

# NIDays

**FORUM TECNOLOGICO**  
SULLA PROGETTAZIONE GRAFICA DI SISTEMI

---

**SOLUZIONI E APPLICAZIONI 2012**

———— 19<sup>a</sup> edizione ► [nidays.it](http://nidays.it) ————

NIDays

NIDays

CONFERENCE



# Banco di test per Encoder di linea in ambito ferroviario

L. Magni<sup>1</sup>, F. Magnino<sup>1</sup>, S. Bacherini<sup>2</sup>, M. Tempestini<sup>2</sup>, L. Cipriani<sup>2</sup> – PRAGMA ENGINEERING<sup>1</sup>, GE TRANSPORTATION SYSTEMS<sup>2</sup>

## LA SFIDA

Realizzare un sistema di test automatico che consenta di effettuare il test funzionale di apparati LEU (Lineside Electronic Unit). Il sistema deve essere in grado di gestire scenari di test multipli verificando tutte le interfacce di comunicazione.

## LA SOLUZIONE

Adottare un'architettura di test (hardware e software) di nuova generazione (Next-Gen ATS) impiegando strumentazione virtuale su PXI e strumentazione tradizionale in abbinamento a moduli custom (booster) per l'interfacciamento (stimoli/risposte) del DUT, e l'engine multithreaded di NI TestStand per il controllo e la gestione dell'intero sistema.

Prodotti utilizzati	SwitchExecutive	Acquisizione Dati
TestStand	LabWindows/CVI	Seriali
LabVIEW	PXI/CompactPCI	Switches

Il banco di test sviluppato per apparati LEU (Lineside Electronic Unit) consente la validazione del dispositivo sia per quanto concerne i parametri elettrici delle interfacce che per quanto concerne gli aspetti funzionali dello stesso. L'architettura "ibrida" (composta da hardware COTS – Commercial off the Shelf - e hardware Custom) adottata è stata scelta al fine di soddisfare i requisiti delle sorgenti di stimolo mantenendo integro l'approccio di design Next-Gen. L'esecuzione di alcuni test ha richiesto l'implementazione di una modalità operativa basata su scenari in cui è possibile definire le condizioni di configurazione/stimolazione/risposta del DUT. A tale scopo è stato sviluppato uno specifico editor che consente la creazione/modifica di scenari sia in modalità integrata nel banco che stand-alone per la creazione off-line degli scenari. Il sistema descritto sfrutta l'architettura software basata su TestStand, SwitchExecutive, LabVIEW e LabWindows/CVI per implementare tutte le funzionalità richieste minimizzandone la complessità e massimizzandone la scalabilità, manutenibilità ed espandibilità.

Nell'ambito del settore dei trasporti ferroviari è di massima importanza l'affidabilità di tutti gli apparati preposti alla segnalazione di informazioni da e verso il treno. Tra i principali componenti che costituiscono il sotto sistema di terra (SST) vi sono gli Encoder o LEU (Lineside Electronic Unit), che hanno il compito, interfacciandosi direttamente con i segnali o con gli impianti del

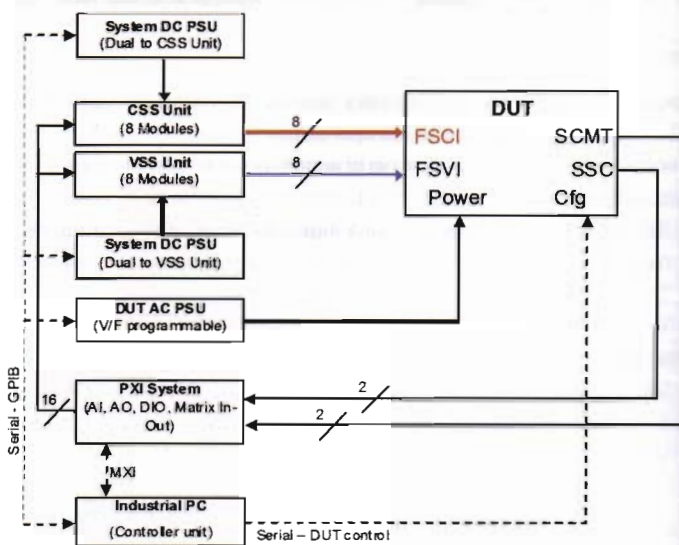


Figura 1: Architettura HW

European Train Control System), essenziale ai fini dell'interoperabilità per le linee ad alta velocità. Entrambe le interfacce rappresentano lo stato dell'arte in fatto di sistemi di ATP (Automatic Train Protection) concepiti per aumentare la sicurezza dell'esercizio ferroviario. Il collaudo di questo apparato, finalizzato alla sua verifica e

**“L'architettura software basata su TestStand, SwitchExecutive, LabVIEW e LabWindows/CVI ha permesso di implementare tutte le funzionalità richieste minimizzandone la complessità e massimizzandone la scalabilità, manutenibilità ed espandibilità.”**

segnalamento di cabina, di selezionare ed inviare le informazioni (telegrammi) contenenti dati significativi per la corretta condotta del treno. Pertanto, secondo una logica a stati ed in base a specifiche lookup-table, l'encoder associa ai segnali di ingresso, provenienti da lampade o indicatori presenti sulla linea, degli specifici bit-stream da trasmettere tramite le interfacce di cui è dotato.

Nel caso in esame il sistema di test deve consentire il collaudo di una nuova generazione di LEU in grado di supportare sia lo standard italiano SSC (Sistema Supporto Condotta), definito da RFI, che quello europeo SCMT (Sistema di Controllo Marcia Treno), definito dal ERTMS/ETCS (European Rail Traffic Management System/

validazione, richiede pertanto una accurata fase di design architetturale del sistema di test in cui il ruolo cardine, ai fini dell'implementazione del sistema stesso, è svolto principalmente dal software oltre che dalla strumentazione.

L'adozione di un'architettura hardware e software di Nuova Generazione (Next-Gen ATS) consente di soddisfare a pieno a tutti i requisiti di collaudo imposti sia dalle caratteristiche peculiari del DUT (hardware e funzionali) che dalle tipologie di test da effettuare. Questo tipo di approccio, oltre che a garantire il soddisfacimento delle necessità funzionali dei moderni sistemi di test (misure definibili dall'utente, elaborazione dei dati in real-time, interfacce





Figura 2: Vista ATE per LEU

MMI custom, etc.), deve consentire la rispondenza alle esigenze del laboratorio di Verifica e Validazione quali: flessibilità, prestazioni, qualità di misura, manutenibilità, espandibilità, riutilizzo, costo etc..

### Requisiti del sistema

Ai fini della validazione della LEU i requisiti più rilevanti del test prevedono:

- Test di alimentazione in AC (verifica dell'operatività del DUT a differenti tensioni e frequenze di rete)
- Test elettrico delle interfacce (SSC e SCMT)
- Verifica della validità del telegramma (rispondenza tra condizione di stimolazione e telegramma atteso)
- Verifica del tempo di commutazione del telegramma (passaggio tra due differenti condizioni di stimolazione)

Nella specifica realizzazione gli aspetti di maggiore impatto nella definizione dell'architettura sono stati quelli di: soddisfare le esigenze di stimolazione, con lo scopo di simulare le varie sorgenti di segnalamento di linea; gestire le differenti interfacce di trasmissione dati con lo scopo di decodificare i telegrammi trasmessi dall'encoder; eseguire i test in base a set di "scenari" definibili programmaticamente dall'operatore in cui ciascuno scenario è caratterizzato dall'insieme di condizioni di stimolazione e dai telegrammi attesi sulle specifiche interfacce oltre ad informazioni sulla configurazione del dispositivo.

### Architettura ATS

Per quanto concerne l'hardware, il sistema è organizzato in un'unità rack 19" ed è composto principalmente da 2 coppie di sorgenti di alimentazione in DC (impiego duale) rispettivamente asservite ai cestelli dei moduli booster CSS (in corrente) e VSS (in tensione), una sorgente in AC per l'alimentazione del DUT ed uno chassis PXI impiegato per tutte le funzionalità di acquisizione e generazione dei segnali analogici e digitali (AI, AO, DIO) e per il routing dei segnali di stimolo e di risposta (Matrix In-Out) (Figura 1).

L'architettura hardware è basata essenzialmente sull'impiego di un sistema PXI dotato di una scheda di acquisizione dati DAQ sincrona

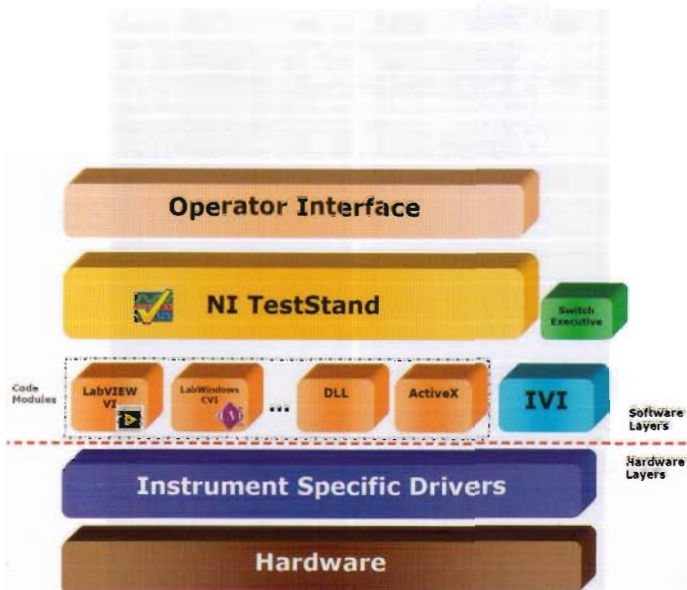


Figura 3: Visione di architettura SW

(per I/O analogico), di due schede multiplexer (per la selezione dell'interfaccia del DUT e l'instradamento dei segnali verso i moduli booster) e di una scheda Digital IO per il controllo del banco e di due cestelli contenenti un numero scalabile (da 1 a 8) di moduli booster appositamente progettati per fornire le opportune stimolazioni in corrente ed in tensione (Figura 2).

La progettazione dell'architettura software si basa su un test manager, quale TestStand, e l'impiego di ambienti di sviluppo (ADE) per il codice di test e l'interfaccia operatore quali LabVIEW e LabWindows/CVI. Per le necessità di switching si è impiegato uno specifico engine quale SwitchExecutive (NiSE) che consente di organizzare e gestire configurazioni di switch complesse rendendone immediato l'interfacciamento verso TestStand senza la necessità di sviluppare del codice dedicato (Figura 3).

### Implementazione

La progettazione custom si è resa necessaria al fine di rispondere alle esigenze di stimolazione del DUT simulando i segnali reali provenienti dalla linea ed in particolare si sono realizzati due tipi di moduli booster: modulo CSS in corrente in grado di generare segnali in corrente fino  $\pm 8A$  con 100KHz di banda; modulo VSS in tensione in grado di generare segnali in tensione fino  $\pm 300V$  con 3KHz di banda; entrambi i moduli ricevono in ingresso un segnale analogico, proveniente dalla scheda DAQ ed opportunamente instradato dalla matrice di uscita, e sono pilotati tramite linee digitali (Figura 4). Lo sviluppo software del banco è stato eseguito interamente ed in maniera nativa su TestStand ovvero concentrando tutto il flusso all'interno dello stesso minimizzando lo sviluppo di codice (negli ADE scelti) e soprattutto atomizzando il più possibile in modo da poter sostenere l'integrazione del codice unicamente nelle sequenze di test implementate su TestStand.

In particolare si è provveduto a sviluppare uno specifico "editor di scenario" al fine di consentire l'editing degli scenari di test per quanto concerne la validazione dei telegrammi e la verifica dei tempi di commutazione sia integrato nel banco che come applicativo stand-alone per un editing off-line.





Figura 4: Moduli booster in corrente ed in tensione

Per quanto concerne la generazione degli stimoli, il banco consente di sintetizzare i segnali come forme d'onda sinusoidali modulate in ampiezza (per simulare il lampeggiamento) con possibilità di aggiungere del rumore bianco, in alternativa possono essere impiegate forme d'onda arbitrarie, acquisite dalla linea, in formato HWS.

L'impiego di LabVIEW e LabWindows/CVI ha consentito di assolvere a tutte le necessità di acquisizione, elaborazione ed user interface richieste dal sistema; in particolare per la decodifica dei telegrammi di risposta trasmessi dai trasponder, per i quali sono state implementate delle specifiche routine per l'elaborazione che consentono la demodulazione, decodifica e la verifica del bit-stream e per l'implementazione di protocolli custom di gestione/configurazione del DUT.

L'impiego della tecnologia IVI ha consentito di gestire la strumentazione tradizionale (almeno per quanto concerne il PSU) in maniera nativa su TestStand senza ricorrere allo sviluppo di codice dedicato ed assicurando l'intercambiabilità futura di quei componenti senza ricorrere ad alcuna modifica del codice. L'impiego di SwitchExecutive ha permesso di "incapsulare" la gestione e la configurazione degli apparati di commutazione dentro il MAX (Measurement & Automation Explorer) svincolandola dal codice e rendendola accessibile in maniera nativa sempre da TestStand. In tal modo una variazione dell'hardware di switch o della sua configurazione non richiede sostanzialmente modifiche al codice o alla sequenza di test ma può essere gestita tramite una riconfigurazione a livello di MAX. Questa caratteristica è altresì maggiormente rilevante qualora si progetti il sistema in funzione di

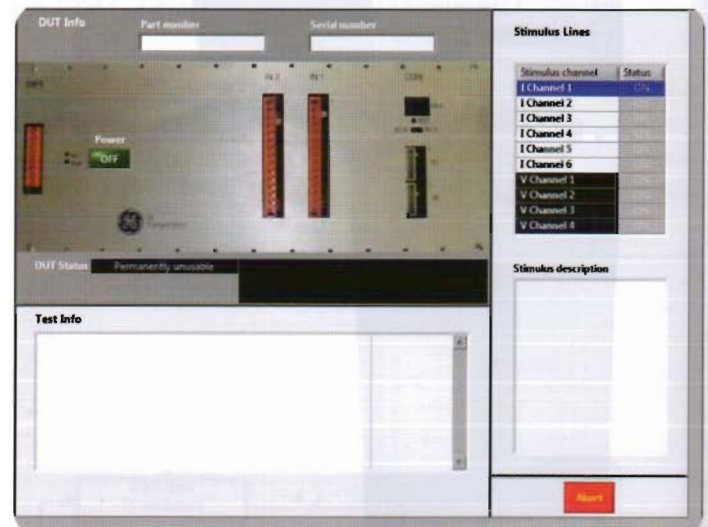


Figura 5: User interface (test-run)

una futura scalabilità dello stesso.

L'implementazione realizzata consente di eseguire le sequenze di test sia in modalità manuale (test singolo eseguito dall'operatore allo scopo di effettuare una verifica puntuale) che automatica (sessione di test).

In Figura 5 è mostrato il pannello della UI inerente l'esecuzione del test che mostra lo stato del DUT, il dettaglio delle informazioni inerenti il test in esecuzione e l'eventuale scenario in esame, nonché i dati inerenti le linee di stimolazione attive e le relative impostazioni.

## Conclusioni

Le soluzioni architetturali Next-Gen ATS basate su hardware e software di National Instruments hanno consentito un'applicazione "virtuosa" delle metodologie di progettazione, sviluppo e realizzazione del sistema di test rendendo possibile ottenere i migliori vantaggi offerti da questa scelta.

L'impiego di hardware custom come front-end di segnale (moduli booster) ha consentito di soddisfare sia i requisiti di stimolazione necessari al DUT che l'integrazione con l'hardware COTS mantenendo i benefici architetturali e consentendo l'ottimizzazione dei costi.

L'utilizzo integrato di TestStand combinato con IVI, NISE e gli ADE National Instruments ha reso efficace ed agevole lo sviluppo anche nell'implementazione di procedure di test complesse (scenari di test multipli) minimizzando lo sviluppo del codice e massimizzando l'indipendenza del codice dall'hardware sottostante (per ciò che è gestito via IVI e NISE).