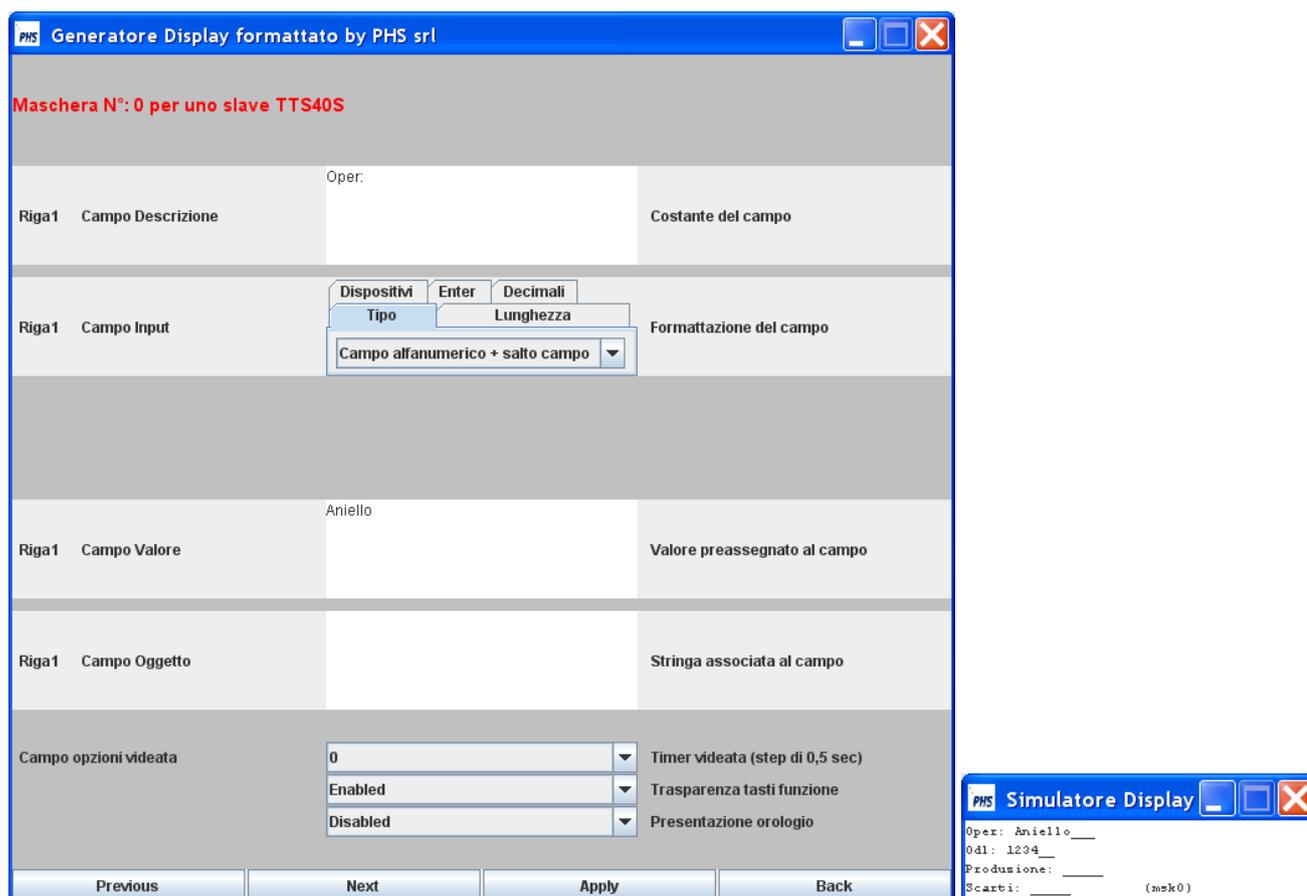


Generatore display formattato

La funzione permette di costruire fino a 128 videate formattate utilizzabili nell'applicazione, preventivamente è necessario scegliere l'indice e il tipo di slave disponibile di un display (numero di righe e di colonne),



dopo, utilizzando il bottone **Generatore display formattato**, si ottengono i pannelli seguenti.

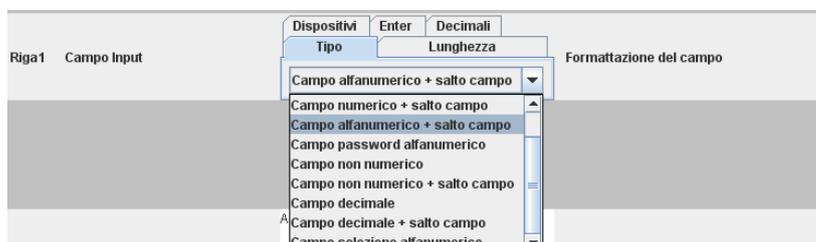


Il simulatore visualizza la videata che apparirà sul display fisico; il generatore permette di definire la struttura della singola riga mediante i campi:

- a) **Campo Descrizione**, che permette di definire la parte costante della riga;

b) **Campo Input**, permette di parametrizzare il campo di input, per l'acquisizione del dato, vale la sequenza:

1) **Tipo**, scelta del tipo di campo;



2) **Lunghezza**, scelta della lunghezza possibile;



3) **Dispositivi**, scelta dei dispositivi abilitati sul campo di input;



4) **Decimali**, numero di decimali per un campo numerico;

5) **Enter**, scelta della modalità di salto campo; sulle righe intermedie si sceglie il **Salta campo**, sull'ultima **Enter abilitato**.



c) **Campo Valore**, preassegnazione di un valore al campo di input;

d) **Campo Oggetto**, associazione di una stringa identificante il campo di input, che viene restituita insieme al dato imputato; generalmente viene utilizzato sull'ultima riga per identificare la videata presente sul display generante la transazione corrente.

Il campo **Opzioni videata**, permette di definire i tre parametri validi per la videata.

Sono utilizzabili quattro bottoni:

- a) **Previous**, permette di ritornare alla riga precedente;
- b) **Next**, permette di passare alla riga successiva;
- c) **Apply**, valida le impostazioni introdotte e riporta al pannello precedente;
- d) **Back**, riporta al pannello precedente.

TTSSIM – Simulatore logico di uno slave TTS

Il simulatore permette di simulare tutti gli oggetti hardware di uno slave TTS(display, tastiera, porta RS 232C,ecc.) ed i dispositivi esterni collegabili ad essi (PLC,bilance,lettore barcode, sensori ON/OFF,ecc.).

Il simulatore va' configurato come un qualsiasi slave del tipo IFxxx (vedi **RETETTS.pdf**), l'unico parametro non modificabile è il suo indirizzo IP,perché assume quello del PC su cui risiede; inoltre usando il browser v'è utilizzata la porta TCP (80 di default) indicata sul pannello Oggetti Hardware illustrato successivamente.

Il simulatore si attiva tramite l'eseguibile slaveTTS.exe contenuto nella cartella slaveTTS, nella quale sono anche presenti i file:

conf.xml (contiene i parametri di tutti gli oggetti hardware dello slave);

parSim.xml (salva lo stato della simulazione corrente);

inprx.txt (setup di default dell' input della porta RS 232C_COM1);

inprx1.txt (setup di default dell' input della porta RS 232C_COM2);

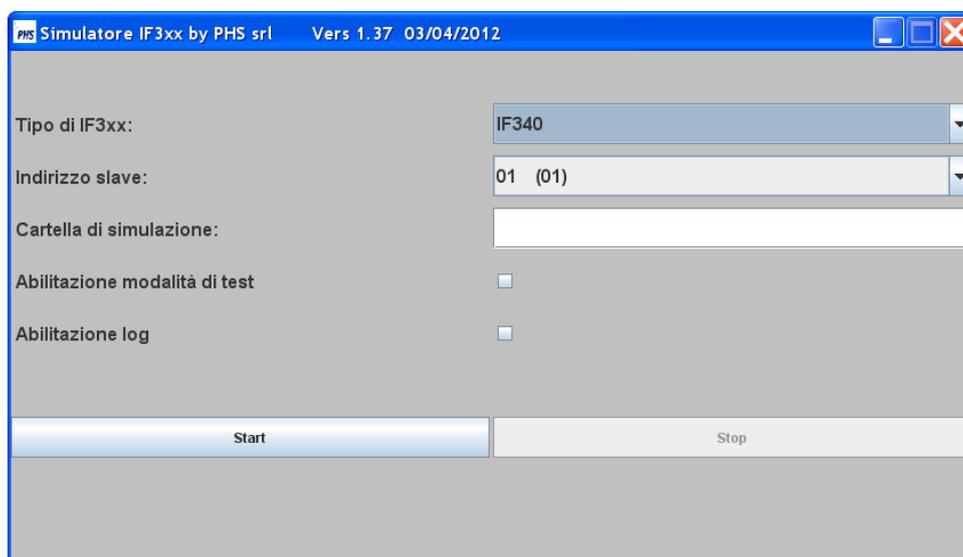
auxinprx.txt (setup di default dell' input della porta ausiliaria RS 232C_COM1);

auxinprx1.txt (setup di default dell' input della porta ausiliaria RS 232C_COM2);

parmBarcode.xml (setup di default della sequenza di 16 codici a barre)

parmDigio.xml. (setup di default delle forme d'onda degli input ON/OFF).

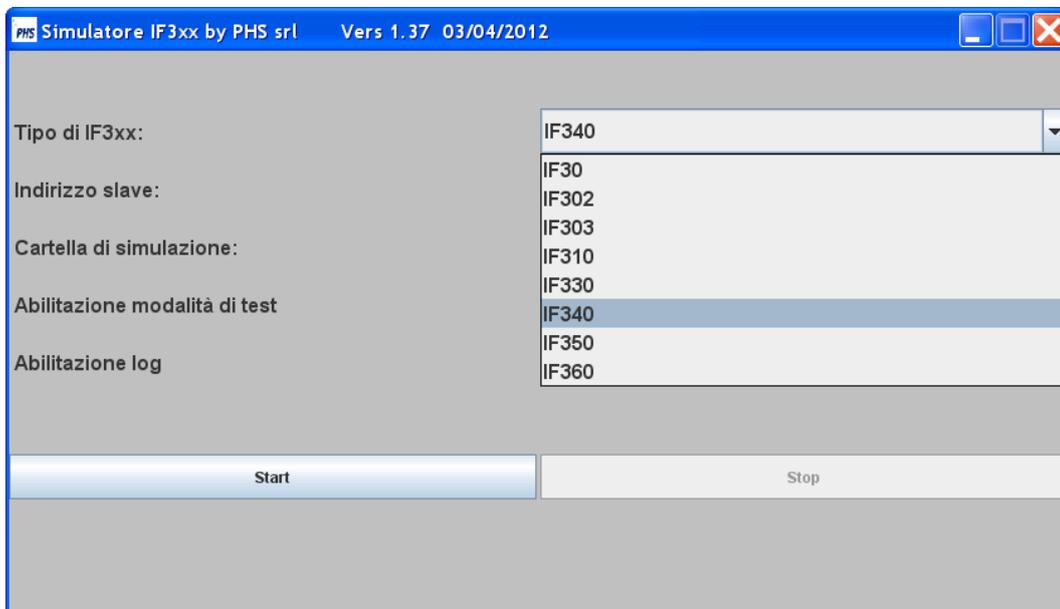
Il primo pannello che si attiva è il seguente:



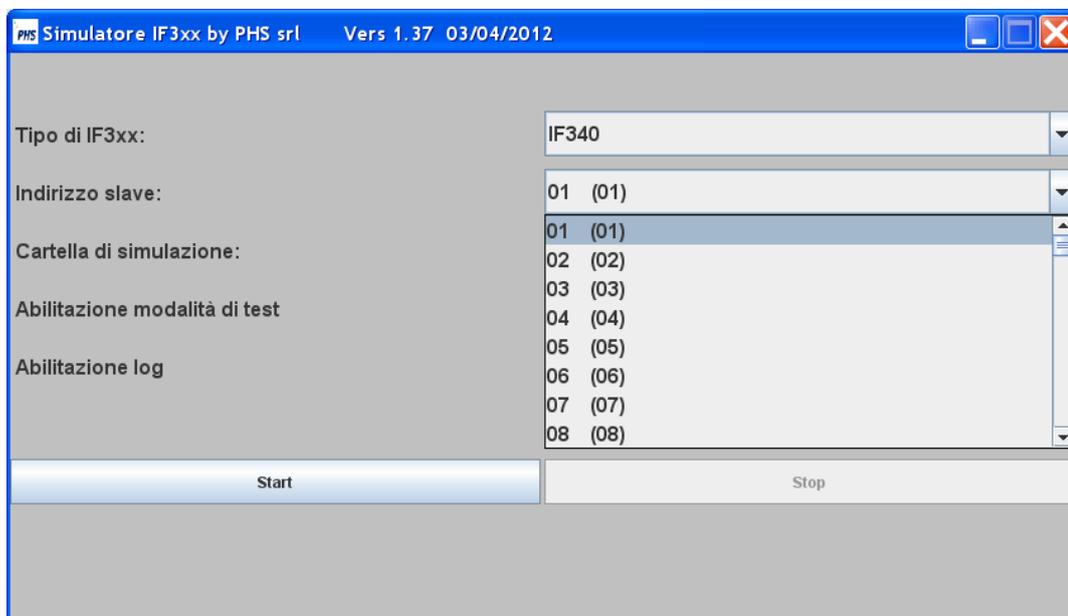
esso permette di configurare la simulazione dello slave desiderato.

La configurazione della simulazione si sviluppa come segue.

1) Scelta del tipo di slave

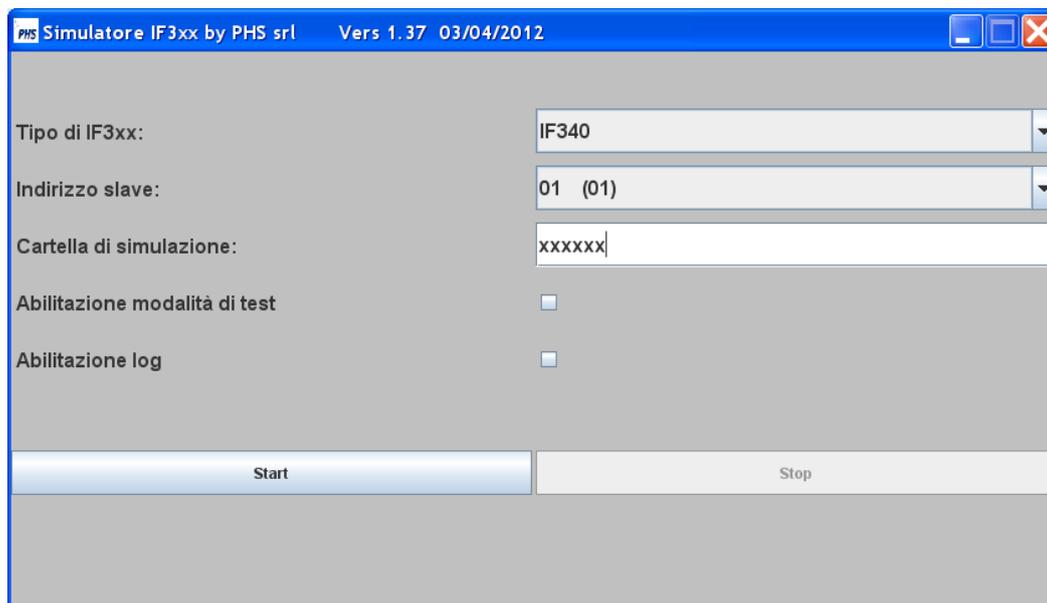


2) Indirizzo fisico dello slave



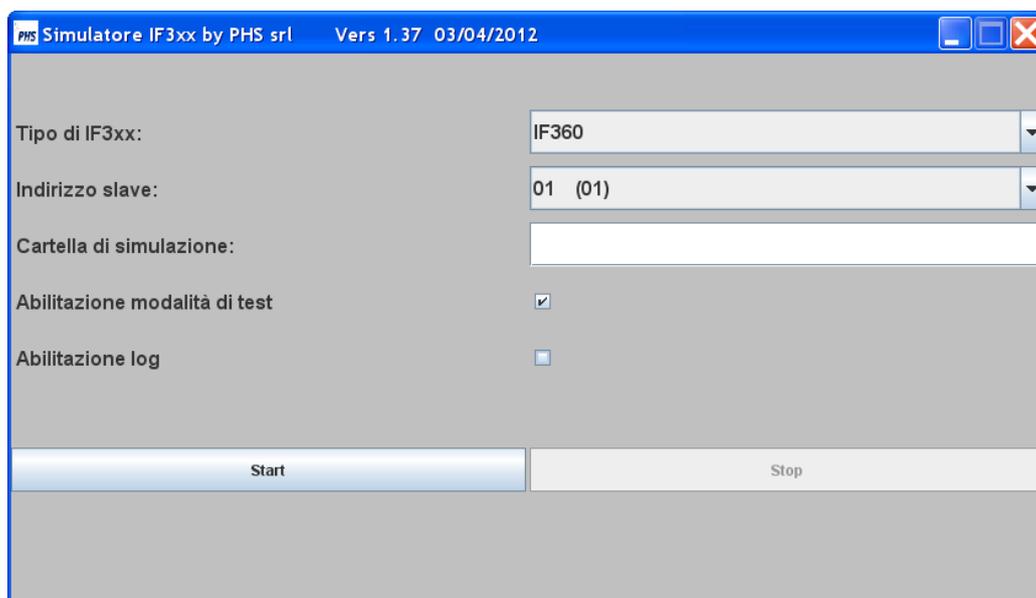
La scelta dell'indirizzo fisico corrisponde alla selezione, tramite microswitch, necessaria sullo slave fisico.

3) Cartella di simulazione



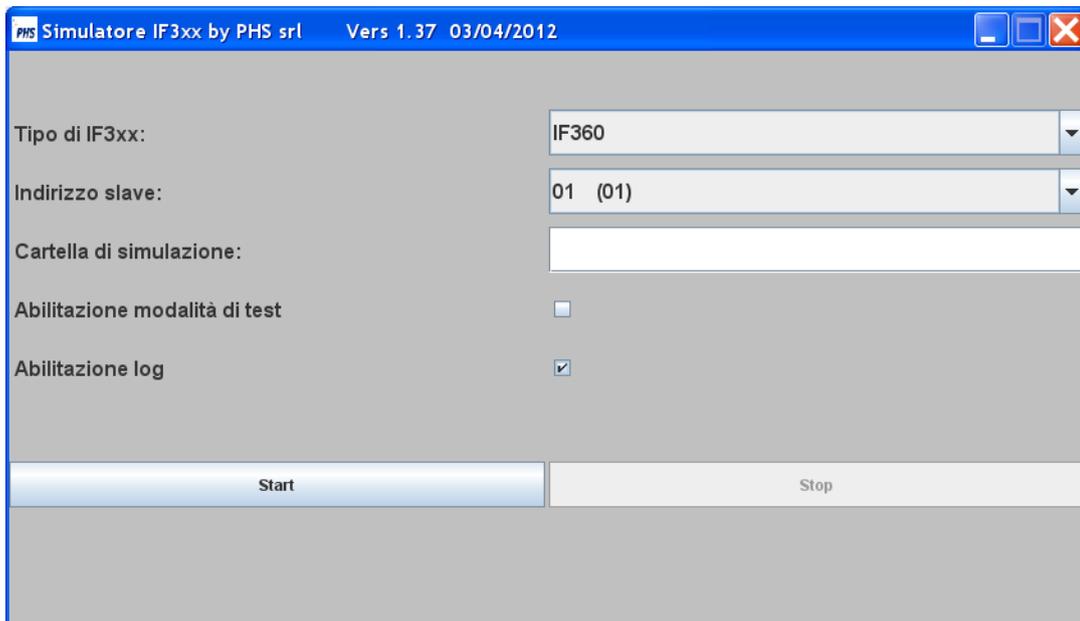
La cartella di simulazione deve contenere tutti i file (.txt,.xml) elencati in precedenza, essa permette di salvare la singola simulazione; se il campo è vuoto il simulatore usa i file contenuti nella cartella principale (slaveTTS); alla partenza della simulazione i file della cartella di simulazione sono copiati in quella principale e viceversa alla chiusura della simulazione.

4) Modalità di test



La modalità di test permette di velocizzare la simulazione, poiché riduce alcuni timeout presenti e necessari sullo slave fisico.

5) Log dei frame TTS sulla porta ethernet



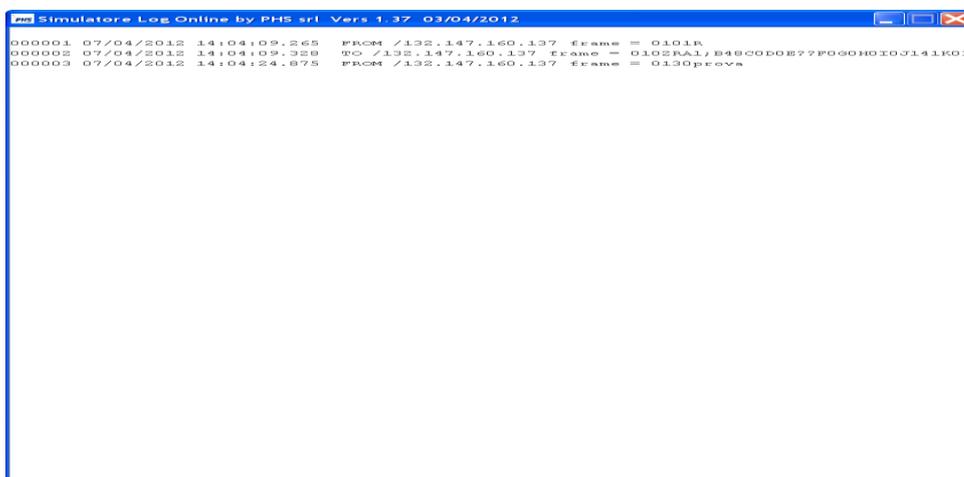
L'abilitazione del log comporta:

- a) la creazione di un file giornaliero **slavelogGGMMAAAA.txt** del tipo:

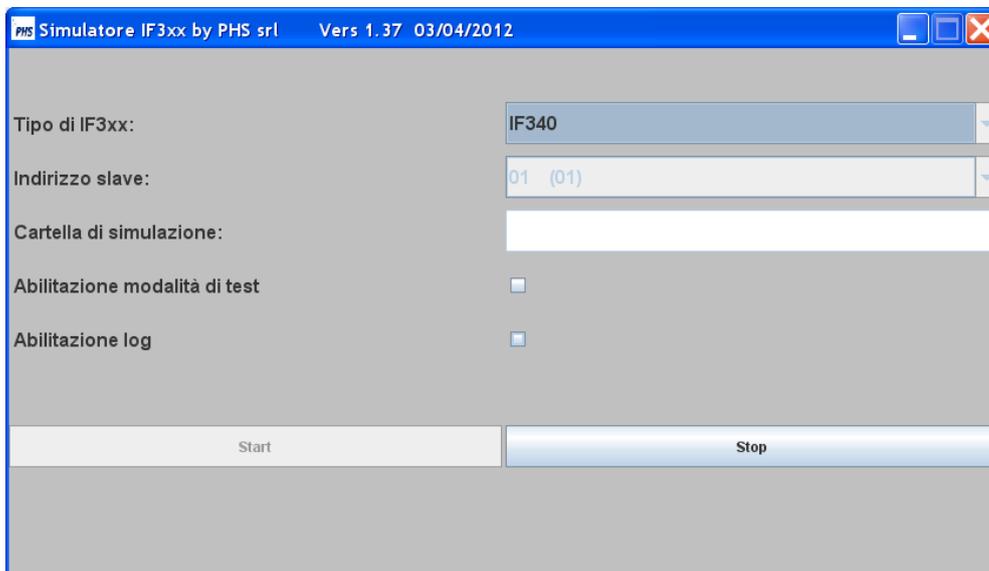
```
000000 02/04/2012 11:00:10.984 FROM /192.168.0.138 frame = 0101R
000001 02/04/2012 11:00:26.125 FROM /192.168.0.138 frame = 0101WA0>B48J141
000002 02/04/2012 11:00:26.187 TO /192.168.0.138 frame = 0102WABJ
000003 02/04/2012 11:00:42.718 TO /192.168.0.138 frame = 0100S|prov
000005 02/04/2012 11:00:45.484 TO /192.168.0.138 frame = 0100S|prov
000006 02/04/2012 11:00:50.062 TO /192.168.0.138 frame = 0100S|NO RD
000007 02/04/2012 11:15:53.968 FROM /192.168.0.138 frame = 0101R
000008 02/04/2012 11:16:01.843 TO /192.168.0.138 frame = 0100S|prov
```

dove FROM e TO sono relazionati al server;

- b) l'attivazione di un pannello presentante il log on line.



Dopo l'attivazione del configuratore è abilitato il bottone di Stop, che permette la ripartenza o il cambio di configurazione della simulazione.



La simulazione, a seconda del tipo di slave, avviene tramite due pannelli:

- a) il pannello per il display, la tastiera e i bottoni;
- b) il pannello degli oggetti hardware (o dispositivi componenti lo slave).

Display e Tastiera

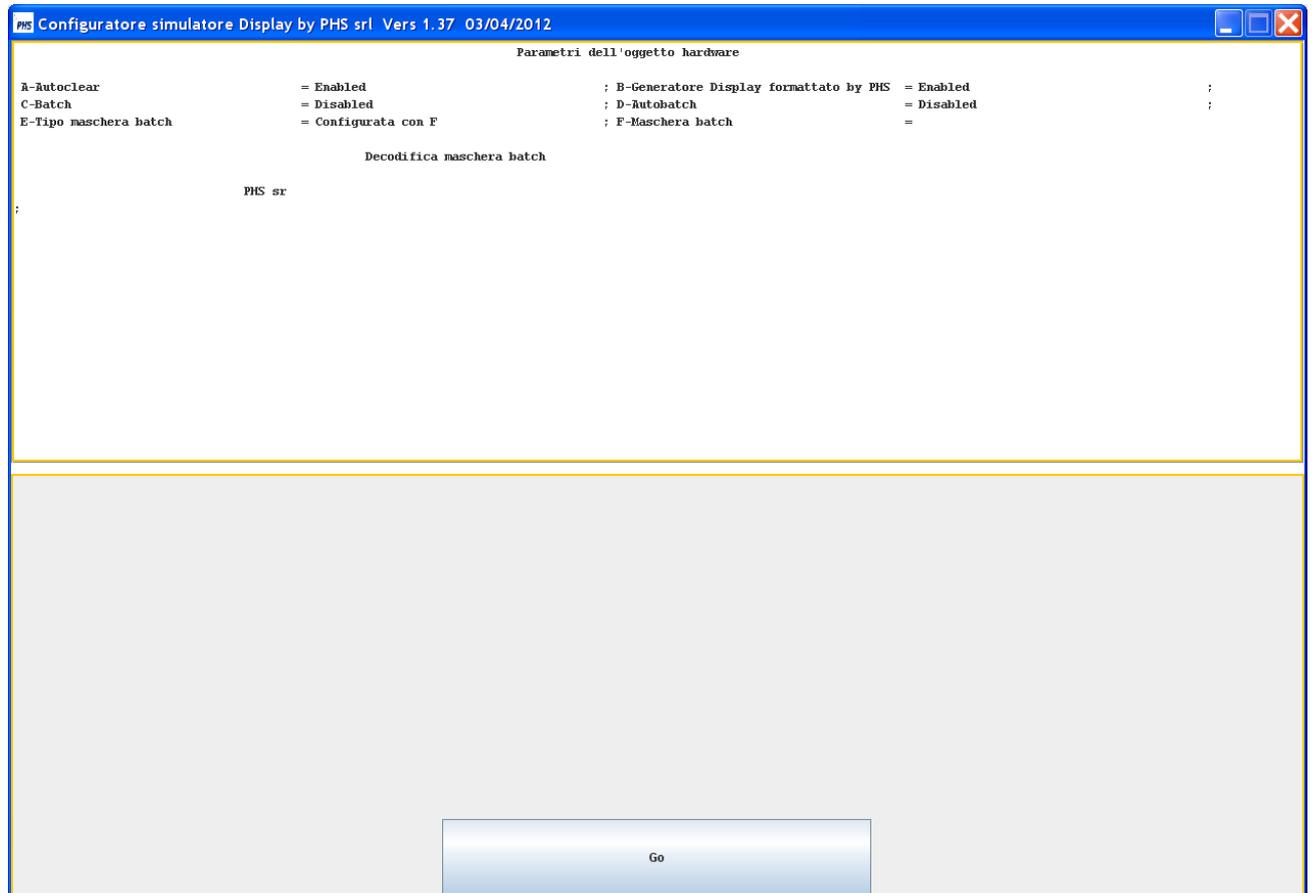


Sul display si attiva il tooltip:

%Tasto destro per Parametri Display% \$Doppio click per videata da host\$

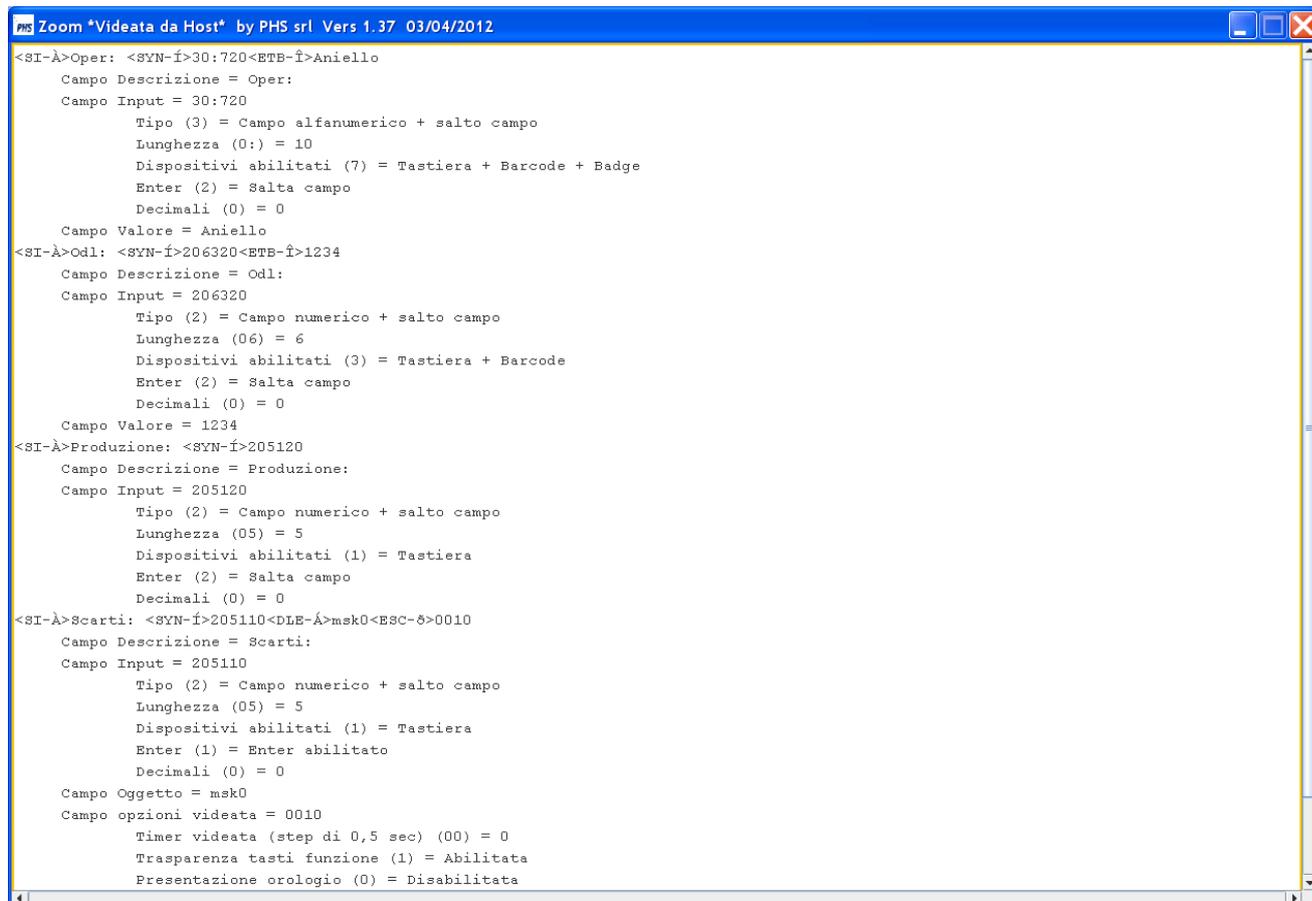
il tasto destro permette di visualizzare i parametri di configurazione che l'oggetto hardware Display ha ricevuto; mentre il doppio click presenta la decodifica della videata presente.

Utilizzo del tasto destro



Il bottone Go permette di chiudere il pannello.

Utilizzo del doppio click

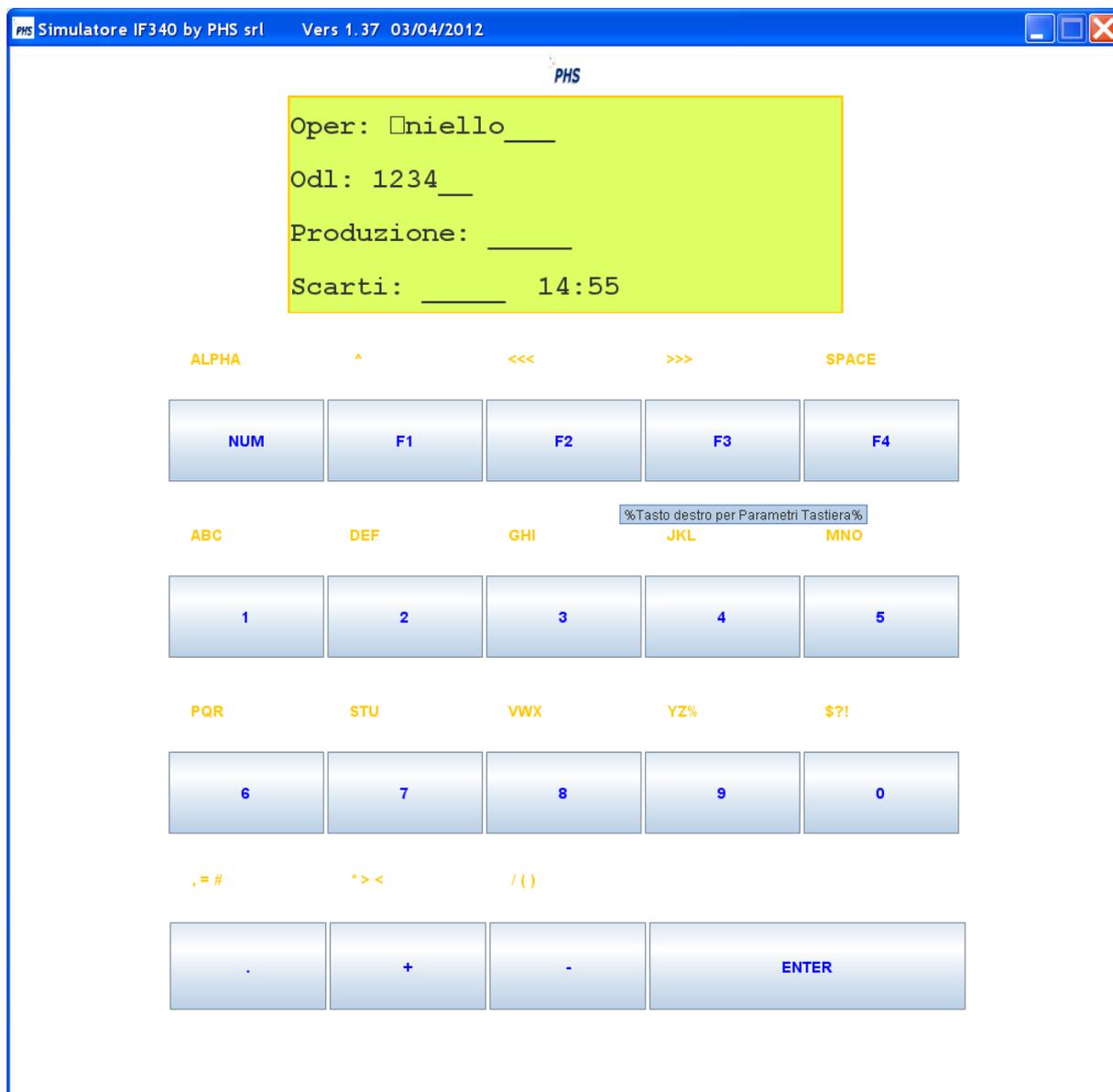


```
PHS Zoom *Videata da Host* by PHS srl Vers 1.37 03/04/2012
<SI-À>Oper: <SYN-í>30:720<ETB-î>Aniello
  Campo Descrizione = Oper:
  Campo Input = 30:720
    Tipo (3) = Campo alfanumerico + salto campo
    Lunghezza (0:) = 10
    Dispositivi abilitati (7) = Tastiera + Barcode + Badge
    Enter (2) = Salta campo
    Decimali (0) = 0
  Campo Valore = Aniello
<SI-À>Odl: <SYN-í>206320<ETB-î>1234
  Campo Descrizione = Odl:
  Campo Input = 206320
    Tipo (2) = Campo numerico + salto campo
    Lunghezza (06) = 6
    Dispositivi abilitati (3) = Tastiera + Barcode
    Enter (2) = Salta campo
    Decimali (0) = 0
  Campo Valore = 1234
<SI-À>Produzione: <SYN-í>205120
  Campo Descrizione = Produzione:
  Campo Input = 205120
    Tipo (2) = Campo numerico + salto campo
    Lunghezza (05) = 5
    Dispositivi abilitati (1) = Tastiera
    Enter (2) = Salta campo
    Decimali (0) = 0
<SI-À>Scarti: <SYN-í>205110<DLE-Á>msk0<ESC-ò>0010
  Campo Descrizione = Scarti:
  Campo Input = 205110
    Tipo (2) = Campo numerico + salto campo
    Lunghezza (05) = 5
    Dispositivi abilitati (1) = Tastiera
    Enter (1) = Enter abilitato
    Decimali (0) = 0
  Campo Oggetto = msk0
  Campo opzioni videata = 0010
    Timer videata (step di 0,5 sec) (00) = 0
    Trasparenza tasti funzione (1) = Abilitata
    Presentazione orologio (0) = Disabilitata
```

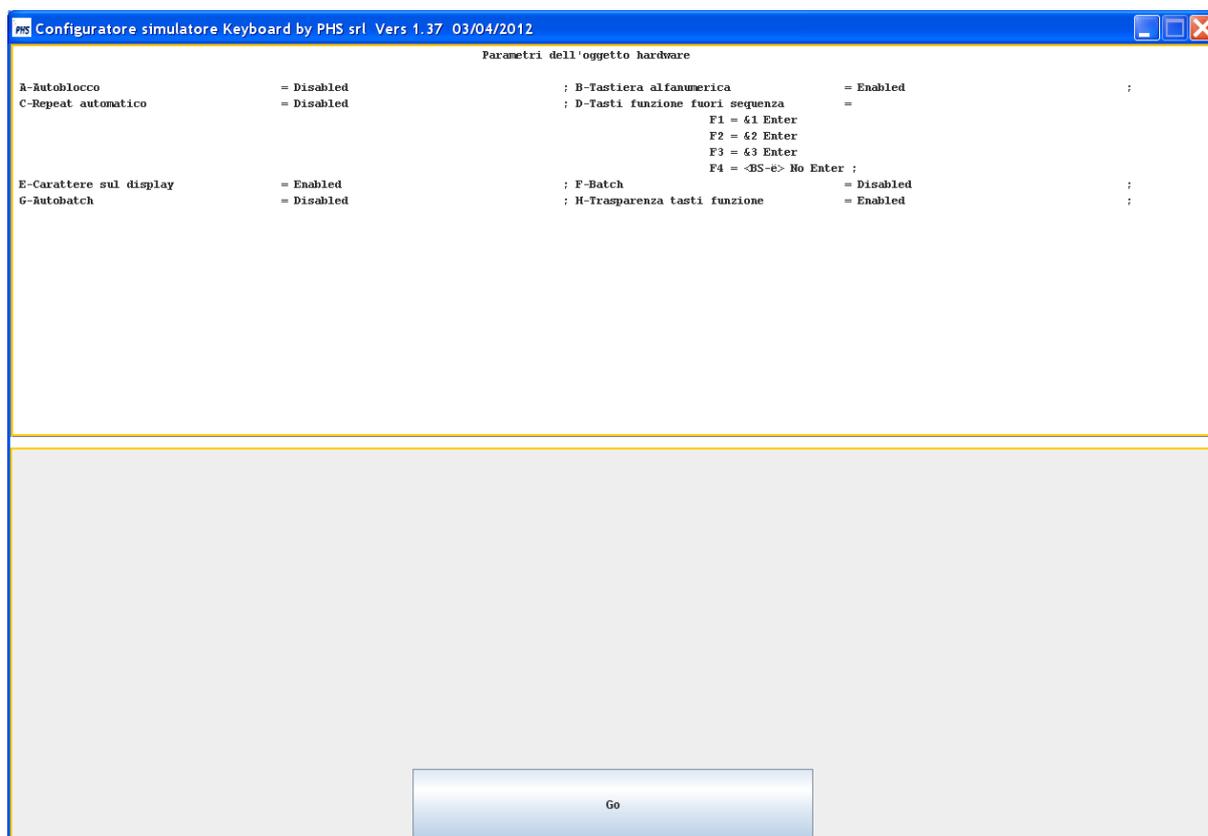
Sulla tastiera si attiva il tooltip:

%Tasto destro per Parametri Tastiera%

il tasto destro permette di visualizzare i parametri di configurazione che l'oggetto hardware Tastiera ha ricevuto.

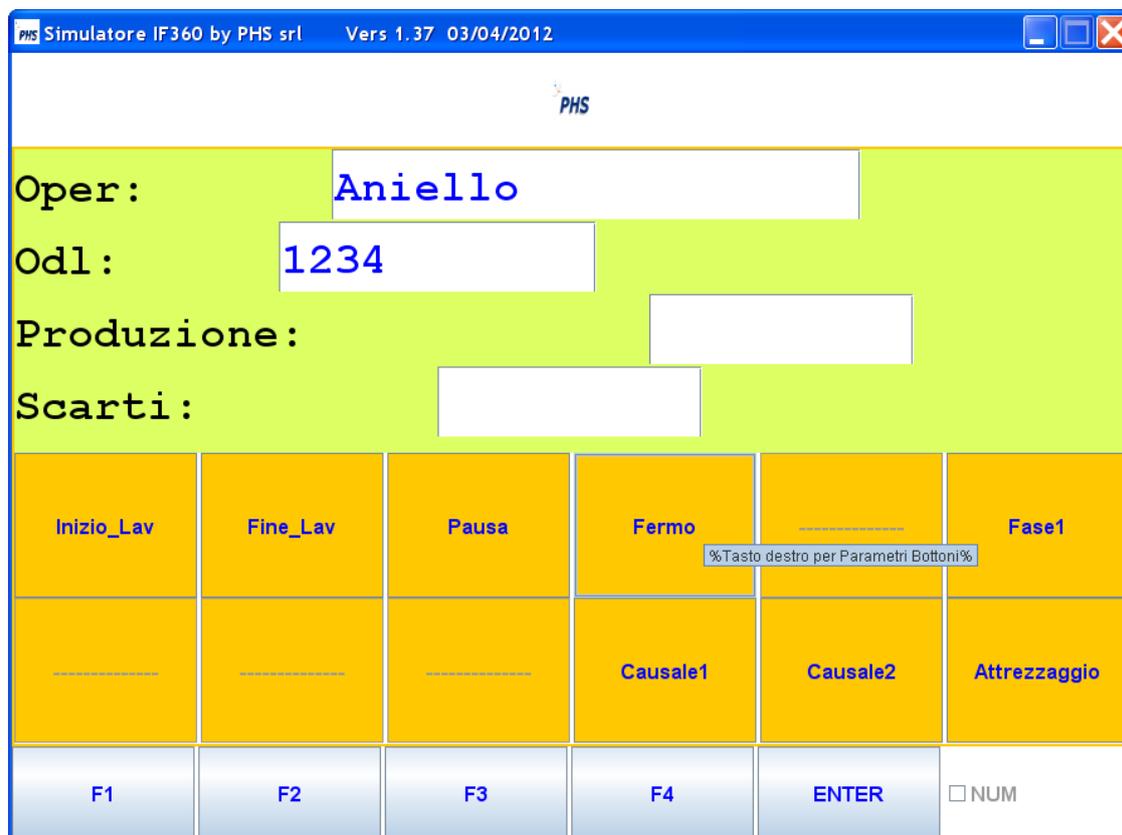


Utilizzo del tasto destro



Il bottone Go permette di chiudere il pannello.

Bottoni

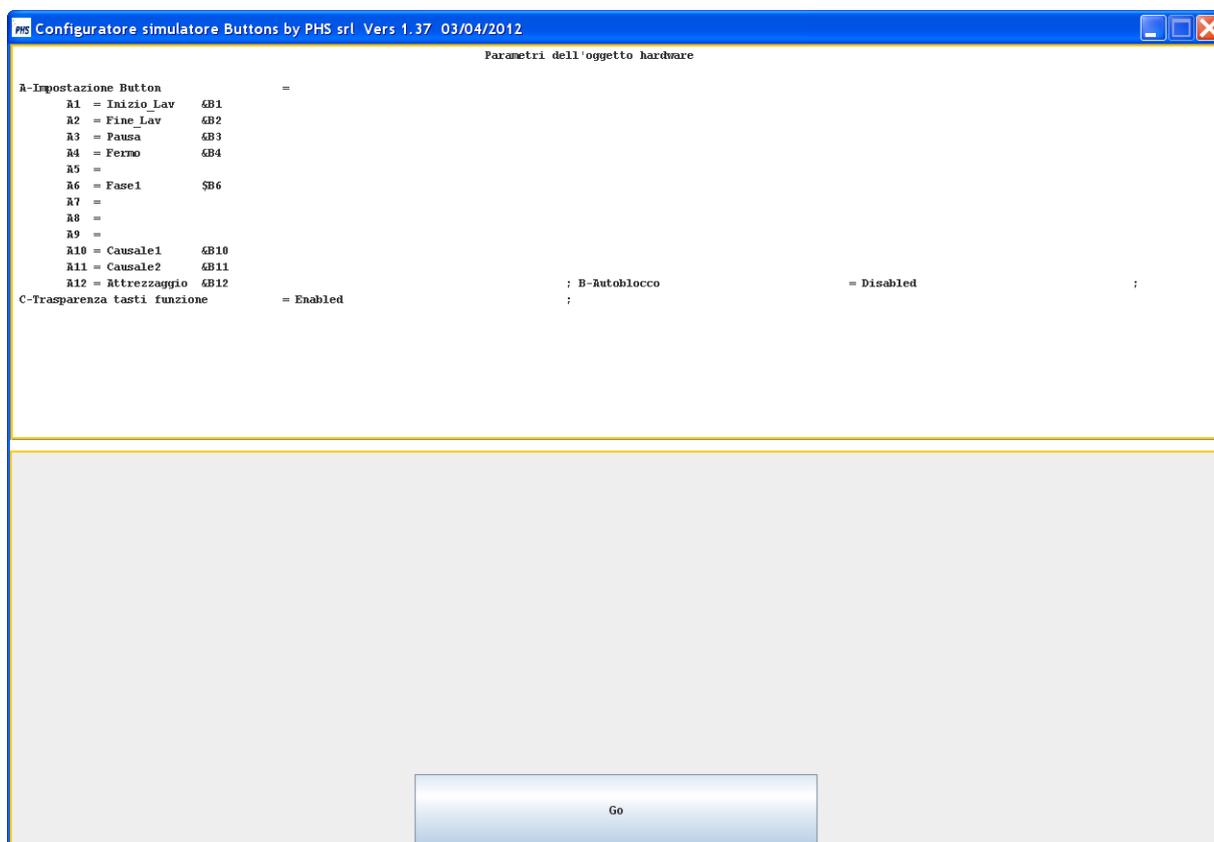


Sui bottoni si attiva il tooltip:

%Tasto destro per Parametri Bottoni%

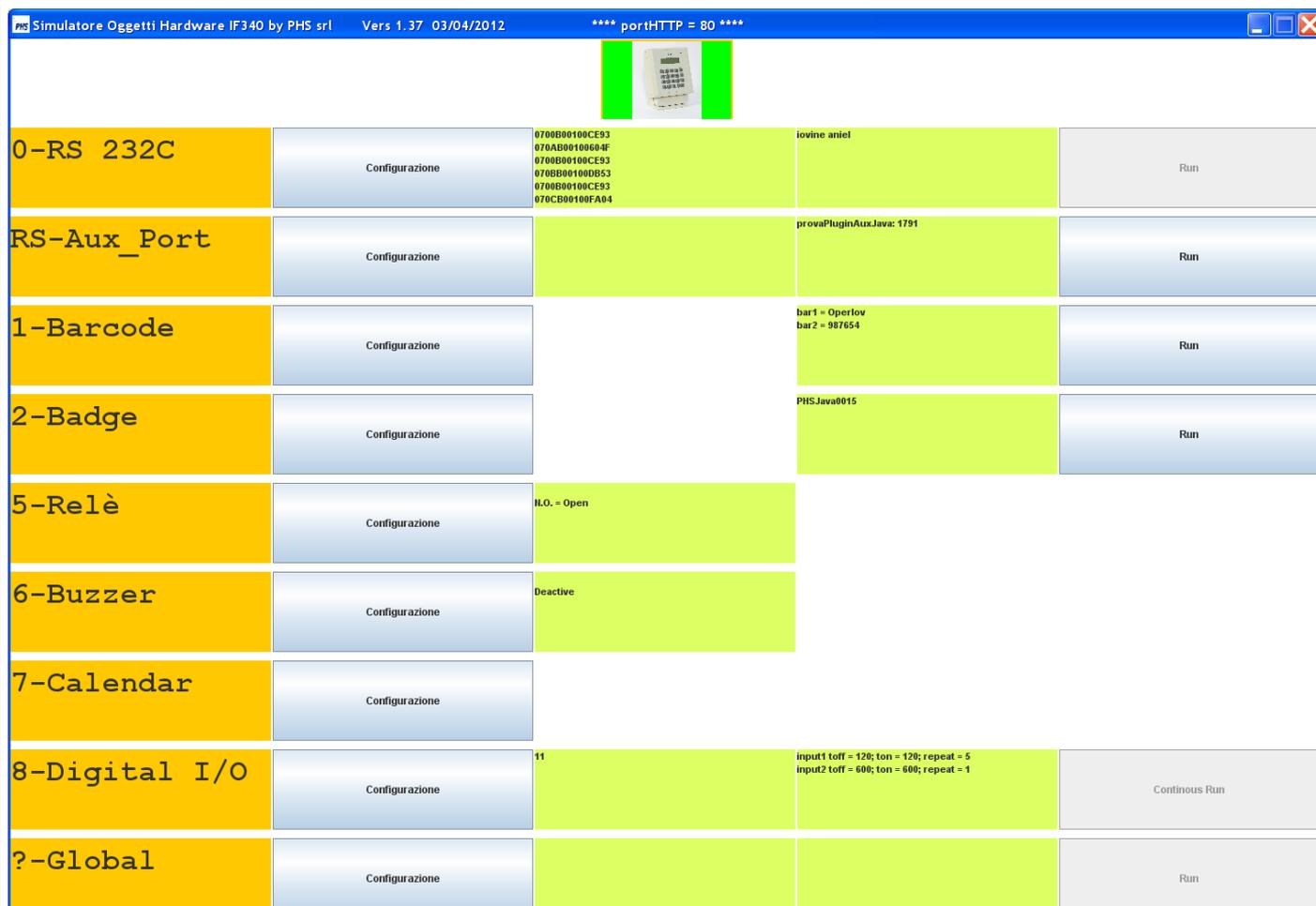
il tasto destro permette di visualizzare i parametri di configurazione che l'oggetto hardware Bottoni ha ricevuto.

Utilizzo del tasto destro



Il bottone Go permette di chiudere il pannello.

Oggetti Hardware



| | | | | |
|---------------|----------------|--|--|----------------|
| 0-RS 232C | Configurazione | 0700B00100CE93 070AB00100604F 0700B00100CE93 0700B001000053 0700B00100CE93 070CB00100FA04 | iovine aniel | Run |
| RS-Aux_Port | Configurazione | | provaPluginAux.Java: 1791 | Run |
| 1-Barcode | Configurazione | | bar1 = Operlov bar2 = 987654 | Run |
| 2-Badge | Configurazione | | PHS.Java0015 | Run |
| 5-Relè | Configurazione | N.O. - Open | | |
| 6-Buzzer | Configurazione | Deactive | | |
| 7-Calendar | Configurazione | | | |
| 8-Digital I/O | Configurazione | 11 | input1 toff = 120; ton = 120; repeat = 5 input2 toff = 600; ton = 600; repeat = 1 | Continuous Run |
| ?-Global | Configurazione | | | Run |

Il pannello è composto da righe, ognuna è relativa ad un oggetto hardware; ogni riga dispone di:

- 1) una descrizione;
- 2) un bottone Configurazione;
- 3) un campo di output verso il dispositivo (eventualmente esterno) da simulare;
- 4) un campo di input dal dispositivo (eventualmente esterno) da simulare;
- 5) un bottone di Run per generare la trasmissione del campo di input.

RS 232C

0-RS 232C

Configurazione

0700B00100CE53
070AB00100694F
0700B00100CE53
0700B00100DB53
0700B00100CE53
070CB00100FA04

Iovine aniel

Run

Attivando il bottone di configurazione si ottiene il pannello seguente:



esso è diviso in due campi:

- 1) il primo permette di visualizzare i parametri di configurazione che l'oggetto hardware RS 232C ha ricevuto;
- 2) il secondo configura la simulazione di un dispositivo seriale collegabile sulla porta RS 232C di uno slave fisico, tramite gli oggetti visualizzati:
 - a) la flag **Trasmissione continua** mantiene il bottone Run sempre attivo, generando la trasmissione continua del contenuto del campo di input;
 - b) la flag **Restore inprx.txt** carica sul campo di input il contenuto del file;
 - c) il bottone **Setup Memory** attiva il pannello di simulazione di una area di memoria di 64k (byte o word);
 - d) il bottone **Go** lancia la simulazione e chiude il pannello.

Utilizzo del bottone Setup Memory

Il valore di ogni locazione di memoria è esadecimale o ASCII a seconda del dispositivo utilizzato (PLC,RFID,ecc.).

| | 0000 | 0001 | 0002 | 0003 | 0004 | 0005 | 0006 | 0007 | 0008 | 0009 | 000A | 000B | 000C | 000D | 000E | 000F |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0000 | 696F | 7669 | 6E65 | 2061 | 6E69 | 656C | 6C6F | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0010 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0030 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0040 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0050 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0060 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0070 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0080 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0090 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 00A0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 00B0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 00C0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 00D0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 00E0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 00F0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0100 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0110 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0120 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0130 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0140 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0150 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0160 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0170 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0180 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 0190 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 01A0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 01B0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 01C0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 01D0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 01E0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |
| 01F0 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 | 2020 |

Setup from address (o Goto) Address = 0000

Fill all memory 00

UP Down Numero Blocchi in dec. 1

L'area di memoria è divisa in blocchi di 512 (byte o word) esadecimale o ASCII; le funzioni disponibili sono:

- 1) **Setup from address:**
 - a) nel campo Address si digita l'indirizzo di partenza (in esadecimale),
 - b) nel campo dati la sequenza di byte da settare (esadecimale o ASCII).
- 2) **Fill all memory** setup di tutta la memoria con la sequenza digitata.
- 3) I bottoni **UP** e **Down** permettono di scorrere la memoria a step di blocchi indicati nel campo relativo.

RS 232C Porta Ausiliaria



Per la porta ausiliaria vale quanto detto precedentemente per la porta primaria.

Barcode

1-Barcode

Configurazione

bar1 = Operlow
bar2 = 987654

Run

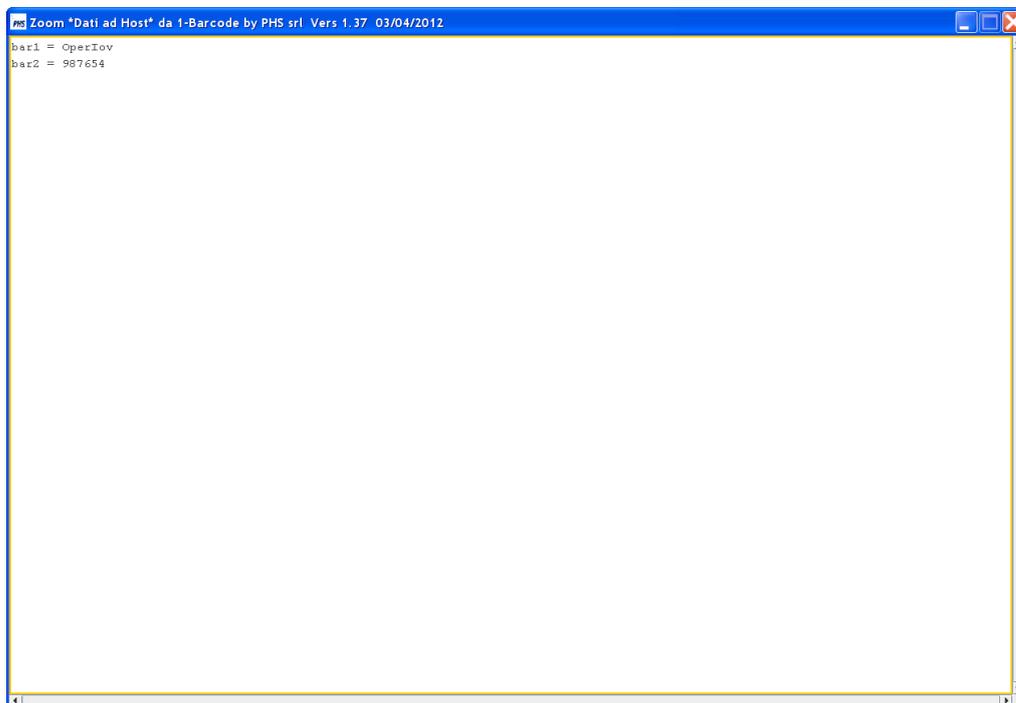
Attivando il bottone di configurazione si ottiene il pannello seguente:



esso è diviso in due campi:

- 3) il primo permette di visualizzare i parametri di configurazione che l'oggetto hardware Barcode ha ricevuto;
- 4) il secondo configura la simulazione del lettore barcode collegabile sulla porta barcode di uno slave fisico, tramite gli oggetti visualizzati:
 - a) la flag **Sequenza abilitata** determina l'utilizzo del menù barcode presente sul campo di input;
 - b) la flag **Restore parmBarcode.xml** carica sul campo di input il contenuto del file;
 - c) il bottone **Go** lancia la simulazione e chiude il pannello.

Per il campo di input è possibile utilizzare, per comodità, (tramite il doppio click) un pannello di zoom per visualizzare o modificare il contenuto del campo tramite la tastiera.

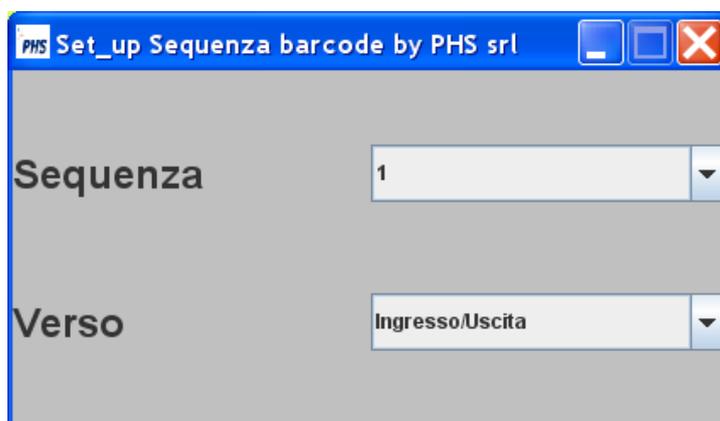


L'inserimento di una voce di menù deve rispettare la sintassi:

barn = xxxxxxxx

dove n va da 1 a 16. I barcode presenti nel menù sono trasmessi in sequenza, se essa è abilitata, altrimenti è trasmesso sempre il primo della lista.

Per gestire la sequenza sul campo di input è possibile utilizzare il pannello seguente, attivabile tramite il tasto destro.

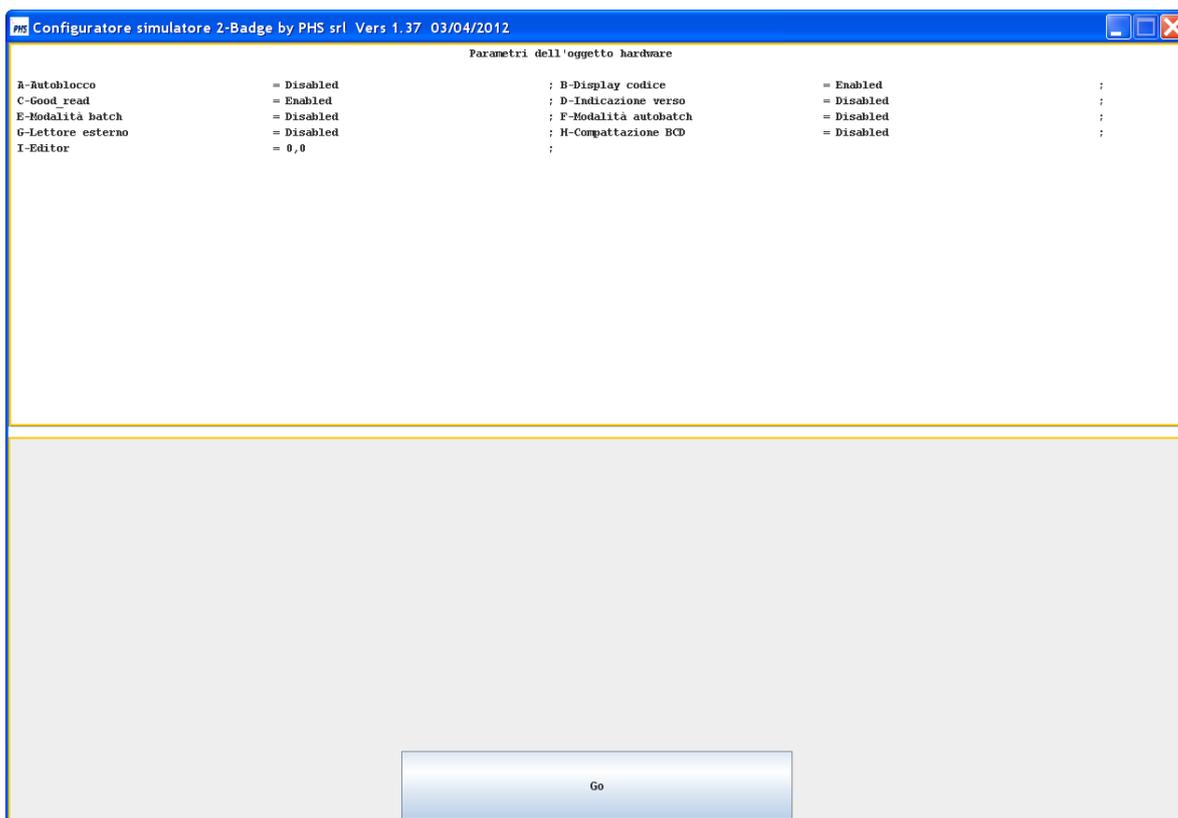


Il campo **Sequenza** permette di settare la voce di menù da cui partire; il campo **Verso** permette di aggiungere l'indicazione di verso se configurata.

Badge



Attivando il bottone di configurazione si ottiene il pannello seguente:



Il bottone Go lancia la simulazione e chiude il pannello.

Per il campo di input è possibile utilizzare, per comodità, (tramite il doppio click) un pannello di zoom per visualizzare o modificare il contenuto del campo tramite la tastiera.

Per gestire il verso sul campo di input è possibile utilizzare il pannello seguente, attivabile tramite il tasto destro.



Il campo **Verso** permette di aggiungere l'indicazione di verso se configurata.

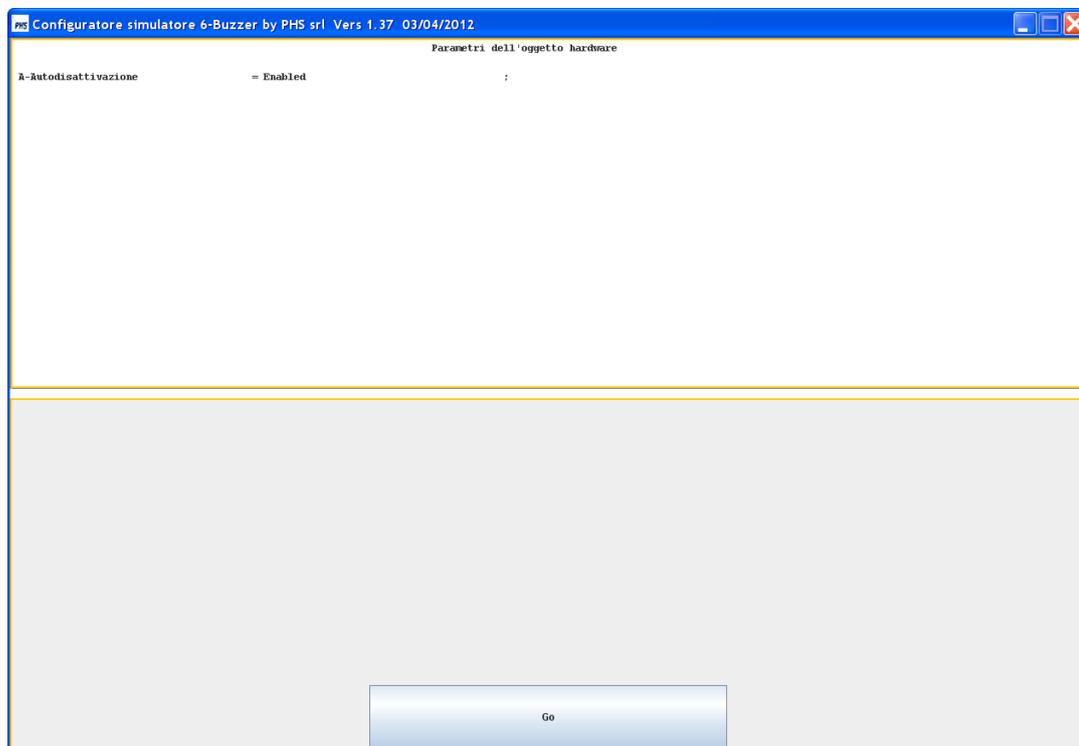
Relè



Buzzer



La simulazione del relè e del buzzer sono simili; attivando il bottone di configurazione si ottiene un pannello del tipo:



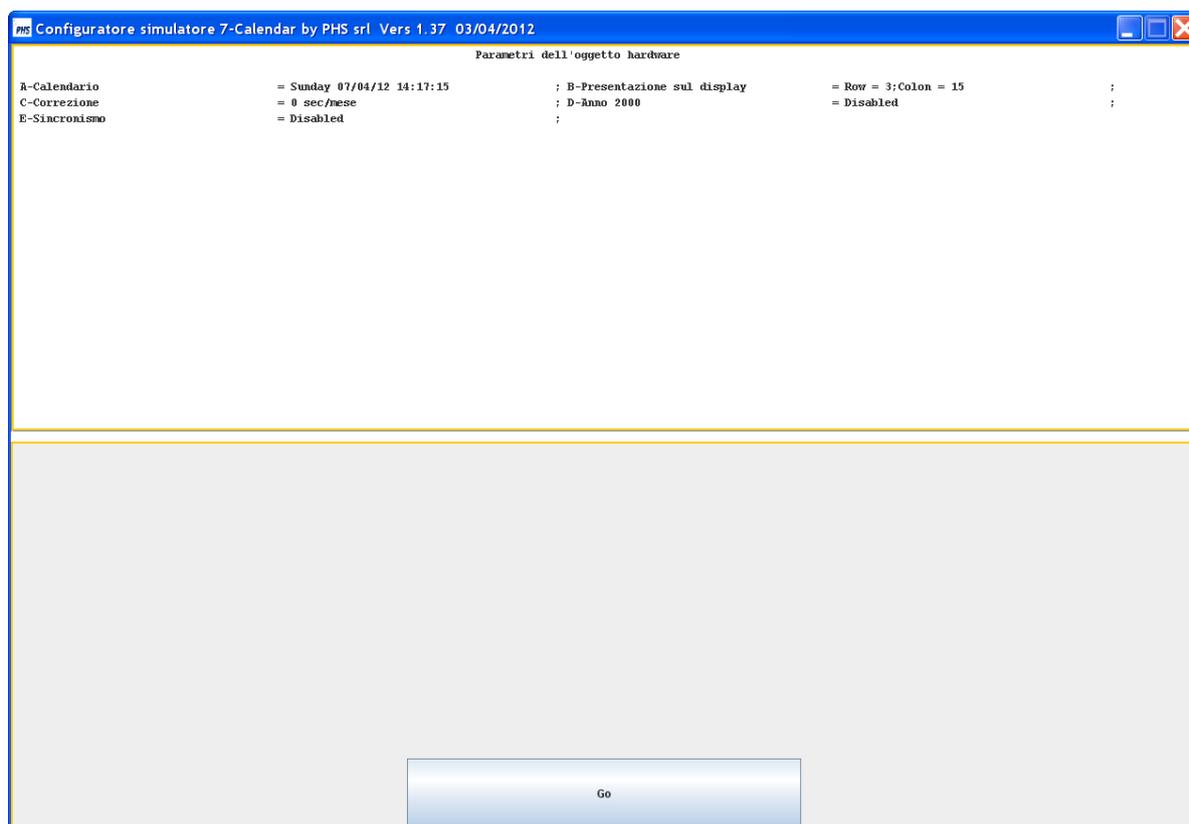
Il bottone Go chiude il pannello.

Calendario

7-Calendar

Configurazione

Attivando il bottone di configurazione si ottiene il pannello seguente:



Il bottone Go chiude il pannello.

Digital I/O

8-Digital I/O

Configurazione

11

input1 toff = 120; ton = 120; repeat = 5
input2 toff = 600; ton = 600; repeat = 1

Run

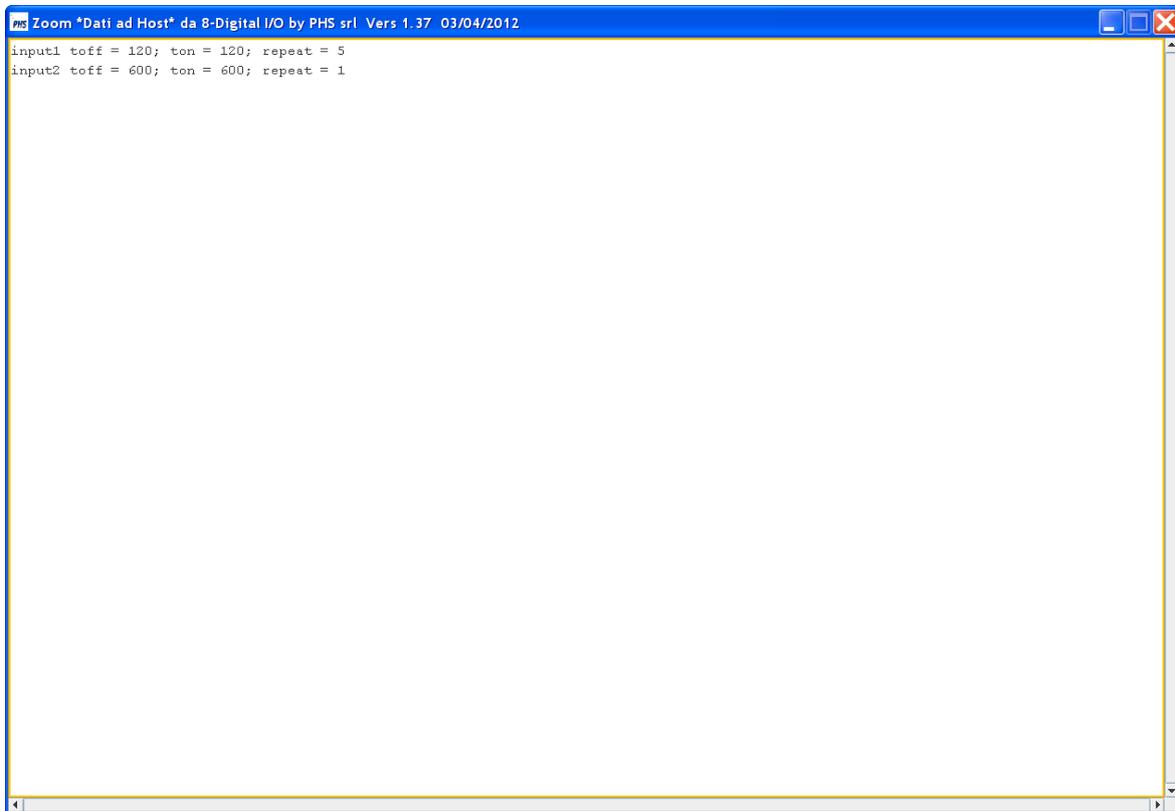
Attivando il bottone di configurazione si ottiene il pannello seguente:



esso   diviso in due campi:

- 1) il primo permette di visualizzare i parametri di configurazione che l'oggetto hardware Digital I/O ha ricevuto;
- 2) il secondo configura la simulazione dei sensori on/off collegabili sulla porta di input digitale di uno slave fisico, tramite gli oggetti visualizzati:
 - a) la flag **Trasmissione continua** mantiene il bottone Run sempre attivo, generando la trasmissione continua delle variazioni dei segnali descritti sul campo di input;
 - b) la flag **Restore parmDigio.xml** carica sul campo di input il contenuto del file;
 - c) il bottone **Go** lancia la simulazione e chiude il pannello.

Per i campi di input ed output è possibile utilizzare, per comodità, (tramite il doppio click) un pannello di zoom per visualizzare o modificare il contenuto del campo, ove permesso, tramite la tastiera.



L'inserimento di un segnale deve rispettare la sintassi:

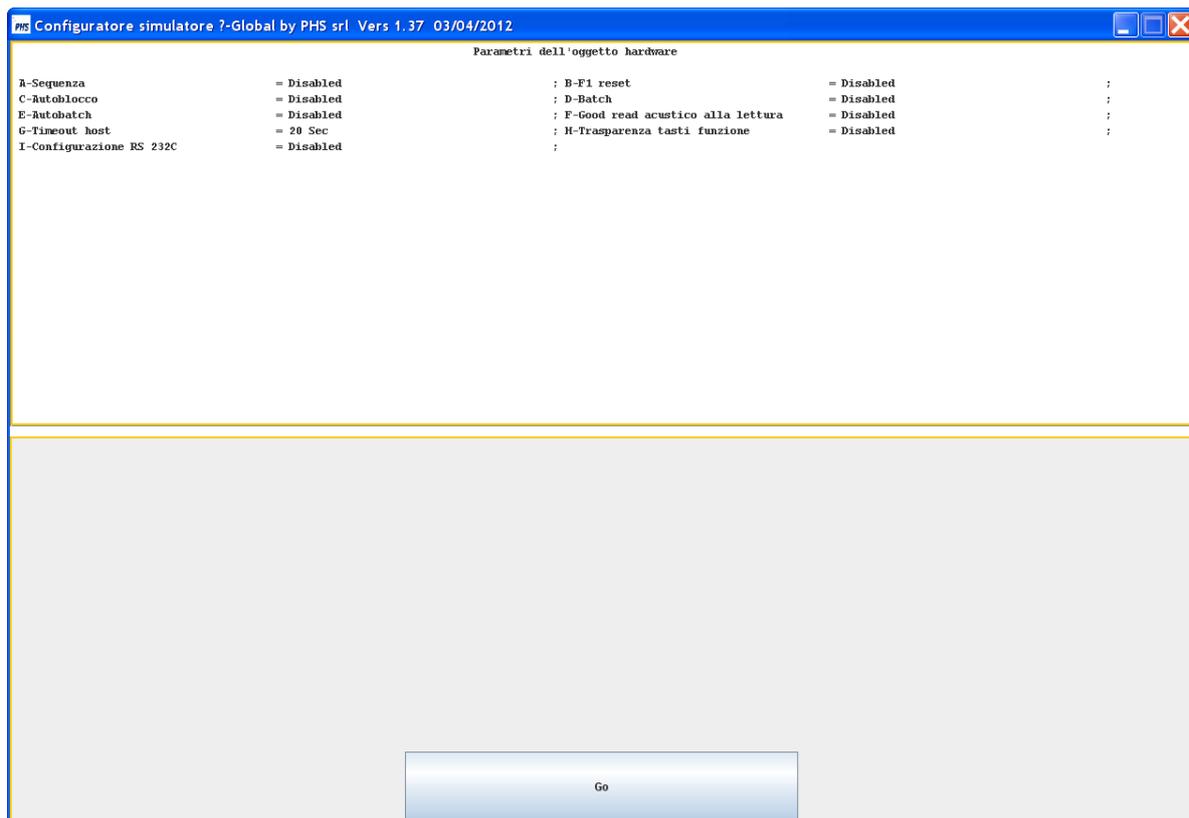
inputn toff = xxx; ton = yyy; repeat = zzz

dove **n** va da 1 a 16 a seconda del tipo di slave; **toff** e **ton** sono espressi in msec e rappresentano il tempo di apertura (o disattivazione) e di chiusura (o attivazione) del sensore; contrariamente al dispositivo fisico il simulatore non accetta tempi inferiori a 40 msec; **repeat** indica il numero di ripetizione della forma d'onda.

Global



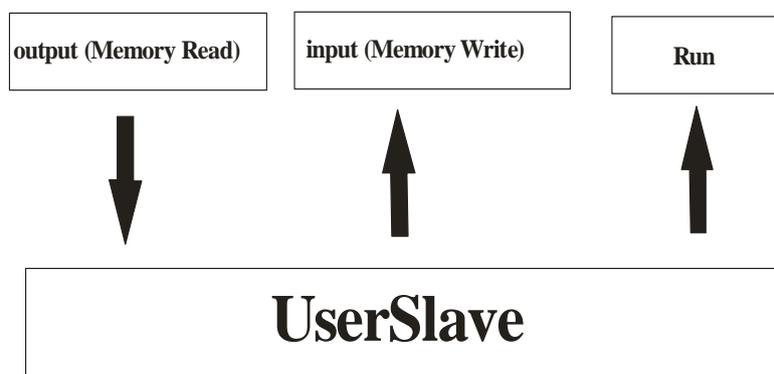
Attivando il bottone di configurazione si ottiene il pannello seguente:



Il bottone Go chiude il pannello.

UserSlave – plugin per il TTSSIM

Il TTSSIM dispone del plugin UserSlave che espone dei metodi per automatizzare e personalizzare la simulazione; il plugin legge il campo di output (o l'area di memoria), scrive sul campo di input (o sull'area di memoria) ed eventualmente attiva il bottone Run.



Il plugin UserSlave, per ogni oggetto hardware e in una struttura definita, rende disponibili delle classi secondo lo schema seguente:

1) RS 232C-COM1

- Classe **ApplicationRS232c**

- Metodi:

-**apRS232C**, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile;

-**memRespPLC**, è il metodo che restituisce il valore delle locazioni di memoria eventualmente richieste in **apRS232C**.

2) RS 232C-COM1 porta ausiliaria

- Classe **ApplicationRS232Aux**

- Metodo:

-**apRS232CAux**, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile.

3) RS 232C-COM2

- Classe **ApplicationRS232c1**

- Metodi:

-**apRS232C**, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile;

-**memRespPLC**, è il metodo che restituisce il valore delle locazioni di memoria eventualmente richieste in **apRS232C**.

4) RS 232C-COM2 porta ausiliaria

- Classe **ApplicationRS232Aux1**

- Metodo:

-**apRS232CAux**, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile.

5) Tastiera

- Classe **ApplicationKeyb**

- Metodo:

-**apKeyb**, è chiamato dal simulatore con una cadenza settabile, e può attivare la digitazione di una sequenza di tasti.

6) Barcode

- Classe **ApplicationBarcode**

- Metodo:

-apBarcode, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile e permette la creazione e l'attivazione di un menù barcode.

7) Badge

- Classe **ApplicationBadge**

- Metodo:

-apBarcode, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile.

8) Digital I/O

- Classe **ApplicationDigio**

- Metodo:

-apDigio, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile e permette di generare le forme d'onda dei segnali di input.

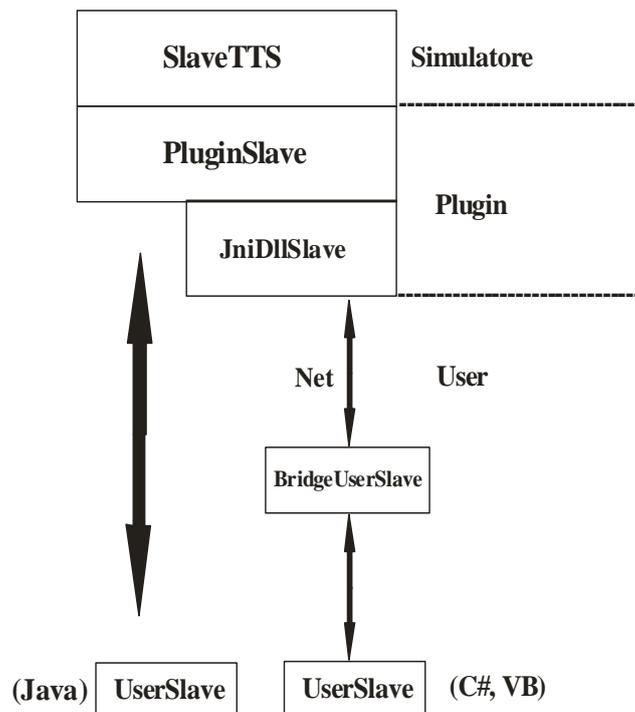
9) Bottoni

- Classe **ApplicationButton**

- Metodo:

-apButton, è il metodo che interfaccia i campi ed il bottone Run; esso è chiamato dal simulatore con una cadenza settabile e permette di attivare un singolo bottone.

Il plugin può essere sviluppato sia in Java che in .Net; vale lo schema seguente;



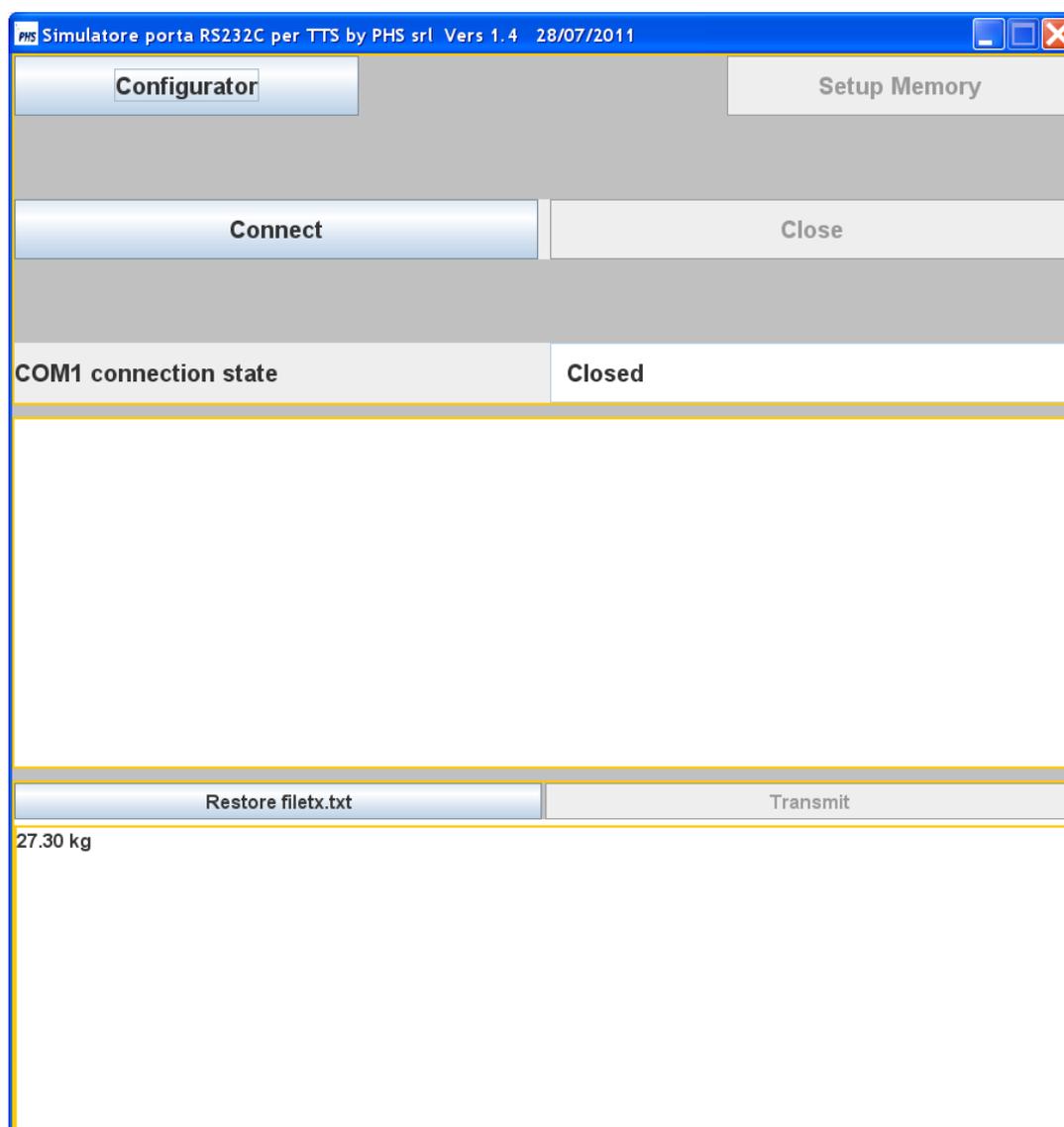
RS232SIM – Simulatore fisico dei protocolli TTS per RS 232C

Il simulatore si attiva tramite l'esecuibile rs232sim.exe contenuto nella cartella rs232sim, nella quale sono anche presenti i file:

confRSsim.xml (salva lo stato della simulazione corrente);

filetx.txt (setup di default dell' output della porta RS 232C).

Attivando il simulatore si ottiene il pannello:



esso è diviso in tre campi:

- 1) il primo permette di configurare,attivare e fermare la simulazione;
- 2) il secondo di input visualizza i caratteri ricevuti sulla porta RS 232C;
- 3) il terzo di output contiene i caratteri che saranno trasmessi sulla porta RS 232C.

Configurazione

Il bottone **Configurator** fornisce il pannello:



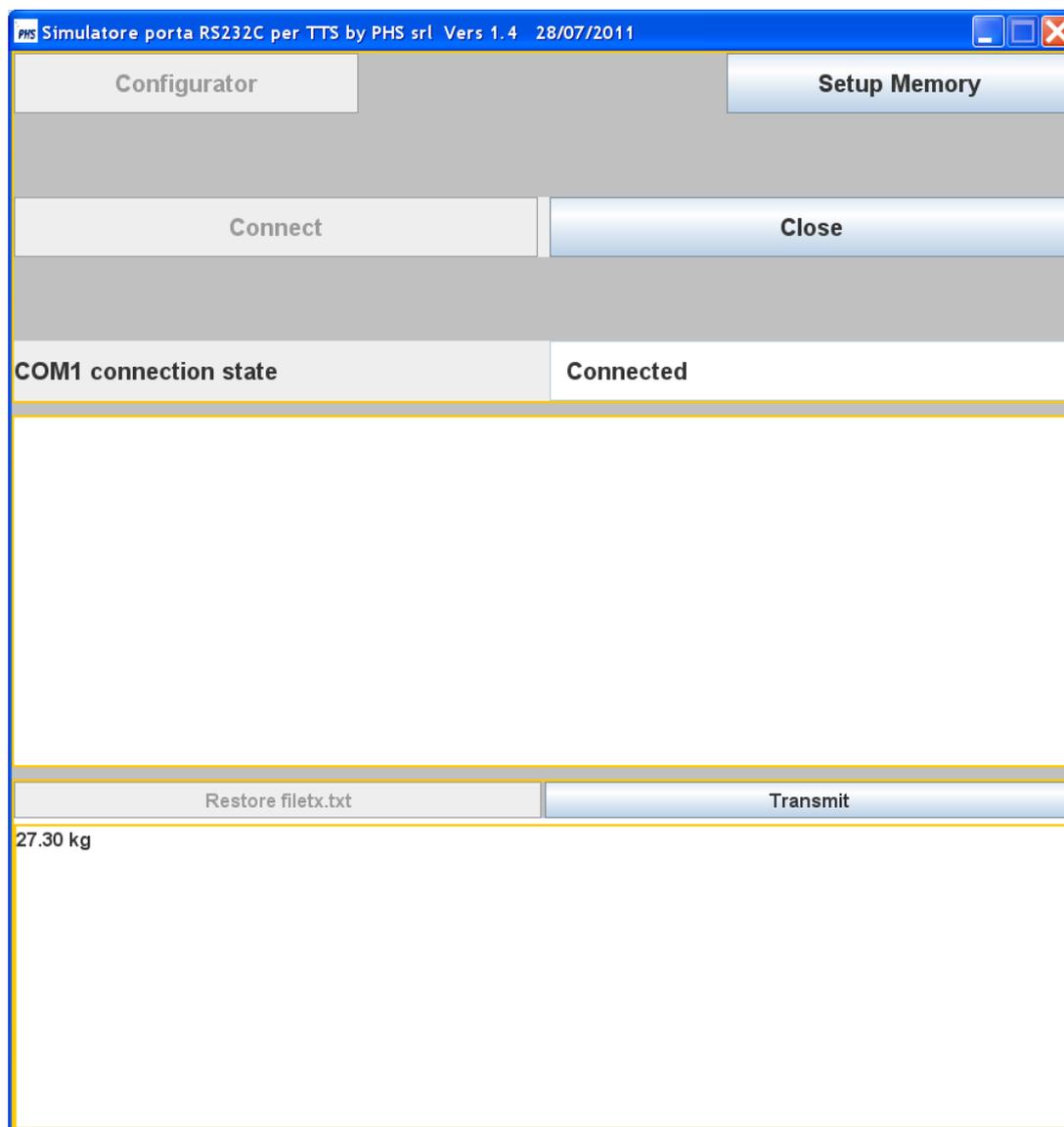
- 1) **TTS protocol**, il campo permette di selezionare uno dei protocolli supportati dalla porta seriale di uno slave TTS;



- 2) **Frame RS232C, Baud_Rate RS 232C e FlowControl**, permettono il classico setup dei parametri di una porta seriale;
- 3) **COM selection**, permette di selezionare la COM;
- 4) **Checksum**, permette di selezionare, tra quelli supportati, l'algoritmo di calcolo della checksum (o CRC) da utilizzare;
- 5) **Continuous Run**, mantiene il bottone Transmit sempre attivo, generando la trasmissione continua del contenuto del campo di output;
- 6) Il bottone **Save**, salva le impostazioni scelte e ritorna al pannello principale.

Attivazione e disattivazione

I bottoni **Connect** e **Close**, permettono di lanciare e fermare la simulazione; dopo la connessione il pannello assume l'aspetto seguente:



Se è previsto dal protocollo il bottone **Setup Memory** attiva il pannello di simulazione di una area di memoria di 64k (byte o word).

Utilizzo del bottone Setup Memory

Il valore di ogni locazione di memoria è esadecimale o ASCII a seconda del dispositivo da simulare (PLC,RFID, ecc.).

| | 0000 | 0001 | 0002 | 0003 | 0004 | 0005 | 0006 | 0007 | 0008 | 0009 | 000A | 000B | 000C | 000D | 000E | 000F |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0000 | BBBF | B97E | 3273 | BFA3 | 0203 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0010 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0020 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0030 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0040 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0050 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0060 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0070 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0080 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0090 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 00A0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 00B0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 00C0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 00D0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 00E0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 00F0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0100 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0110 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0120 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0130 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0140 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0150 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0160 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0170 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0180 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 0190 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 01A0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 01B0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 01C0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 01D0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 01E0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 01F0 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |

Setup from address (o Goto) Address = 0000

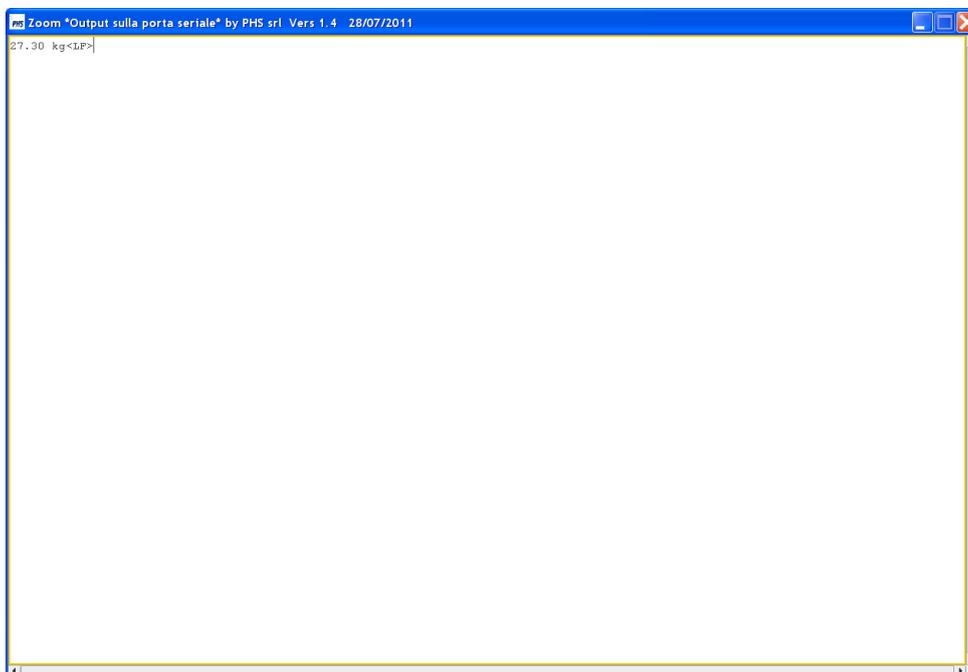
Fill all memory 00

UP Down Numero Blocchi in dec. 1

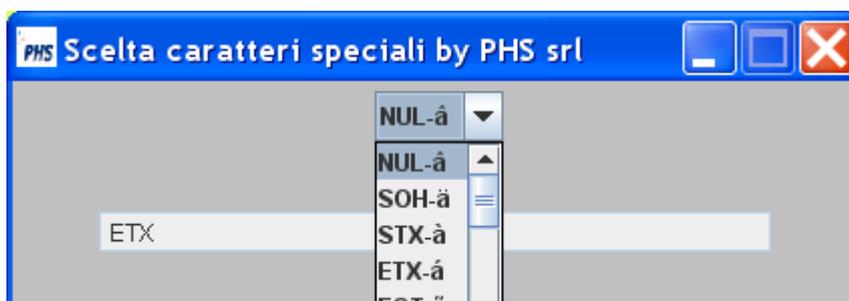
L'area di memoria è divisa in blocchi di 512 (byte o word) esadecimale o ASCII; le funzioni disponibili sono:

- 1) **Setup from address:**
 - a) nel campo Address si digita l'indirizzo di partenza (in esadecimale),
 - b) nel campo dati la sequenza di byte da settare (esadecimale o ASCII).
- 2) **Fill all memory** setup di tutta la memoria con la sequenza digitata.
- 3) I bottoni **UP** e **Down** permettono di scorrere la memoria a step di blocchi indicati nel campo relativo.

Per i campi di input ed output è possibile utilizzare, per comodità, (tramite il doppio click) un pannello di zoom per visualizzare o modificare il contenuto del campo, ove permesso, tramite la tastiera.

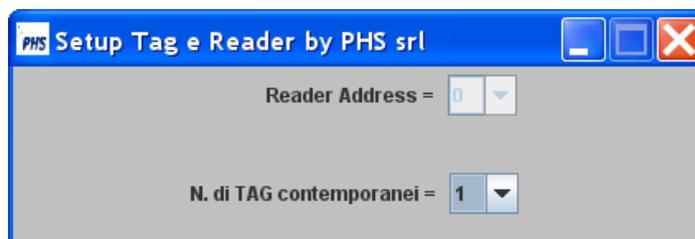


Per digitare caratteri speciali sul campo di output è possibile utilizzare, quando permesso, il pannello seguente, attivabile tramite il tasto destro.



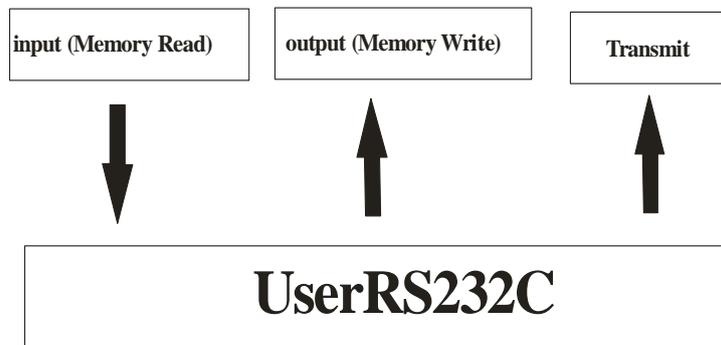
Nel campo dati sono visualizzati in sequenza tutti i caratteri speciali già presenti sul campo di output a cui si aggiungerà la nuova scelta.

Se si è presenza di un Reader RFID è possibile utilizzare il pannello seguente, attivabile tramite il tasto destro.



UserRS232C – plugin per lo RS232SIM

Lo RS232SIM dispone del plugin UserRS232C che espone dei metodi per automatizzare e personalizzare la simulazione; il plugin legge il campo di input (o l'area di memoria), scrive sul campo di output (o sull'area di memoria) ed eventualmente attiva il bottone Trasmit.



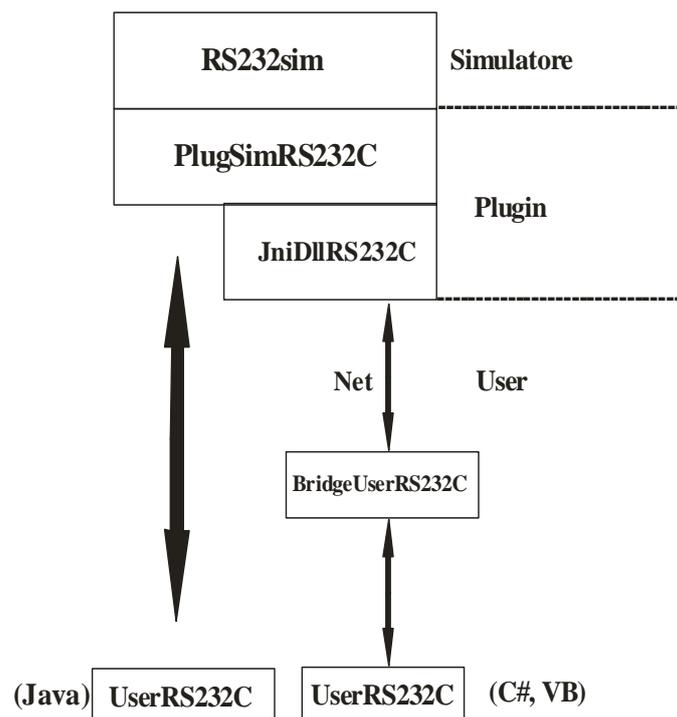
Il plugin UserRS232C, in una struttura definita, rende disponibile una classe secondo lo schema seguente:

- Classe **Application**

- Metodi:

- apRS232C**, è il metodo che interfaccia i campi ed il bottone Transmit; esso è chiamato dal simulatore con una cadenza settabile;
- memRespPLC**, è il metodo che restituisce il valore delle locazioni di memoria eventualmente richieste in apRS232C.

Il plugin può essere sviluppato sia in Java che in .Net; vale lo schema seguente:



Configurazione

Il bottone **Configurator** fornisce il pannello:

The screenshot shows a software window titled "Configurazione simulazione VC per TTS by PHS srl" with version "Vers 1.5" and date "31/10/2011". The window contains the following fields and controls:

- VC type:** A dropdown menu currently set to "VSC".
- VAC_Master IP Address:** A text input field containing "192.168.000.125".
- VAC_Port:** A dropdown menu currently set to "60000".
- VSC_protocol:** A dropdown menu currently set to "FINS OMRON".
- Continous Run:** A checkbox that is checked.
- Save parameter:** A button labeled "Save".

Esistono due tipi di canali virtuali VSC e VAC, selezionati tramite il campo **VC type**.

Se il canale virtuale è di tipo VSC il campo **VSC_protocol** permette la scelta del protocollo del dispositivo collegabile ad un master TTS30M.

This screenshot is identical to the previous one, but the "VSC_protocol" dropdown menu is open, showing the following options:

- FINS OMRON (highlighted)
- ISO on TCP S7
- MODBUS
- FINS OMRON
- Host Link OMRON
- RFID EMS DATALOGIC

Se il canale virtuale è di tipo VAC bisogna configurare:

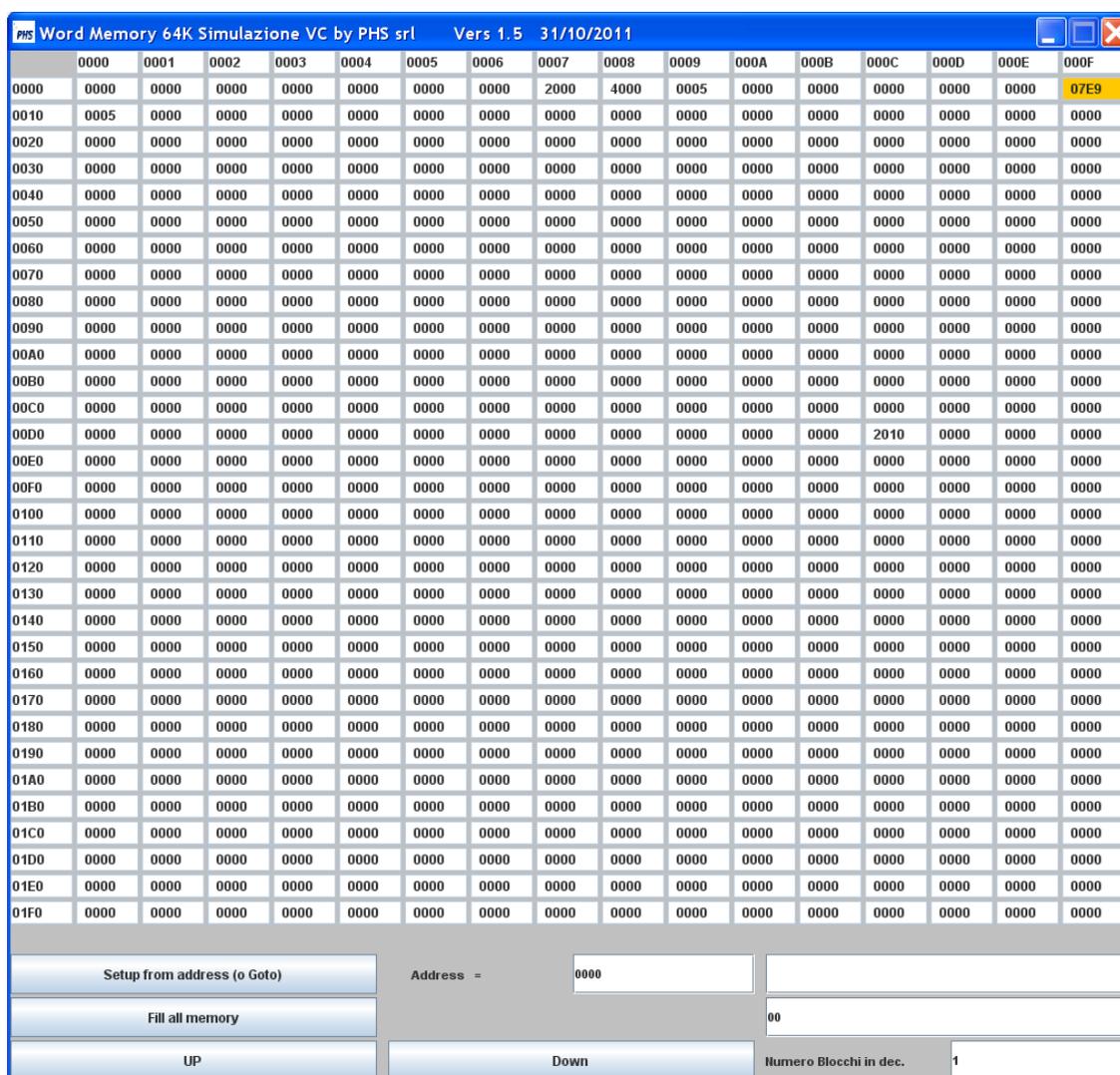
- 1) tramite il campo **VAC_Master IP Adress** l'indirizzo ip del master TTS30M di riferimento;
- 2) tramite il campo **VAC_Port** la porta sorgente del socket UDP prevista dal master TTS30M.

Il bottone **Continous Run**, mantiene quello di Transmit sempre attivo, generando la trasmissione continua del contenuto del campo di output.

Il bottone **Save**, salva le impostazioni scelte e ritorna al pannello principale.

Utilizzo del bottone Setup Memory

Il valore di ogni locazione di memoria è esadecimale o ASCII a seconda del dispositivo da simulare (PLC,RFID,ecc.).



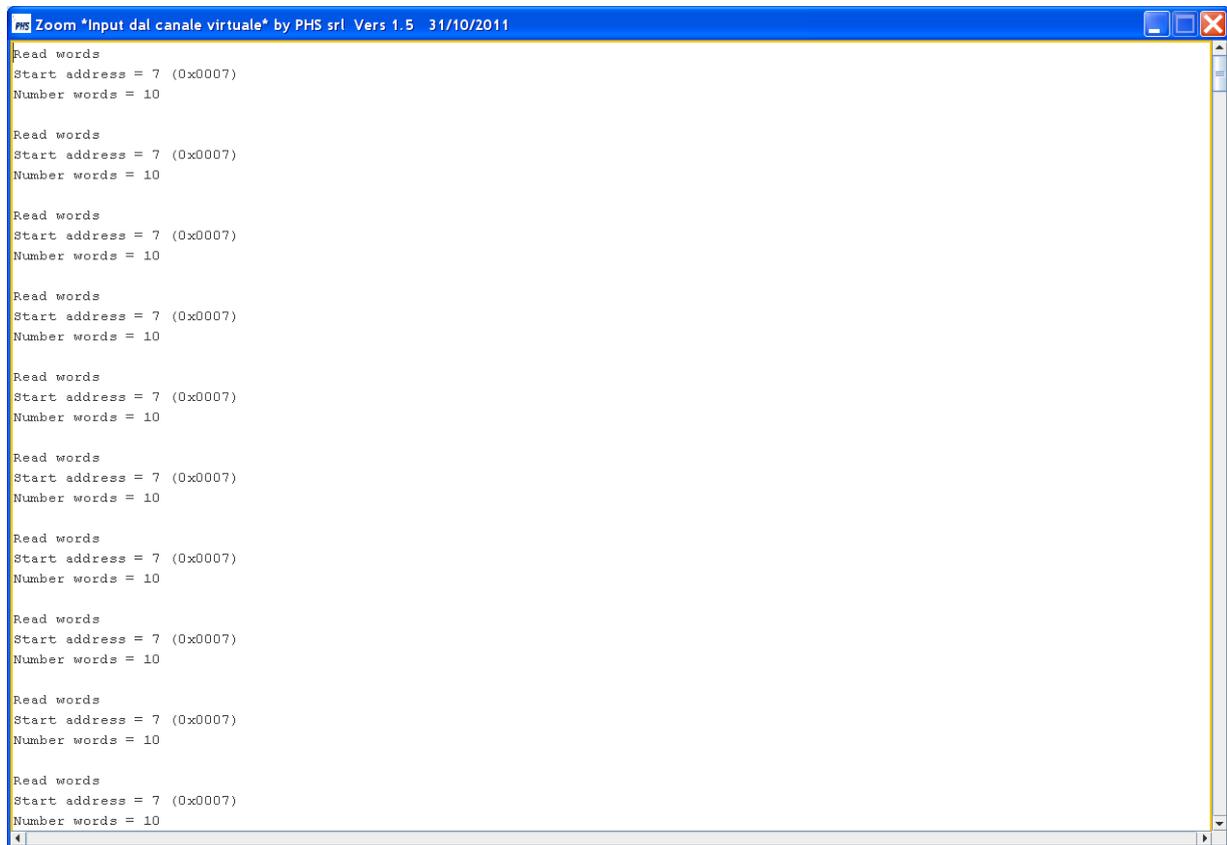
L'area di memoria è divisa in blocchi di 512 (byte o word) esadecimale o ASCII; le funzioni disponibili sono:

- 4) **Setup from address:**
 - c) nel campo Address si digita l'indirizzo di partenza (in esadecimale),
 - d) nel campo dati la sequenza di byte da settare (esadecimale o ASCII).
- 5) **Fill all memory** setup di tutta la memoria con la sequenza digitata.
- 6) I bottoni **UP** e **Down** permettono di scorrere la memoria a step di blocchi indicati nel campo relativo.

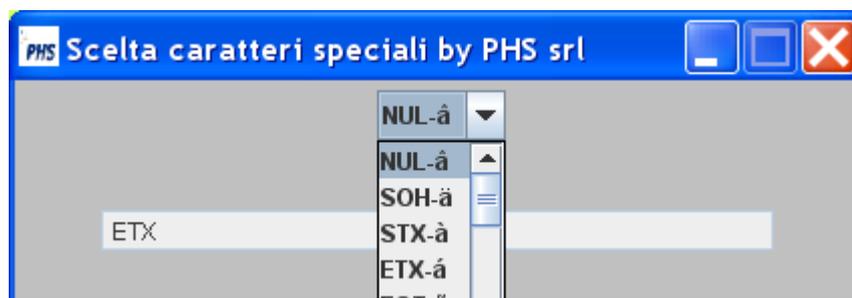
L'utilizzo della memoria cambia a secondo dei dispositivi simulati, valgono le note seguenti:

- **PLC S7 SIEMENS**; un DB è simulato con un blocco di 512 byte;
- **Reader RFID**; per la loro simulazione i byte (o word), pari alla lunghezza del TAG_ID, sono memorizzati a partire dall'indirizzo 0000; per cui all'indirizzo del primo byte (o word) di dato è sommato l'offset che si determina; per esempio per i TAG EPC l'offset è 6, poiché il TAG_ID occupa 6 word.
L'input GPI è simulato dall'indirizzo 65504 (0xFFE0); mentre l'output GPO dall'indirizzo 65520 (0xFFF0).

Per i campi di input ed output è possibile utilizzare, per comodità, (tramite il doppio click) un pannello di zoom per visualizzare o modificare il contenuto del campo, ove permesso, tramite la tastiera.



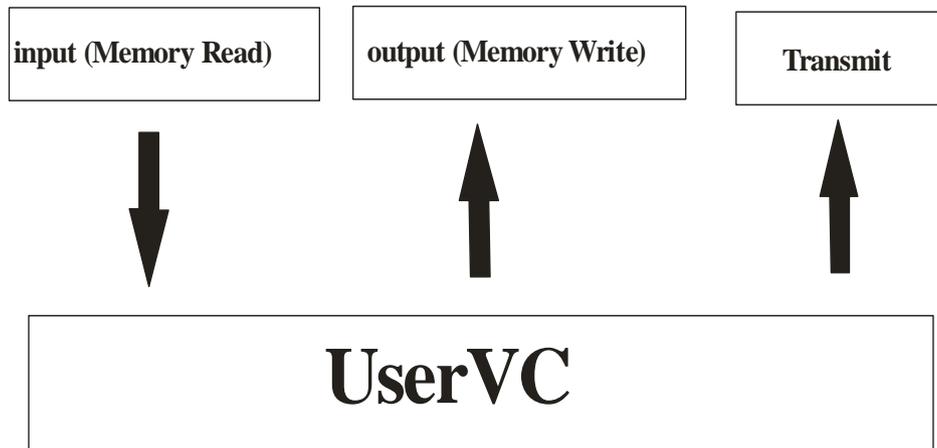
Per un canale VAC è possibile digitare caratteri speciali sul campo di output utilizzando il pannello seguente, attivabile tramite il tasto destro.



Nel campo dati sono visualizzati in sequenza tutti i caratteri speciali già presenti sul campo di output a cui si aggiungerà la nuova scelta.

UserVC – plugin per il VCSIM

Il VCSIM dispone del plugin UserVC che espone dei metodi per automatizzare e personalizzare la simulazione; il plugin legge il campo di input (o l'area di memoria), scrive sul campo di output (o sull'area di memoria) ed eventualmente attiva il bottone Trasmit.



Il plugin UserVC, in una struttura definita, rende disponibile una classe secondo lo schema seguente:

- Classe **Application**

- Metodi:

-**apVC**, è il metodo che interfaccia i campi ed il bottone Transmit; esso è chiamato dal simulatore con una cadenza settabile;

-**memRespPLC**, è il metodo che restituisce il valore delle locazioni di memoria eventualmente richieste in apVC.

Il plugin può essere sviluppato sia in Java che in .Net; vale lo schema seguente:

