

Scuola di Scienze
Corso di Laurea in Fisica

USB monitor in LabVIEW per il controllo
remoto delle schede DAQ del Pixel
Detector di ATLAS

Relatore:
Prof. Alessandro Gabrielli

Presentata da:
Martina Polito

Correlatore:
Dott. Luca Lama

Sessione I
Anno Accademico 2014/2015

Alla mia famiglia, per l'aiuto e il sostegno di questi anni.

*Agli amici di sempre, a quelli persi, e a quelli da poco
conosciuti.*

Sommario

L'esperimento ATLAS al CERN di Ginevra ha un complesso sistema di rivelatori che permettono l'acquisizione e il salvataggio di dati generati dalle collisioni di particelle fondamentali. Il rivelatore per cui trova una naturale applicazione il lavoro di questa tesi è il *Pixel Detector*. Esso è il più vicino alla beam pipe e si compone di più strati, il più interno dei quali, l'*Insertable B-Layer* (IBL), aggiunto in seguito allo shut down dell'LHC avvenuto nel 2013, ha apportato diverse innovazioni per migliorare la risoluzione spaziale delle tracce e la velocità di acquisizione dei dati. E' stato infatti necessario modificare il sistema di acquisizione dati dell'esperimento aggiungendo nuove schede chiamate ROD, *ReadOut Driver*, e BOC, *Back Of Crate*. Entrambe le due tipologie di schede sono montate su un apparato di supporto, chiamato *Crate*, che le gestisce. E' evidente che avere un sistema remoto che possa mostrare in ogni momento il regime di funzionamento del crate e che dia la possibilità di pilotarlo anche a distanza risulta estremamente utile. Così, tramite il linguaggio di programmazione LabVIEW è stato possibile progettare un sistema multiplatforma che permette di comunicare con il crate in modo da impostare e ricevere svariati parametri di controllo del sistema di acquisizione dati, come ad esempio la temperatura, la velocità delle ventole di raffreddamento e le correnti assorbite dalle varie tensioni di alimentazione. Al momento il software viene utilizzato all'interno dell'Istituto Nazionale di Fisica Nucleare (INFN) di Bologna dove è montato un crate W-Ie-Ne-R, speculare a quello presente al CERN di Ginevra, contenente delle schede ROD e BOC in fase di test. Il progetto ed il programma sviluppato e presentato in questa tesi ha ulteriori possibilità di miglioramento e di utilizzo, dal momento che anche per altri esperimenti dell'LHC le schede di acquisizione vengono montate sullo stesso modello di crate.

Indice

Introduzione	8
Capitolo 1 – LHC ed esperimento ATLAS	9
1.1 LHC	9
1.1.1 Principio di funzionamento dell’LHC	9
1.2 ATLAS	11
1.2.1 Rivelatori ATLAS	13
1.2.2 Spettrometro a Muoni	14
1.2.3 Calorimetri	15
1.2.4 Inner Detector	15
1.2.4.1 Transition Radiation Tracker	16
1.2.4.2 Semi Conductor Tracker	16
1.2.4.3 Pixel Detector	17
Capitolo 2 – Elettronica DAQ di IBL	19
2.1 IBL BOC	20
2.2 IBL ROD	20
Capitolo 3 – Crate W-le-Ne-R	23
3.1 Crate W-le-Ne-R 9U VME64x 6021	23
3.2 Standard VME	25
3.2.1 Descrizione del bus	26
3.3 Crate W-le-Ne-R con controllo remoto tramite CAN-bus	27
3.4 Protocollo CAN-bus	28
Capitolo 4 – Software LabVIEW di monitoraggio del Crate	36
4.1 Interfaccia NI USB-8473	36
4.2 Sviluppo del software di gestione del Crate W-le-Ne-R	37
4.2.1 Analisi delle funzioni	41
4.2.2 Analisi dei <i>Case</i>	43
4.2.3 Specifica dei requisiti software	43
Conclusioni e sviluppi futuri	56

Introduzione

Questa tesi nasce in seguito ad un progetto sviluppato nel laboratorio di progettazione elettronica dell'Istituto Nazionale di Fisica Nucleare INFN di Bologna dove vengono sviluppate e testate le schede per l'elettronica di acquisizione dell'esperimento ATLAS. Col passare del tempo si è reso necessario avere la possibilità di monitorare e controllare da remoto l'apparato che gestisce queste schede, chiamato *crate*. Per fare ciò è stato scritto un software in linguaggio LabVIEW capace, grazie all'interfaccia della *National Instrument* NI USB-8473, di comunicare con il *crate* e di ricevere informazioni sul suo stato, nonché di accenderlo e spegnerlo.

In questa tesi, dopo aver accennato al *Large Hadron Collider* (LHC), viene illustrato brevemente, nel primo capitolo, l'esperimento ATLAS e ciascun rivelatore di cui si compone. Nel secondo capitolo si parlerà dell'elettronica di acquisizione di ATLAS, e in particolare delle due schede che si occupano dell'elaborazione e della formattazione dei dati acquisiti da IBL (*Insertable B-Layer*), lo strato più recente e più prossimo alla beam pipe. Il capitolo tre è invece più approfondito ed è dedicato al contenitore delle schede, ossia al *crate* distribuito dall'azienda tedesca W-Ie-Ne-R, con l'aggiunta delle sue caratteristiche tecniche e degli standard per la comunicazione. In particolare verrà spiegato cos'è il protocollo CAN-bus, quali sono le sue caratteristiche e in cosa consistono i messaggi di richiesta e ricezione dei parametri. Infine nell'ultimo capitolo, a seguito di una breve introduzione sull'interfaccia NI USB-8473, verrà ampiamente descritto il software da me sviluppato. Verrà fatta un'analisi delle funzioni e degli oggetti che ne hanno permesso la costruzione, nonché dei *case*, ovvero delle strutture di cui si compone, e sarà anche presentata una scheda tecnica del software per il suo utilizzo o per eventualmente apportare delle modifiche. Infine verranno presentati gli sviluppi futuri di questo software, oltre alle sue possibilità di utilizzo.

Capitolo 1

LHC ed esperimento ATLAS

1.1 LHC

LHC (*Large Hadron Collider*) è attualmente il più grande e potente acceleratore di particelle al mondo. Lungo 27 km , è stato costruito all'interno di un tunnel sotterraneo posto a circa 100 m di profondità lungo il confine franco-svizzero, precisamente fra l'aeroporto di Ginevra e il Massiccio del Giura, ed è gestito dal Centro Europeo per la Ricerca Nucleare (CERN). Originariamente il tunnel ha ospitato il LEP (*Large Electron-Positron collider*), acceleratore di particelle elementari smantellato nel 2000 per lasciare posto all' LHC.

1.1.1 Principio di funzionamento dell'LHC

All'interno dell'acceleratore due fasci di particelle ad alta energia, composti da protoni ottenuti dalla ionizzazione di atomi di idrogeno gassoso o da ioni di piombo, vengono fatti viaggiare a velocità prossime alla velocità della luce, per poi collidere. L'energia di progettazione per le collisioni è di 14 TeV nel centro di massa. I fasci viaggiano in direzione opposte all'interno di due tubi a vuoto paralleli, detti *beam pipe*, per poi scontrarsi in quattro diversi punti di interazione. Lungo il loro tragitto nell'anello principale, vengono guidati da un forte campo magnetico generato da 1232 dipoli magnetici superconduttori. Questi ultimi sono composti da speciali bobine che operano in stato di superconduzione,

conducendo elettricità senza resistenza o perdita di energia. Per fare ciò i magneti devono essere raffreddati fino a circa 1.9 K , temperature raggiunte grazie al condotto di refrigerazione attraverso il quale viene fatto scorrere elio superfluido. Il condotto non serve solo a raffreddare i magneti, ma anche gran parte dei sistemi che compongono l'acceleratore.

Prima di raggiungere l'anello principale, i fasci di protoni devono essere accelerati per poter avere già una elevata energia all'interno dell'LHC. Essi, infatti, devono prima attraversare quattro pre-acceleratori che portano i fasci ad una energia di 450 GeV .

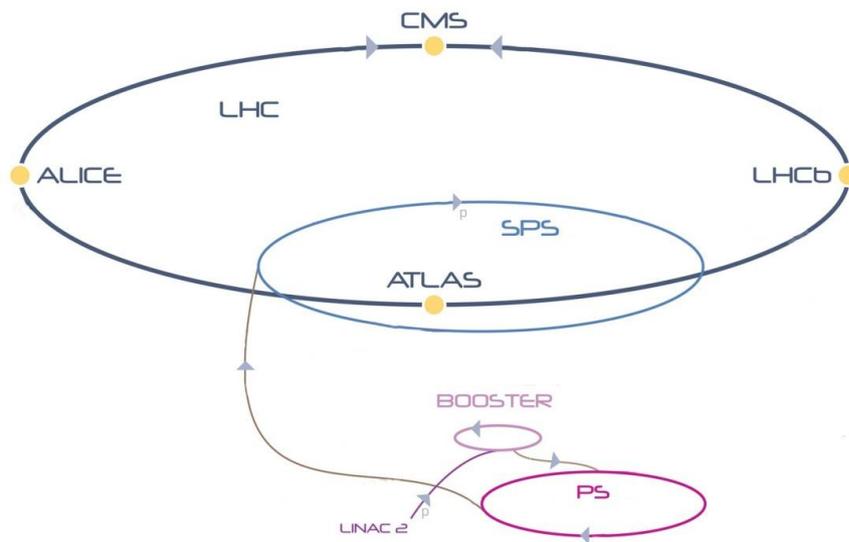


Fig. 1.1 Complesso di acceleratori del CERN.

Il complesso di acceleratori consiste in una serie di apparecchiature che accelerano le particelle ad energie sempre maggiori. Il primo della catena è l'acceleratore lineare Linac 2, che porta i fasci ad una energia di 50 MeV ; successivamente questi ultimi attraversano tre sincrotroni: il primo, il *Proton Synchrotron Booster* PSB, accelera i protoni a 1.4 GeV ; il secondo, il *Proton Synchrotron* PS, spinge il fascio a 25 GeV ; infine, l'ultimo sincrotrone, il *Super Proton Synchrotron* SPS, fa raggiungere ai protoni l'energia di 450 GeV che porta il fascio nell'anello principale.

Come precedentemente accennato, i fasci di protoni collidono in quattro punti di interazione localizzati lungo l'anello principale, che corrispondono ciascuno ad un diverso rivelatore di particelle. I quattro esperimenti legati a ciascun rivelatore sono: ATLAS (*A Thoroidal Lhc ApparatuS*); ALICE (*A Large Ion Collider Experiment*); CMS (*Compact Muon Solenoid*); LHCb (*Large Hadron Collider beauty*).

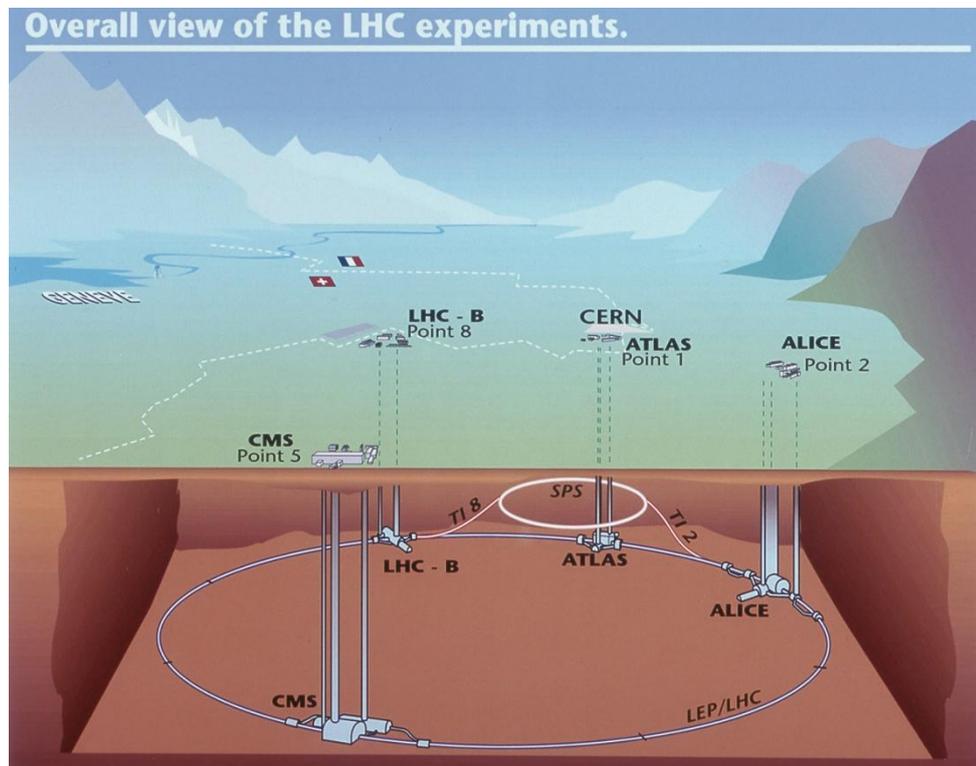


Fig. 1.2 Panoramica dell'LHC con i relativi esperimenti

1.2 ATLAS

L'apparato sperimentale ATLAS consiste in un rivelatore cilindrico *general purpose* (Fig. 1.3) lungo circa 44 m, alto più di 25 m, e del peso complessivo di circa $7 \cdot 10^6$ kg. Esso si compone di più sotto-rivelatori posti in maniera simmetrica, ciascuno dei quali ha il compito di acquisire dati risultanti da diverse particelle. Le quattro parti principali in cui può essere suddiviso sono:

- *Inner Detector*, rivelatore interno per il tracciamento delle particelle cariche e la misura dei vertici di interazione e dei vertici secondari;
- Sistema di Magneti, per misurare l'impulso delle particelle cariche tramite una deflessione della traiettoria;
- Calorimetri elettromagnetico ed adronico per la misura di elettroni, fotoni e jet di adroni;
- Spettrometro a muoni, per l'identificazione e la misura dell'impulso dei muoni penetranti.

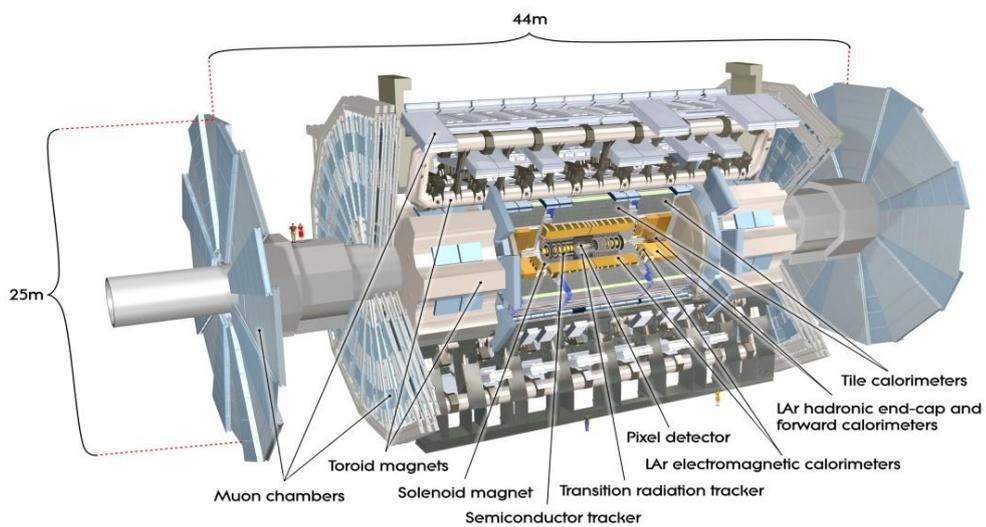


Fig. 1.3 Sezione del rivelatore ATLAS

La localizzazione di un punto è effettuata utilizzando le coordinate polari in cui l'origine delle coordinate risiede nel punto di collisione dei fasci. L'asse z è lungo la direzione del fascio, l'angolo θ indica l'angolo polare misurato a partire dalla direzione del fascio e l'angolo φ è l'angolo azimutale misurato sul piano perpendicolare all'asse z .

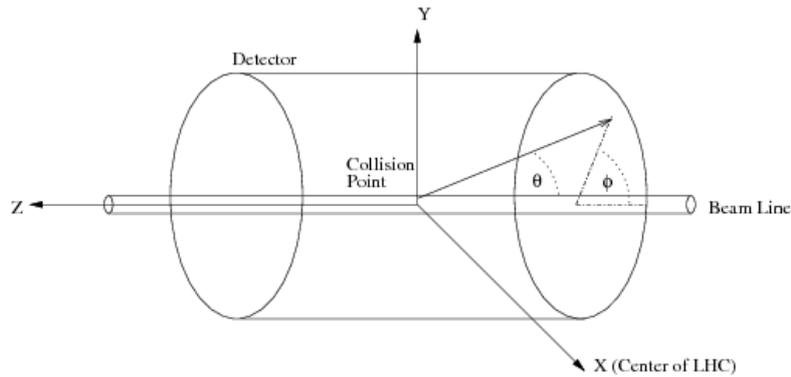


Fig. 1.4 Coordinate spaziali rivelatore ATLAS

1.2.1 Rivelatori ATLAS

Per il funzionamento del detector è necessario un grande campo magnetico uniforme che viene creato mediante l'utilizzo di magneti superconduttori che avvolgono il rivelatore. Le geometrie del sistema di magneti sono una solenoidale e una toroidale. La prima consiste in un grande solenoide di $5,3\text{ m}$ di lunghezza e $2,4\text{ m}$ di diametro, formato da fili superconduttori che garantiscono un campo di 2 T d'intensità, con un massimo di $2,6\text{ T}$. La struttura toroidale conta due diverse componenti presenti in figura Fig. 1.5 a) e b) : un *Barrel Toroid* e due *End Cap Toroid*. Entrambi sono formati da 8 bobine a superconduttore, inclinate di 45° l'una rispetto all'altra, che operano a temperature di circa 4 K , facendo circolare una corrente di 20 kA che produce un campo magnetico di 4 T circa. Il Barrel Toroid si sviluppa attorno alla beam pipe per $25,3\text{ m}$, con un'estensione radiale che va dai $9,4$ ai $20,1\text{ m}$. Gli End Cap Toroid hanno invece una lunghezza di 5 m ed un'estensione radiale che va da $1,65$ a $10,7\text{ m}$.

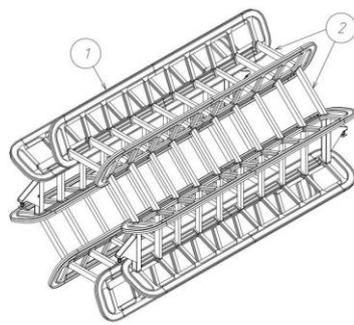
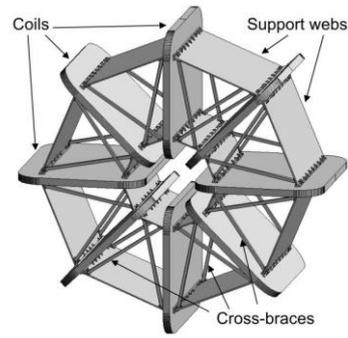


Fig. 1.6 a) Barrel Toroid



b) End Cap Toroid

La struttura interna e lo schema di funzionamento del detector ATLAS sono presenti in Fig. 1.7.

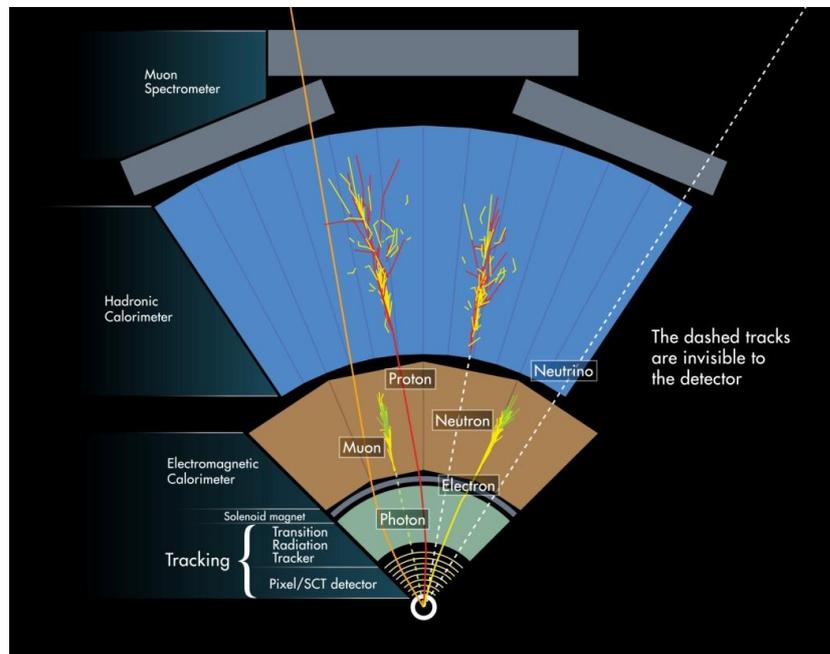


Fig. 1.7 Schema di funzionamento del rivelatore

1.2.2 Spettrometro a Muoni

L'involucro più esterno del detector ATLAS è lo spettrometro a muoni. I magneti toroidali generano un campo magnetico grazie al quale lo spettrometro può misurare la deflessione di muoni e misurarne l'impulso con una risoluzione che va

dal 3% al 12% per valori di energia compresi tra 10 GeV e 10^3 GeV . E' composto da due sistemi: camere di trigger, per misurare rapidamente l'impulso dei muoni, e camere di tracciamento ad alta precisione, per una misura accurata, in tempi più lunghi, dell'impulso. I muoni sono le uniche particelle che raggiungono lo spettrometro. Questo è dovuto al fatto che le particelle in considerazione sono estremamente massive, $m_{\mu^-} \cong 105,7 \frac{\text{MeV}}{c^2}$, e quindi non creano lo sciame elettromagnetico caratteristico di particelle con masse inferiori, e attraversano i calorimetri perdendo una piccola frazione della loro energia.

1.2.3 Calorimetri

I due calorimetri presenti nel detector ATLAS sono il calorimetro adronico e il calorimetro elettromagnetico. Entrambi i calorimetri sono composti da strati di materiale pesante, usati come convertitore, alternati a strati di materiale attivo. In entrambi i calorimetri l'Argon liquido è usato come materiale attivo, mentre per convertitore è usato il piombo nel calorimetro elettromagnetico, e ferro, rame e tungsteno, a seconda della posizione, nel calorimetro adronico. Quando un elettrone o un fotone ad alta energia incide su un materiale come quelli sopra indicati, la particella genera uno sciame elettromagnetico composto dalle stesse particelle della fonte. L'energia della particella originale si deposita nel materiale sotto forma di energia di ionizzazione, rivelando la quale si può stabilire quale fosse la particella sorgente e la sua energia di partenza. Generalmente elettroni e fotoni vengono direttamente assorbiti nel calorimetro elettromagnetico, mentre quello adronico è raggiunto solo dalle particelle adroniche. L'energia di queste ultime, infatti, viene assorbita solo in piccolissima parte dal calorimetro elettromagnetico.

1.2.4 Inner Detector

L'*Inner Detector* (ID) è il rivelatore più vicino alla beam pipe, ed è stato progettato per ricostruire le tracce delle particelle cariche e misurarne le

caratteristiche (momento, momento trasverso e vertice). Ha un raggio esterno di 115 cm ed è lungo 6.2 m . È immerso in un campo magnetico solenoidale parallelo all'asse del fascio dell'intensità di 2 T necessario per il calcolo del momento. Il rivelatore interno si compone di tre rivelatori.

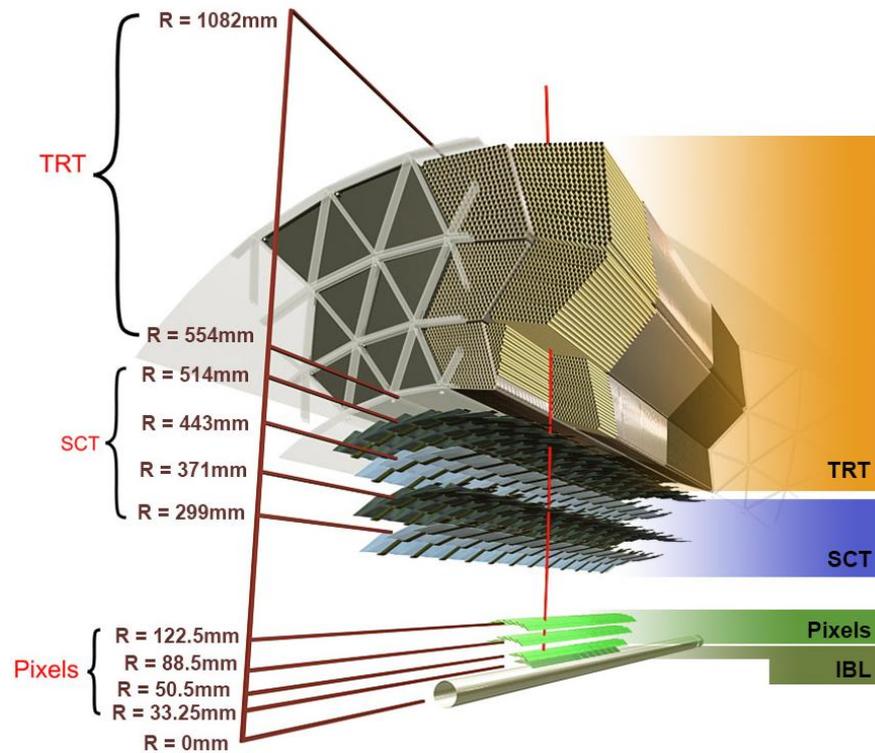


Fig. 1.8 Struttura interna Inner Detector

1.2.4.1 Transition Radiation Tracker

Partendo dall'esterno, il primo rivelatore che si incontra è il *Transition Radiation Tracker* (TRT), che utilizza strati di rivelatori a forma di tubo con un diametro di 4 mm . Esso rivela la radiazione di transizione, ovvero la radiazione che viene emessa da una particella carica al passaggio attraverso un mezzo non omogeneo.

1.2.4.2 Semi Conductor Tracker

Subito dopo il TRT si trova il *Semi Conductor Tracker* (SCT), un detector costruito da strisce di rivelatori con tracciatori al silicio. **Ciascuna** rivelatore al silicio ha un'area di $6.36 \times 6.40\text{ cm}^2$ con 780 strisce di readout.

1.2.4.3 Pixel Detector

L'ultimo rivelatore, ovvero il più vicino alla beam pipe, è il *Pixel Detector*. Quest'ultimo è stato progettato con tecnologie più avanzate rispetto ai primi due rivelatori dell'Inner Detector poiché deve sopportare maggiori radiazioni elettromagnetiche durante l'acquisizione dei dati. Esso era originariamente diviso in tre strati denominati, dal più esterno al più interno, *Layer 2*, *Layer 1*, *Layer B0*. Ciascuno strato è formato da strutture meccaniche che ospitano i rivelatori. Tali strutture sono dette *stave*, e su ciascuna di esse vi sono 13 moduli formati, a loro volta, da componenti che contribuiscono, qualora si verifichi un evento, alla raccolta dei dati e al loro invio alle schede ROD, *ReadOut Drivers*. Come gli altri rivelatori è a simmetria cilindrica e sono presenti tre *End Cap* per lato.

Dopo lo spegnimento dell'LHC, avvenuto il 13 febbraio 2013, grazie a lavori di upgrade sono stati raggiunti valori di energia sempre maggiori e sono state apportate migliorie alla struttura. Una di queste è stata l'aggiunta di un nuovo layer all'interno del Pixel Detector chiamato *Insertable B-Layer*, IBL. Esso è stato inserito fra il Layer B0 e la beam pipe, opportunamente ridimensionata. Le ragioni per le quali è stato aggiunto tale strato sono diverse. Le più rilevanti sono le seguenti:

- I Layer 2, 1 e B0 soffrono di mortalità di pixel dovuta alla forte esposizione a radiazioni. Si è inoltre stimato che il *B-Layer* con le attuali energie e quindi con luminosità dell'ordine di $10^{34} \text{cm}^{-2} \text{s}^{-1}$ degradi fino a rendersi inutilizzabile nell'arco di 3 anni. Questa inefficienza causa una perdita consistente nella funzionalità del sistema, la quale viene ripristinata pienamente dall'aggiunta dell'IBL.

- Limiti di banda.

Aumentando l'energia di collisione aumenta anche la luminosità, un parametro che esprime il numero di collisioni che possono essere prodotte nell'unità di tempo in unità di superficie efficace.

$$L = n_b \frac{N_1 \cdot N_2 \cdot f}{\Sigma_{\text{efficace}}}$$

Dove N_1 ed N_2 sono il numero di particelle del pacchetto (*Bunch*) considerato, n_b numero di bunch, Σ_{eff} la superficie efficace per la collisione che dipende dalla sezione d'urto σ dei fasci di particelle secondo la relazione:

$$\Sigma_{efficace} = 4 \pi \sigma_1 \sigma_2$$

I vecchi B-layers non sono in grado di supportare l'aumento del flusso di dati dovuto appunto all'aumento di luminosità.

- Vi è infine una maggiore precisione nel tracciamento, in quanto pixel più vicini al punto di interazione aumentano fortemente la sensibilità del detector rispetto alle collisioni.

Per ottenere una maggiore sensibilità nell'IBL sono stati utilizzati dei chip di *Front End* FE-I4, garantendo così una maggiore area sensibile rispetto al più obsoleto FE-I3, montato tutt'ora negli strati più esterni del Pixel Detector.

Capitolo 2

Elettronica DAQ di IBL

Il sistema di acquisizione dati dell'esperimento ATLAS è piuttosto elaborato e richiederebbe un'approfondita analisi. Tuttavia l'argomento non verrà che introdotto e descritto solo superficialmente in questa tesi.

I dati acquisiti dai chip di Front End di IBL seguono un percorso piuttosto semplice. Dai 32 chip di Front End FE-I4 partono 16 linee seriali verso la scheda BOC, *Back Of Crate*, tramite collegamento ottico. Dalla BOC i dati vengono parallelizzati in 8 bus da 12 bit ciascuno e trasferiti alla scheda ROD, *Read Only Driver*. Di lì i dati vengono incapsulati in un *frame* chiamato *Rod Data Frame*, che viene poi stoccato nei computer dedicati. Dalla scheda ROD i frame possono seguire due diversi instradamenti: nel primo i dati formattati vengono rispediti alla scheda BOC e in seguito spediti in maniera seriale tramite connessione ottica ai computer di stoccaggio; il secondo, utilizzato per la calibrazione, invia direttamente i dati ad un PC senza ritornare alla scheda BOC. In Fig. 2.1 si può vedere direttamente il sistema di acquisizione utilizzato da IBL.

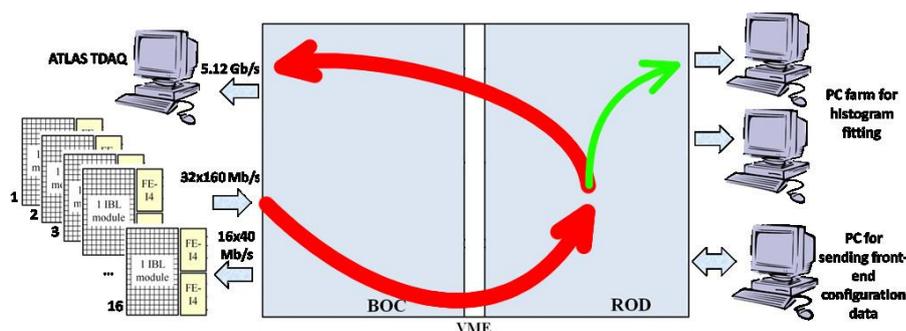


Fig. 2.1 Sistema di acquisizione di IBL

2.1 IBL BOC

Tre Xilinx Spartan 6 FPGA, *Field Programmable Gate Array*, hanno il compito di controllare la BOC e di elaborare i dati. La *BOC Control FPGA*, detta BCF, è responsabile della gestione dello standard ethernet e della comunicazione con altri dispositivi. La *BOC Main FPGA*, detta BMF, è composta da due dispositivi Spartan 6 e si occupa di ricevere i dati dai chip FE-I4, di elaborarli e di inviarli alla scheda ROD. Infine la BOC ha il compito di inoltrare un clock a 40 MHz a tutto il sistema di acquisizione.

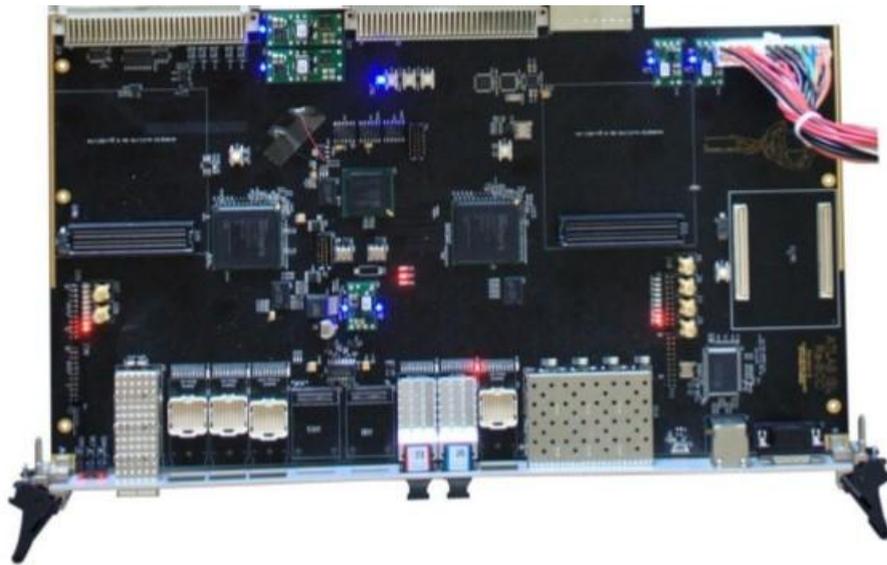


Fig. 2.2 Scheda BOC

2.2 IBL ROD

La scheda ROD, *Read Only Driver*, collega la catena di acquisizione dati standard di ATLAS con gli specifici componenti contenuti all'interno del rivelatore. Nello specifico i compiti di questa scheda sono:

- Propagare i segnali di trigger ai chip di Front End;
- Inviare le giuste configurazioni ai chip FE-I4 connessi;

- Ricevere i frammenti di un evento dai 32 chip di Front End per poi compattarli e rispedirli alla BOC.

Con l'implementazione dei chip FE-I4 le velocità di trasmissione dei dati sono incrementate notevolmente, ed è stato necessario sostituire la vecchia scheda SiROD con una nuova ROD capace di sostenere il flusso di dati da elaborare. Parallelamente anche la scheda BOC è stata sostituita così da essere compatibile con la nuova ROD. L'attuale sistema BOC-ROD è in grado di gestire 32 chip Front End, ed è inoltre capace di interfacciarsi con i vecchi FE-I3 grazie all'interfaccia VME che è rimasta sulla ROD.

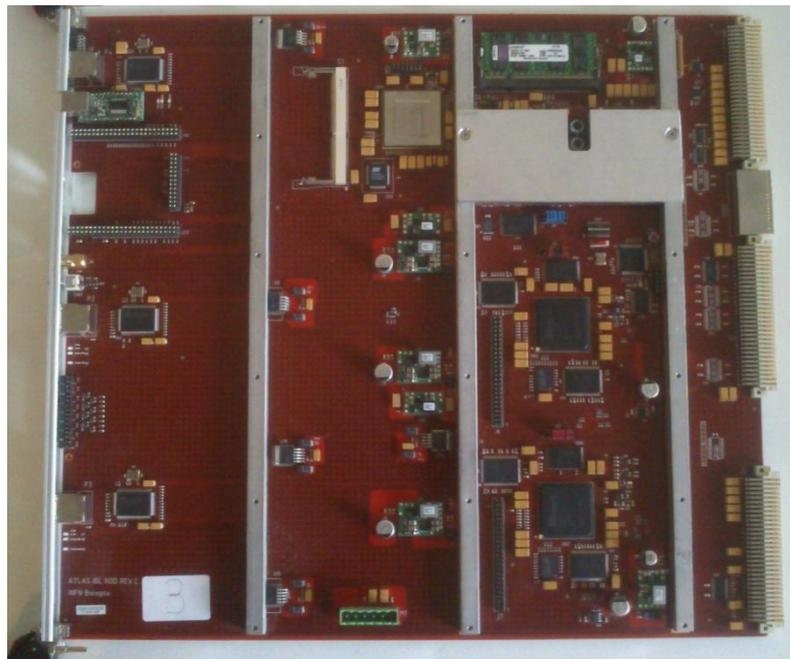


Fig. 2.3 Scheda ROD

Sulla scheda vi sono 14 layer, nei quali sono presenti i seguenti componenti:

- SDRAM DDR 32 MB;
- Memoria FLASH da 64 Mbit Atmel AT45DB642D;
- 2 GB DDR2 SODIMM;
- Memoria FLASH da 4 Mbit SST39VF040-70-4C-NH;

- Interfaccia Ethernet 3 Gbit DP83865;
- Processore digitale TMS320C6201-GJC200;
- FPGA Xilinx Spartan-6 XC6SLX45-FGG484. Si tratta di un dispositivo *PRM (Program Reset Manager)* che permette alla *ROD* di venire pilotata tramite interfaccia *VME*. Si occupa delle operazioni di reset e della programmazione della scheda stessa;
- FPGA Xilinx Virtex-5 XC5VFX70T-FF1136. È il controller della *ROD* e contiene un core PPC all'interno;
- 2 FPGA Xilinx Spartan-6 XC6SLX150-FGG900. Si occupano dell'elaborazione dei dati all'interno della *ROD*. Ad ognuna di queste FPGA è collegata una memoria statica più veloce utilizzata come cache e una memoria dinamica DDR2 più lenta per l'elaborazione effettiva dei dati.

Infine sul pannello frontale della *ROD* sono presenti le seguenti interfacce per il collegamento esterno:

- Porta USB;
- Connettore JTAG per la programmazione delle FPGA;
- Pulsante di reset;
- 3 Porte Gbit Ethernet;
- 8 status led.

Per collegare la scheda BOC alla *ROD* vi sono 96 fili, 8 bus di 12 *bit* ciascuno, divisibili in 12 gruppi che gestiscono i dati provenienti dai chip Front End. Ciascun gruppo è formato da: un singolo bit attivo alto, detto *Write Enable*, che controlla la validità dei dati correnti; un bit attivo alto di controllo che etichetta i dati entranti come parola di controllo; due bit di indirizzo che indicano il canale che ha generato i dati correnti; 8 *bit* che costituiscono una parte dell'output dei chip FE-I4.

Capitolo 3

Crate W-Ie-Ne-R

Come precedentemente accennato le schede BOC e ROD sono inserite all'interno di un *Crate*, una vera e propria scatola con appositi innesti per le schede. Il CERN ha deciso di utilizzare come contenitore il Crate dell'industria tedesca W-Ie-Ne-R, specializzata in elettronica. Per l'LHC la suddetta azienda ha creato un modello esclusivo di Crate VME64x 6021. Tutti i *sub-rack* sono costruiti secondo le specifiche IEEE 1101.10 sia per moduli di formato 6U che 9U. Oltre alla principale funzione di alloggiamento delle schede, il crate si occupa anche di:

- Fornire tre tipi di tensione differenti alle BOC e alle ROD;
- Dissipare il calore generato dalle schede attraverso un sistema di raffreddamento composto da ventole poste nel sub-rack del crate;
- Fare accedere da remoto alle schede con protocollo VME tramite connettori J1-J2-J3.

3.1 Crate W-Ie-Ne-R 9U VME64x 6021

Il Crate Wiener 9U VME64x 6021 (Fig. 3.1) è stato adottato sia dal CERN per l'esperimento ATLAS che dall'INFN, Istituto Nazionale di Fisica Nucleare e subnucleare, a Bologna per testare le schede ROD da poter spedire al CERN.

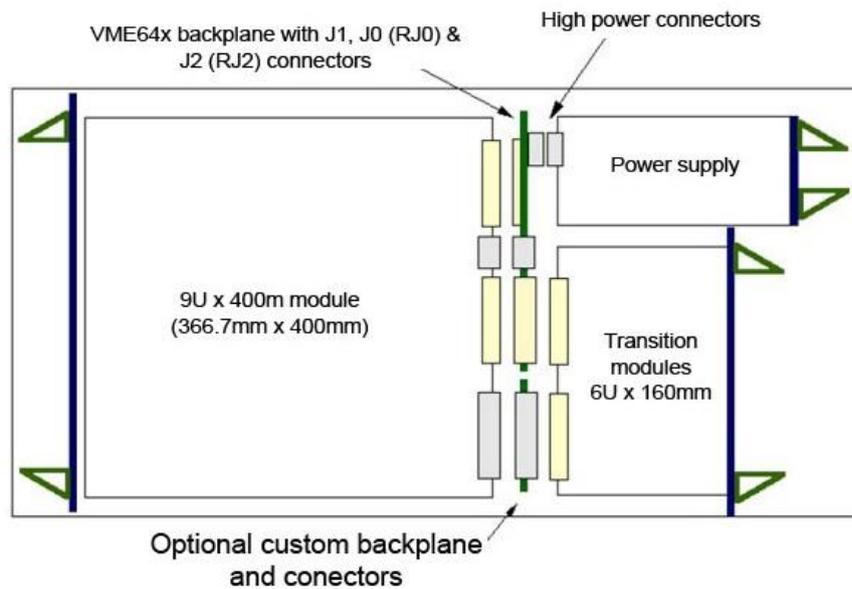
Le caratteristiche tecniche del crate sono le seguenti:

- Alloggiamenti di 19" per 21 schede VME64x 6U x 160mm o 9U x 400mm;

- VME64x compatibile con lo standard IEEE 1101.10/P, schermatura avanzata EMC / ESD;
- Struttura in acciaio-alluminio molto robusta di 5mm di spessore;
- Ventole UEL 6020 EX progettate per raffreddare superiormente i moduli VME con un flusso d'aria dal basso verso l'alto;
- Dotato di 6 ventole in corrente continua con regolazione automatica MTBF >60,000h/40°C;
- Monitoraggio della velocità delle ventole e della temperatura integrato;
- Microprocessore che permette la visualizzazione dei seguenti parametri: ventole, tensione, corrente, accensione, temperature, dati di set-up, etc. tramite display LCD;
- Costruito con possibilità di interfaccia multipla, Ethernet, RS232, CAN-bus, per il monitoraggio e controllo da remoto;
- 94V – 260V auto-range per la tensione alternata di input.



Fig. 3.1 a) Crate Wiener 9U VME64x 6021



b) Struttura interna del crate

3.2 Standard VME

Il VME, *Versa Module Eurocard*, è un protocollo standard utilizzato per determinare i bus e le dimensioni delle schede BOC e ROD. Ogni scheda può essere monitorata ed aggiornata tramite lo standard VME64x. Il crate adottato dal CERN, infatti, come già scritto nelle specifiche tecniche, supporta questo standard. I formati delle schede compatibili sono i seguenti:

- 6U x 160 mm, usato per le schede BOC, consente una trasmissione di 32 bit di indirizzi;
- 9U x 400 mm, usato per le schede ROD, anche in questo caso la lunghezza di trasmissione degli indirizzi è 32 bit .

In figura 3.2 sono mostrati i due moduli con rispettivi connettori e dimensioni.

Per quanto riguarda le caratteristiche dell'hardware, la connessione fra i moduli e il *backplane* è fatta di due connettori DIN 41612 da 96 pin. Il numero di connettori utilizzati definisce le dimensioni della scheda e la lunghezza dell'indirizzo. I pin scollegati forniscono un supporto per delle specifiche trasmissioni e per moduli posteriori di transizione. I moduli 9U possono presentare un terzo connettore per specifici usi.

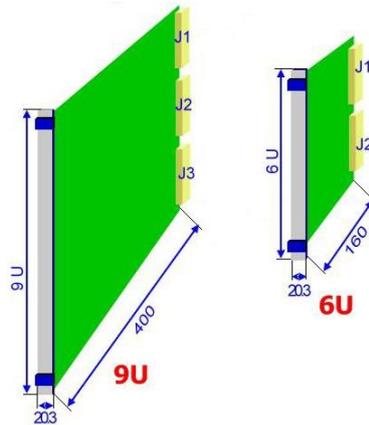


Fig. 3.2 Formati delle schede ROD e BOC

3.2.1 Descrizione del bus

Il bus VME è dotato di 31 linee di indirizzo. Le prime 23 linee sono presenti sul connettore J1, mentre le restanti si trovano sul connettore J2. Il bit di indirizzo meno significativo (A0) è sottointeso nel ciclo di trasferimento dati e non è infatti presente nel backplane.



Fig. 3.3 Vista del backplane del crate con connettori VME

Lo standard VME64x può multiplexare gli indirizzi e le linee dati consentendo 31 *bit* di indirizzi sul connettore J1 ed ulteriori 32 *bit* di indirizzi su J2.

Il bus VME ha anche 32 linee dati. Le prime 16 linee si trovano sul J1, e le rimanenti 16 sono sul connettore J2.

3.3 Crate W-Ie-Ne-R con controllo remoto tramite CAN-bus

A causa dell'incremento dei set-up elettronici degli attuali esperimenti è sempre più importante poter pilotare e monitorare strumenti come il crate. Attualmente è quindi possibile equipaggiare quest'ultimo con un collegamento CAN che permette il controllo remoto dei parametri.

L'utente può controllare e programmare ciascun parametro del crate attraverso l'interfaccia, come:

- Le tensioni;
- I limiti di corrente;
- I punti di lavoro di *Over-* e *Under-Voltage*;
- Le misure di temperature di: alimentatore, vano della presa d'aria, temperatura esterna dell'aria al di sopra dello slot 1;
- I segnali di stato;
- Le velocità delle ventole;
- L'identificativo del crate;
- Eventuali configurazioni o modifiche.

È inoltre possibile, ed estremamente importante, accendere o spegnere da remoto il crate. Alcuni dei valori soprascritti possono essere modificati, però, solo da personale autorizzato, ragione per la quale sono stati aggiunti dei *jumper* all'interno della *mainboard* per disabilitarne la modifica da utente.

Se qualunque errore fosse rivelato, un messaggio ad alta priorità verrebbe trasmesso senza richiesta alcuna dalla stessa interfaccia CAN. L'identificativo del crate è un numero che va da 1 a 126, ma è possibile anche abilitare una "chiamata generale" di accesso, per esempio per accendere più crate contemporaneamente (*broadcast*), e per disabilitare l'interfaccia.

3.1 Protocollo CAN-bus

Il *Controller Area Network* (CAN) definito da Bosch nel 1985 è un protocollo di comunicazione seriale avanzato. Grazie alla sua affidabilità e alle sue capacità tecniche, nonché al basso costo delle sue componenti, il CAN si presta bene per l'applicazione in sistemi di comunicazione. Le principali peculiarità del CAN sono:

- Numero illimitato di nodi;
- Comunicazione seriale, asincrona, orientata ad oggetti, e *multi-master*;
- Fino a 2032 *message IDs* in un *frame* standard;
- Massimo 8 *bytes* nel *frame data*;
- CSMA/CA *Collision Avoidance*, grazie al quale se due o più dispositivi iniziano a trasmettere contemporaneamente, si applica un meccanismo di arbitrato basato sulla priorità per decidere a quale dispositivo permettere di proseguire la trasmissione;
- Trasmissione ad ampio raggio, con velocità fino a 1.6 *Mbit/s* ($577 \frac{\text{Kbit}}{\text{s}}$ per informazione);
- Doppino a topologia stella o linea;
- Trasmissione dati immediata, per messaggi con alta priorità si ha un tempo di latenza garantito inferiore a 134 μs (1 *Mbit/s*);
- Elevato livello di affidabilità e sicurezza dovuto al rilevatore di errore integrato, grazie al quale il numero di errori per messaggio è minore di 10^{-13} .

Le specifiche e lo standard CAN si basano sul modello ISO (*International Organization for Standardization*)/OSI (*Open System Interconnections*), riassumibile con la seguente tabella.

OSI-Layer 7	Application	specified by system designer, several proposals as CMS(CAL)
OSI-Layer 6	Presentation	empty
OSI-Layer 5	Session	empty
OSI-Layer 4	Transport	empty
OSI-Layer 3	Network	empty
OSI-Layer 2	Data Link	covered by CAN-protocol specs and ISO standard, implemented on CAN-controller ICs
OSI-Layer 1	Physical	covered by ISO standard and partially by CAN protocol

Il protocollo CAN è definito nelle specifiche CAN versione 2.0 parte A (standard frame) e parte B (extended frame). L'ISO ha elaborato due tipologie di protocollo:

<ul style="list-style-type: none"> ISO/DIS11898 	CAN high-speed; Da 125kbit/s a 1Mbit/s, massimo 30 nodi.
<ul style="list-style-type: none"> ISO/DIS11519 part 1 	CAN low-speed; Fino a 125kbit/s, max. 20 nodi.

Sia il CAN high-speed che il low-speed usano un canale differenziale a due cavi con stessa terminazione. La massima distanza tra due nodi è determinata dal rateo di trasmissione (*Transmission Rate*), anche chiamato *Bauderate*, come indicato da tabella.

Max. Distance	Bit Rate	Type
10 m	1.6 Mbit/s	high- speed
40 m	1.0 Mbit/s	
130 m	500 kbit/s	
270 m	250 kit/s	
530 m	125 kbit/s	
620 m	100 kbit/s	low-speed
1300 m	50 kbit/s	
3300 m	20 kbit/s	
6700 m	10 kbit/s	
10.000 m	5 kbit/s	

Delle due tipologie ciò che interessa la nostra trattazione è l'high-speed CAN, poiché l'interfaccia che sarà utilizzata per la comunicazione prevede questo tipo di connessione.

Nel caso dell' high-speed CAN le linee di trasmissione sono terminate a 120Ω (impedenza caratteristica), come si può vedere nella figura 3.4.

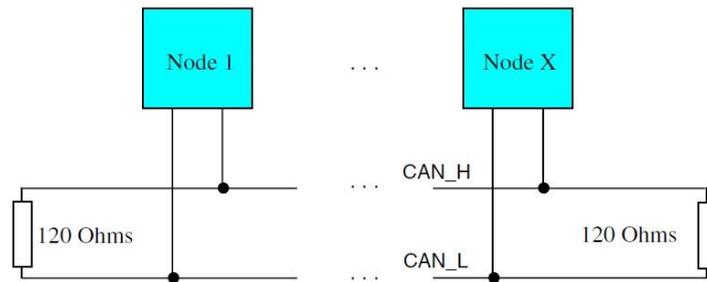


Fig. 3.4 Terminazione a 120 Ω per high-speed CAN

Per garantire un'alta flessibilità e un numero illimitato, almeno teoricamente, di nodi il trasferimento dei dati tramite CAN-bus è organizzato secondo il principio di trasmissione orientata ad oggetti. I nodi CAN non hanno un indirizzo fisso ma, come accennato in precedenza, ciascun messaggio può essere rilevato da qualunque nodo senza un indirizzo fisso, inviando il messaggio all'indirizzo di broadcast.

Il CAN *data frame*, ovvero il messaggio completo secondo il protocollo CAN, è mostrato in figura 3.5.

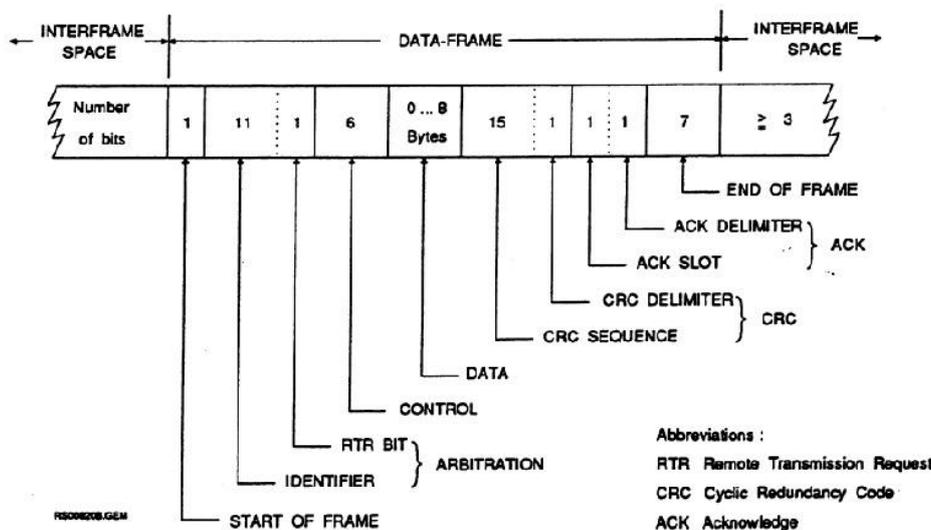


Fig. 3.5 CAN data frame

L'interfaccia CAN-bus del crate VME64x per il controllo remoto è un componente aggiuntivo che può essere montato all'interno del contenitore delle ventole. E' equipaggiato con un microcontrollore P80C592 della PHILIPS che include una CPU 80C51 e un CAN Controller secondo le specifiche CAN 2.0A. Per l'interfaccia CAN-bus il vano delle ventole è equipaggiato sul *front panel* con un connettore maschio DSUB a 9-pin.



Fig. 3.6 Connettore DSUB a 9-pin del crate

La struttura dell'*Identifier* o semplicemente ID è riportata nella tabella sottostante.

ID-bit	10	9	8	7	6	5	4	3	2	1	0
ACR- range	ac.7	ac.6	ac.5	ac.4	ac.3	ac.2	ac.1	ac.0	-	-	-
NodeID					n6	n5	n4	n3	n2	n1	n0
SubObjectID	s3	s2	s1	s0							

L'ID è una parte del frame che viene inviato al crate. Esso è composto da 11 *bit*.

I primi 4 *bit* più significativi identificano il *SubObject*, ovvero quella parte dell'ID che si riferisce alle 13 funzioni del crate. Le funzioni a cui il SubObject fa riferimento sono le seguenti, indicate con il corrispettivo valore dei 4 *bit*:

ID-Function	Name	s3	s2	s1	s0
Read Status from Crate	IDstat	0	0	0	0
Write Control Command to Crate	IDctrl	0	0	0	1
Read voltage/current channel 0+4	IDvc04	0	0	1	0
Read voltage/current channel 1+5	IDvc15	0	0	1	1
Read voltage/current channel 2+6	IDvc26	0	1	0	0
Read voltage/current channel 3+7	IDvc37	0	1	0	1
Read fan speed	IDfan	0	1	1	0
Read temperatures	IDtemp	0	1	1	1
reserved		1	0	0	0
Crate sends voltage configuration data	IDucfgC	1	0	0	1
Host requests/programs voltage configuration data	IDucfgH	1	0	1	0
Crate sends configuration data	IDcfgC	1	0	1	1
Host requests/programs configuration data	IDcfgH	1	1	0	0

Gli ultimi 7 *bit* dell'ID individuano il nodo e, più precisamente, l'address del crate, ovvero con quale crate si vuole comunicare. Come più volte ripetuto, il *Node-ID* può assumere tutti i valori da 1 a 127. Il valore 0 è vietato per evitare che l'oggetto del messaggio sia 0.

I tipi di messaggi che possono essere scambiati con il crate secondo il protocollo CAN-bus sono quattro. Ciascuno di essi ha una struttura ben definita e per la nostra trattazione se ne sono resi necessari solo tre, che andremo ora ad analizzare in dettaglio.

- *Data Frame*. Questo è il frame che contiene i dati che devono essere trasmessi al crate. Serve quindi per settare i parametri di quest'ultimo ed è stato utilizzato, in particolare, per accendere e spegnere il crate da remoto. Il messaggio che deve essere inviato si trova su un singolo nodo, o su più nodi nel caso venga utilizzato l'ID di broadcast. Come si può vedere nell'immagine sottostante (Fig. 3.7) il bit di Richiesta Remota di Trasmissione, detto RTR, deve essere uno 0 logico.

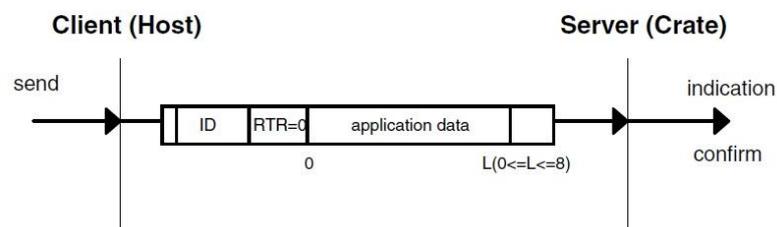


Fig. 3.7 Scrittura di parametri del crate

- *Remote Frame*. Questo frame richiede le informazioni su un determinato parametro del crate ($RTR = 1$ logico) in base all'ID che si imposta. In risposta il crate invia un messaggio con i dati contenuti nell'*application data*, come si può vedere in figura 3.8.

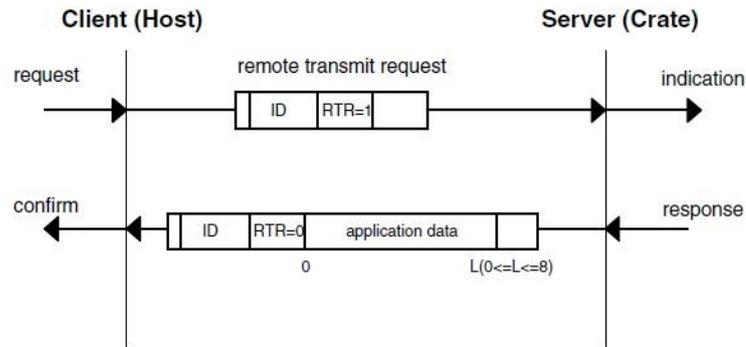


Fig. 3.8 Richiesta da remoto standard

- *Error Frame*. Nel caso in cui sia rilevato un errore del crate viene inviato automaticamente questo messaggio all'utente. Il *failure message* ha la più alta priorità e raggiunge pertanto l'utente nel minor tempo possibile, indipendentemente dall'ID del nodo.

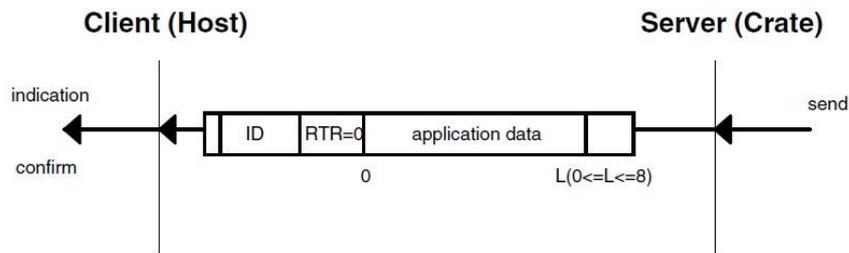


Fig. 3.9 Messaggio di errore

Prendendo in considerazione solo il data frame e il remote frame, differenti solo per RTR, le informazioni che si possono richiedere o ricevere dal crate sono le seguenti:

- IDstat – Get crate status

	RTR	Byte Count	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Host request	1	1-8	-	-	-	-	-	-	-	-
Crate confirm	0	same as host request	status byte 0	status byte 1	under-voltage error flags	over-voltage error flags	min. current error flags	over-current error flags	ovp error flags	temperature error flags

Status Byte 0		
Bit	0	1
0	power is off	power is on
1	external power inhibit	no external power inhibit
2	power fail	ac is in limit
3	any error, see other flags for specification	no error condition
4	fans are broken	fans are ok
5	trip off if fans are broken is disabled	trip off if fans are broken is enabled
6	trip off if any error is disabled	trip off if any error is enabled
7	vme bus signal sysfail active (low)	vme bus signal sysfail inactive (high)

Status Byte 1		
Bit	0	1
0	reserved (0)	
1	reserved (0)	
2	reserved (0)	
3	reserved (0)	
4	reserved (0)	
5	flash/eprom data has not changed since last access via can bus	flash/eprom data has changed (e.g. with the manual control of a fan tray)
6	no flash/eprom data checksum error	flash/eprom checksum error, default values are used
7	no write protect (service only)	hardware write protect

Dal 3° all'8° byte i segnali di errori saranno 0 se nessun errore si è presentato, altrimenti saranno 1. Se il terzo bit dello *Status Byte 0* è 0 allora il messaggio di errore verrà inviato come *failure message*, ovvero senza alcuna richiesta da parte dell'utente.

- IDctrl - Send Control Command

	RTR	Byte Count	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Host writes data to crate	0	1, 2	control byte	fan speed	-	-	-	-	-	-

Control Byte		
Bit	0	1
0	disable crate switch (see bit 1)	enable crate switch (see bit 1)
1	switch crate off (only used if bit 0 = 1)	switch crate on (only used if bit 0 = 1)
2	nothing to do	generate vme-sysreset
3		
4		
5		
6	error trip off enable	error trip off disable
7	don't change the fan speed	change the fan speed (new value in byte 2)

- IDvc04, IDvc15, IDvc26 and IDvc37 - Get measured Voltage and Current of channels 0 ... 7

	RTR	Byte Count	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Host request	1	1-8	-	-	-	-	-	-	-	-
Crate confirm	0	same as host request	u0 (u1, u2, u3), low byte	u0 (u1, u2, u3), high byte	i0 (i1, i2, i3), low byte	i0 (i1, i2, i3), high byte	u4 (u5, u6, u7), low byte	u4 (u5, u6, u7), high byte	i4 (i5, i6, i7), low byte	i4 (i5, i6, i7), high byte

Nel caso di letture di tensione o corrente, aspetto che verrà poi approfondito nel prossimo capitolo, il *low byte* e l'*high byte* dovranno essere invertiti per ottenere la misura corretta.

- IDfan - Get Fan speed

	RTR	Byte Count	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Host request	1	1-8	-	-	-	-	-	-	-	-
Crate confirm	0	same as host request	middle fan speed	nominal fan speed	fan 1 speed	fan 2 speed	fan 3 speed	fan 4 speed	fan 5 speed	fan 6 speed

L'unità di misura della velocità delle ventole è *giri/s*. Quando una ventola non è collegata o non è presente nel crate si leggerà il valore 255.

- IDTemp - Get Temperatures

	RTR	Byte Count	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Host request	1	1-8	-	-	-	-	-	-	-	-
Crate confirm	0	same as host request	ext/bin temp 1 (if not existing: fan air inlet temp.)	ext/bin temp 2 (or power supply air temp.)	ext/bin temp 3	ext/bin temp 4	ext/bin temp 5	ext/bin temp 6	ext/bin temp 7	ext/bin temp 8

Il range di temperature va da -127°C a 127°C . Se la temperatura indicata è -128°C il sensore non è collegato o non è presente nel crate.

Per quanto riguarda le funzioni di acquisizione e settaggio delle impostazioni del crate e del voltaggio di quest'ultimo, queste non verranno riportate poiché inutili ai fini della trattazione. Tuttavia per eventuali sviluppi futuri del software potrebbero risultare interessanti, per esempio per il settaggio da remoto delle velocità delle ventole o delle tensioni o correnti critiche.

Capitolo 4

Software LabVIEW di monitoraggio del Crate

Come accennato nel capitolo precedente è estremamente importante poter monitorare i parametri di un dispositivo complesso come il crate. Conoscere in ogni istante come quest'ultimo sta funzionando e sapere a che regimi lavora risulta utile soprattutto nella fase di test delle schede ROD, poiché permette di avere una panoramica globale e di ricevere immediatamente eventuali segnali di errore che potrebbero compromettere questa fase.

4.1 Interfaccia NI USB-8473

Poter controllare e impostare in qualsiasi momento i parametri più importanti del crate è stato reso possibile dallo sviluppo di un software tramite linguaggio LabVIEW, ovvero un linguaggio di programmazione ad alto livello prodotto dall'azienda americana National Instrument (NI). Quest'ultimo è stato scelto come linguaggio di programmazione poiché l'interfaccia tra il crate e il PC è anch'essa fornita dalla NI. In questo modo l'azienda garantisce all'utente driver compatibili e un insieme di funzioni, chiamato *Application Programming Interface* (API), utilizzabili per dialogare con l'hardware.

Il dispositivo di collegamento utilizzato è denominato NI USB-8473, e possiede le seguenti caratteristiche tecniche:

- 1 porta high-speed CAN;
- Hi-Speed USB;

- Cavo USB integrato al CAN-BUS transceiver;
- Supporto protocollo CAN 2.0A (11-bit) e protocollo CAN 2.0B (29-bit) arbitration ID;
- API e NI-CAN driver e software compatibili con LabVIEW;



Fig. 4.1 Interfaccia NI USB-8473

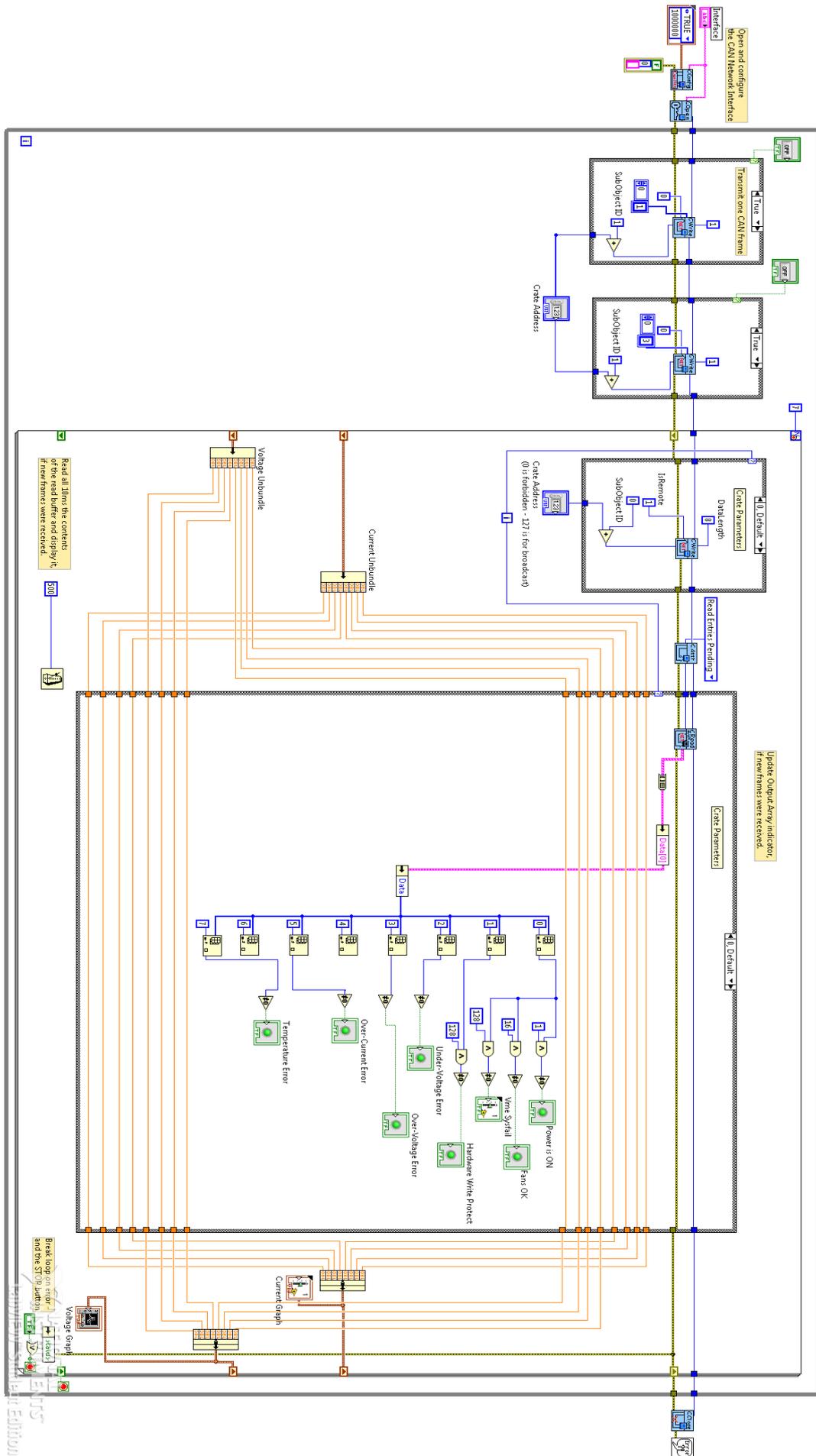
L'interfaccia NI USB-8473, così come le altre interfacce NI USB-847x, utilizza il connettore maschio standard D-SUB (DB9) a 9-pin per il collegamento CAN. La connessione USB Hi-Speed assicura che non vi sia un sovraccarico di frame. Il controllo degli errori di invio e ricezione gestito dall'hardware della scheda aiuta ad identificare eventuali problemi nei messaggi.

Come accennato in precedenza, la National Instrument fornisce i driver NI-CAN per Windows7/Vista/XP/2000. Questi driver includono una libreria dettagliata di oggetti, degli esempi di programmi per LabVIEW, e un Bus Monitor utile in fase di sviluppo software.

4.2 Sviluppo del software di gestione del Crate W-Ie-Ne-R

Il software sviluppato per la gestione del crate dà la possibilità all'utente di accendere e spegnere il dispositivo da un PC collegato tramite l'interfaccia NI

USB-8473 e fa visualizzare a schermo eventuali segnali di errore, l'andamento di tutte le tensioni e tutte le correnti, la velocità di tutte le ventole del crate, e le temperature rilevate dai sensori. Come il programma è stato costruito, ossia lo schematico del software, è il *Block Diagram* e ne verrà riportata l'immagine nella pagina seguente. Successivamente si può osservare invece il *Front Panel*, ossia l'interfaccia grafica che l'utente si troverà davanti all'apertura del programma.



STOP

Under-Voltage Error

Over-Voltage Error

Over-Current Error

Temperature Error

Vme Sysfail

Power is ON

Fans OK

Hardware Write Protect

OFF **ON**

Crate Address (0 is forbidden - 127 is for broadcast)

1

Crate Address

1

Interface

CAN0

Current Graph

Plot 0

Current Channel 0: 0,00

Current Channel 1: 0,00

Current Channel 2: 0,00

Current Channel 3: 0,00

Current Channel 4: 0,00

Current Channel 5: 0,00

Current Channel 6: 0,00

Current Channel 7: 0,00

Voltage Graph

Plot 0

Voltage Channel 0: 0,00

Voltage Channel 1: 0,00

Voltage Channel 2: 0,00

Voltage Channel 3: 0,00

Voltage Channel 4: 0,00

Voltage Channel 5: 0,00

Voltage Channel 6: 0,00

Voltage Channel 7: 0,00

Middle Fan Speed RPM: 0

Fan 1 Speed RPM: 0

Fan 2 Speed RPM: 0

Fan 3 Speed RPM: 0

Fan 4 Speed RPM: 0

Fan 5 Speed RPM: 0

Fan 6 Speed RPM: 0

Nominal Fan Speed RPM: 0

Exd/bin Temp1 (°C): 0

Exd/bin Temp2 (°C): 0

Exd/bin Temp3 (°C): 0

Exd/bin Temp4 (°C): 0

Exd/bin Temp5 (°C): 0

Exd/bin Temp6 (°C): 0

Exd/bin Temp7 (°C): 0

Exd/bin Temp8 (°C): 0

4.2.1 Analisi delle funzioni

Le prime due funzioni che sono state utilizzate sono *ncConfigCANNet* e *ncOpen*. La prima configura l'interfaccia e la seconda dà inizio alla comunicazione fra il crate e il PC. Viste nel dettaglio si presentano nel Block Diagram come in figura 4.1.

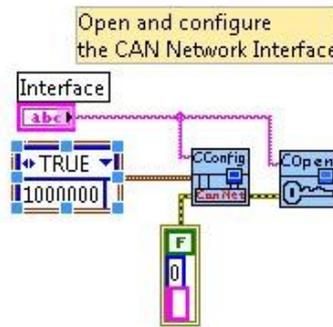
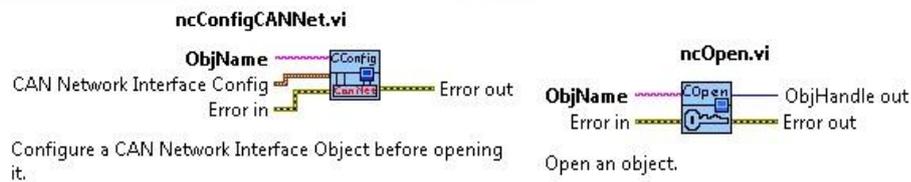


Fig. 4.1 CConfig e COpen



Un dato particolarmente importante è il Baud Rate. Come già visto nel capitolo precedente, questo dà l'informazione di quanti dati devono essere trasmessi al secondo. Il Baud Rate è stato impostato a 1000000 direttamente nel *CAN Network Interface Config* (dato di tipo Bundle), poiché il crate si aspetta, come indicato sul datasheet, dati ad una velocità di trasmissione di 1 *Mbit/s*.

Assieme ad esso è stato settato su TRUE lo *Start On Open*, che dà l'informazione all'oggetto di fare partire la procedura all'apertura del programma.

All'interno del ciclo While colorato in grigio ha inizio la vera e propria comunicazione fra l'interfaccia CAN-bus e il crate. Le funzioni utili a questo scopo sono principalmente due: *ncWriteNet* e *ncReadNetMult*. Come già affrontato in dettaglio nel capitolo precedente, il protocollo CAN-bus prevede che sia l'utente, salvo il caso di failure message, a contattare il crate per settare dei parametri o per richiederne le specifiche. La prima delle due funzioni sovrascritte ha il compito di inviare un messaggio codificato al crate. Nella funzione *CWrite* è

possibile inserire molte delle informazioni che un frame prevede. Il resto del messaggio viene completato direttamente dall'oggetto.

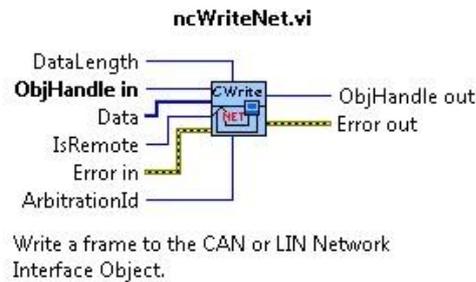


Fig. 4.2 CWrite

Come si può vedere in figura 4.2 in ingresso all'oggetto si possono impostare:

- *ArbitrationId*, che corrisponde agli 11 *bit* che danno l'informazione di quale parametro è richiesto o è da impostare, compreso l'address del crate;
- *IsRemote*, ovvero l'RTR, il bit che imposta la sola scrittura o la richiesta-lettura del parametro specificato;
- *Data*, che definisce gli 8 *byte* del messaggio. Questo valore è stato specificato solo nel caso in cui l'*IsRemote* è 0, ovvero quando si sta inviando un messaggio di sola scrittura per l'impostazione del parametro dell'*ArbitrationId*;
- *Data Length*, ovvero il numero di byte dell'array *Data*;

Nel caso in cui si sia impostato nel *CWrite* l'RTR a 1, la funzione *CRead* dà la possibilità di leggere il messaggio che il crate invia in risposta alla richiesta di un determinato parametro. Nel caso specifico di questo software è stato utilizzato l'*ncReadNetMult* così che tutte le informazioni, ovvero *IsRemote*, *ArbitrationId*, *Data Length*, *Data*, e *Timestamp* fossero contenute in un unico pacchetto, detto *Cluster*. Questa scelta verrà giustificata nel prossimo paragrafo, dove verranno mostrati i singoli *Case* con all'interno le relative funzioni.

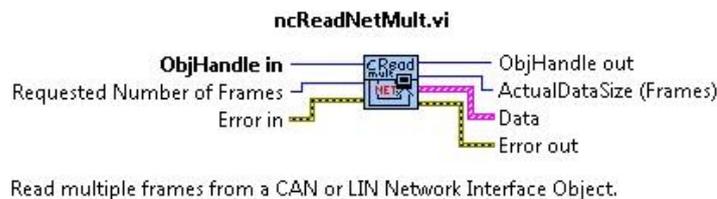


Fig. 4.3 CRead

In conclusione, per chiudere l'oggetto, e quindi la connessione fra crate e interfaccia, è presente la funzione CClose, alla quale è collegato in uscita all'ErrorOut un oggetto nominato Simple Error Handler che ha il compito di evidenziare eventuali errori durante l'esecuzione del software.

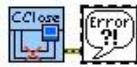
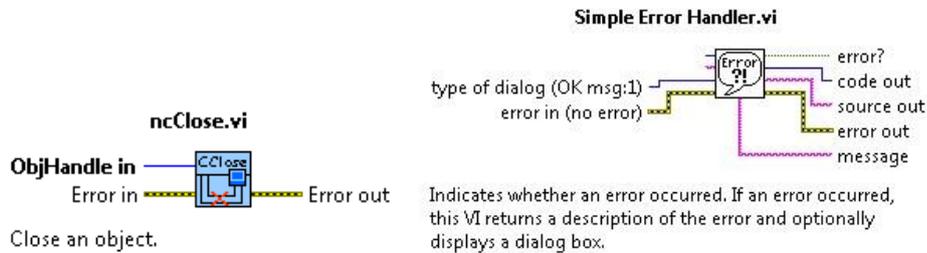


Fig. 4.4 CClose e Simple Error Handler



4.2.2 Analisi dei Case

Dal momento che i parametri da monitorare sono diversi si è resa necessaria una struttura condizionale, il *Case*. Questa si compone di più sottodiagrammi, ciascuno dei quali si esegue quando si verificano determinate condizioni impostate dall'utente. E' stato così possibile raggruppare più sottodiagrammi, ciascuno con il proprio compito, in modo da non solo ridurre in maniera considerevole le dimensioni del Block Diagram, ma anche rendere più fluido e comprensibile a chiunque il codice, in previsione anche di eventuali modifiche future.

I primi due Case si occupano rispettivamente dello spegnimento e dell'accensione del crate. Come si può vedere in figura 4.5, le due possibili condizioni sono TRUE o FALSE. Quest'ultima è la condizione di default perciò, a meno che l'utente non intervenga cliccando i pulsanti OFF o ON presenti sul Front Panel, sarà il sottodiagramma nel FALSE ad essere eseguito. Se, invece, si interviene operando su uno dei due pulsanti verrà eseguito il sottodiagramma nel TRUE. Per lo spegnimento del crate il primo case contiene nel TRUE un oggetto CWrite con: IsRemote = 1, ArbitrationId corrispondente alla somma del *SubObjectID* = 0001 ovvero "Write Control Command to Crate" più l'address del crate

impostabile direttamente dal Front Panel, Data Length = 1 poichè è solo il primo byte quello contenente il bit di *switch-off*, e, infine, il Data corrispondente allo spegnimento, ovvero, in esadecimale, Data = 1 . Il secondo Case per l'accensione del crate ha la stessa struttura del precedente, ma questa volta il CWrite ha in ingresso Data = 3.

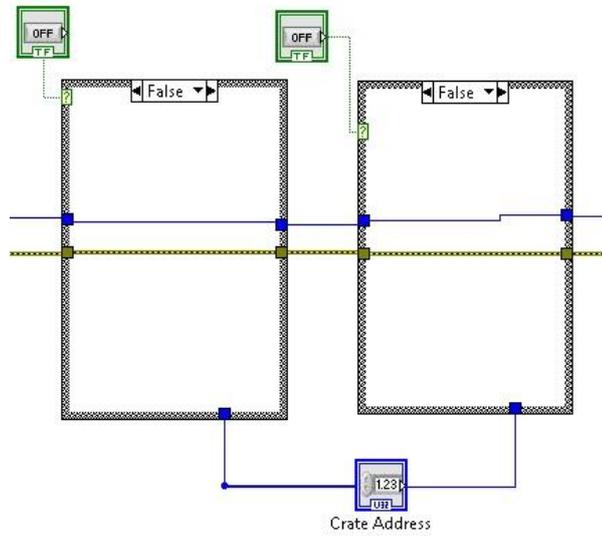
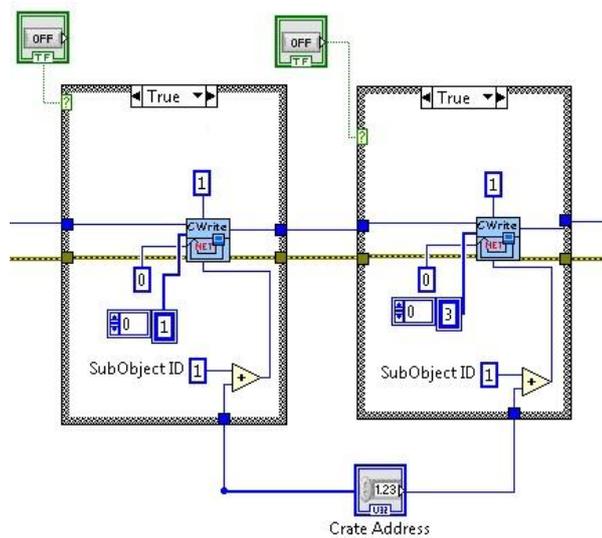


Fig. 4.5 Case di spegnimento e accensione. a) FALSE condition



b) TRUE condition

Sul Front Panel i due pulsanti si presentano all'utente come in figura 4.6.



Fig. 4.6 Switch-off e Switch-on

Gli ultimi due Case si trovano all'interno di un ciclo For che si ripete tante volte quanto è il numero delle condizioni dei case stessi. Nel caso del programma il numero è impostato come costante e vale 7, ovvero il ciclo si ripete sette volte, da 0 a 6. La condizione del ciclo è estremamente importante perché ad ogni incremento di quest'ultimo incrementa anche la condizione del case, così che tutti e sette verranno eseguiti, con un clock impostato a 500 ms.

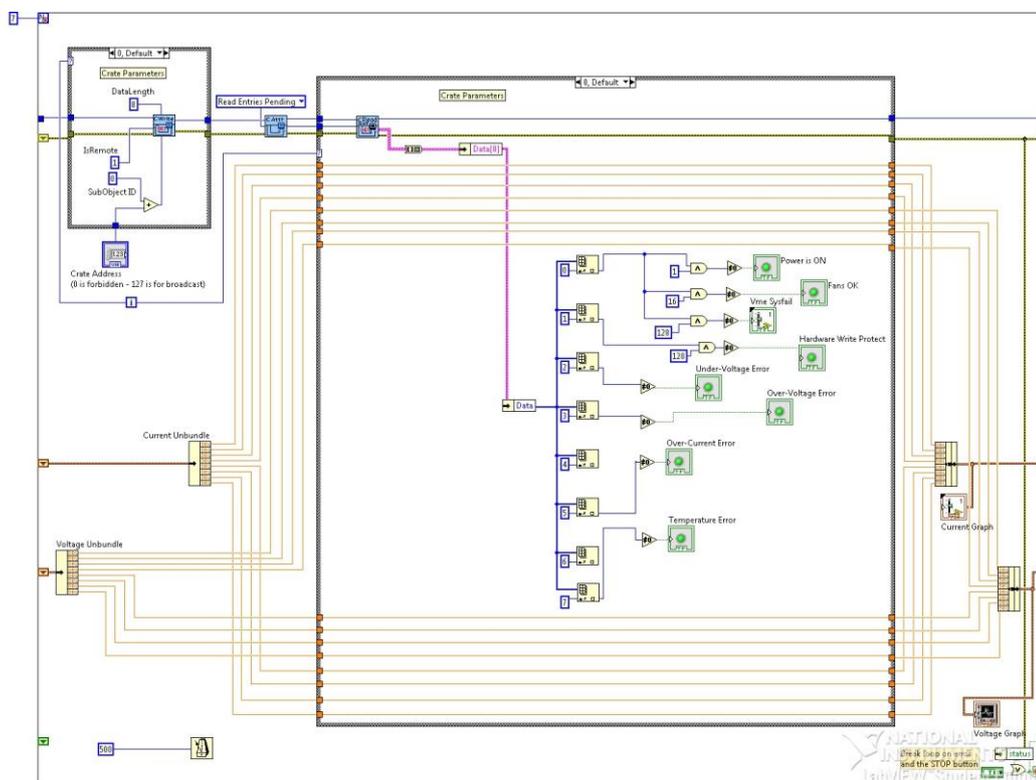


Fig. 4.7 FOR loop

Come si può vedere dall'immagine sovrastante, il ciclo è piuttosto elaborato. Sono presenti *Bundle* e *Unbundle* dei quali verrà spiegato l'utilizzo quando saranno analizzati i case per il monitoraggio di tensione e di corrente. Tutti gli altri case, infatti, sono stati solo attraversati dai collegamenti delle due funzioni senza che da essi venisse preso alcun valore.

I sette case del ciclo sono i seguenti:

- 0- *Crate Parameters*. I primi sottodiagrammi di questo case permettono di visualizzare se il crate è acceso, se le ventole sono funzionanti, se il *Write Protect* è attivo, ed eventuali segnali di errore. Per poter ricevere questo tipo di informazioni l'utente deve prima inviare un messaggio di richiesta di

ricezione di questi parametri, che avviene nel primo dei due case in successione. Come si può vedere nel dettaglio in figura 4.7, nella CWrite l'RTR è ora impostato a 0, così come sono settati secondo ciò che si vuole conoscere anche il Data Length e soprattutto il SubObjectID. Anche qui l'address del crate può essere impostato dall'utente direttamente nell'interfaccia grafica. Dopo che il messaggio di richiesta è stato inviato i dati vengono acquisiti ed elaborati dalla CRead. Come è stato detto in precedenza questa funzione riporta in uscita un array, che viene convertito in cluster. Di quest'ultimo l'unico elemento importante ai fini del monitoraggio è il Data, ossia gli 8 *byte* contenenti le informazioni richieste. Dal momento che ogni byte contiene 8 *bit*, ciascuno dei quali fornisce un dato diverso, per ciascun byte è stato selezionato il bit d'interesse aggiungendo un corrispettivo nel Front Panel, mentre gli altri sono stati trascurati.

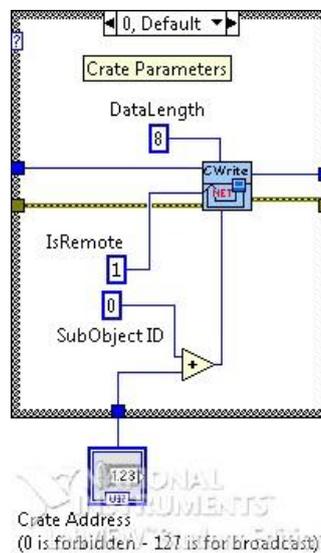
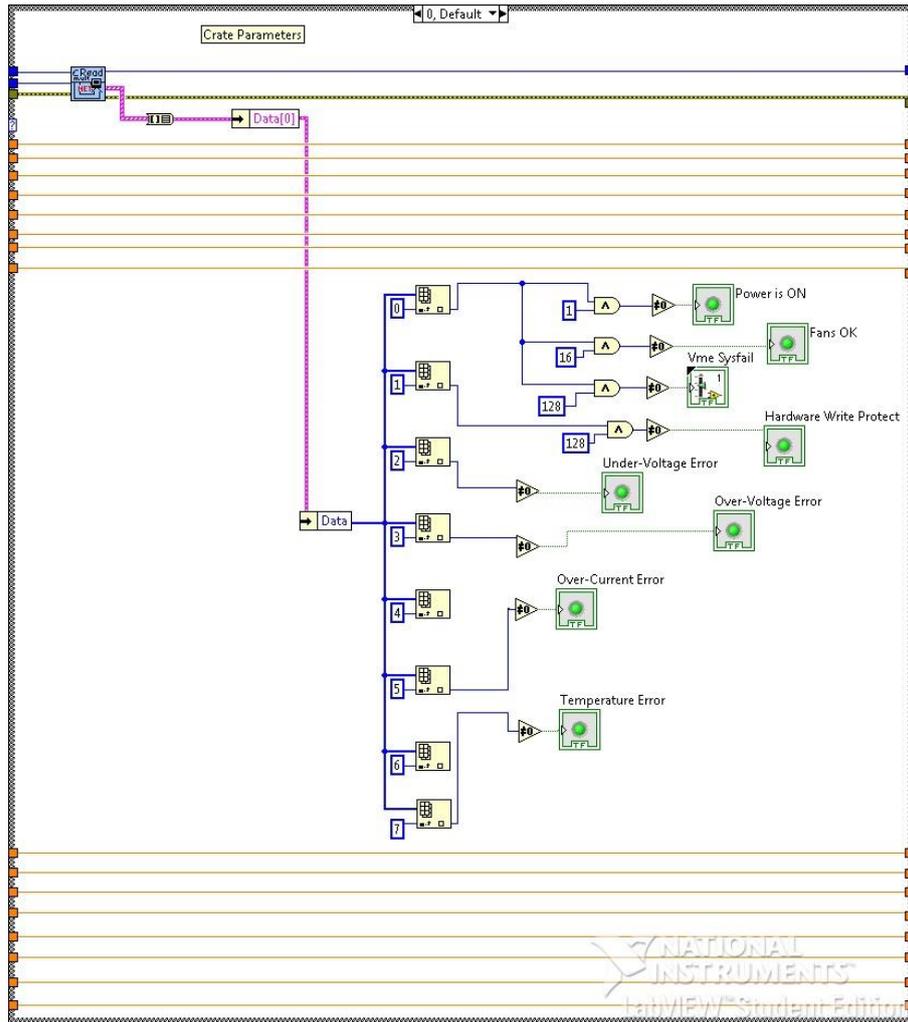


Fig. 4.8 Crate Parameters. a) Case di richiesta



b) Case di lettura

Sul Front Panel i dati di interesse, quali: *Power is ON*, *Fans Ok*, *Vme Sysfail*, *Hardware Write Protect*, *Under-Voltage Error*, *Over-Voltage Error*, *Over-Current Error* e *Temperature Error*, si presentano come LED.



Fig. 4.9 LED dei parametri del crate

- 1- *Fan Speed*. Se la condizione dei due case nel ciclo For è questa il primo dei due invia un messaggio di richiesta al crate per ricevere le velocità delle ventole, mentre nel secondo il messaggio viene spaccettato per poter visualizzare sul Front Panel un indicatore di ciascuna velocità. Il case di scrittura è strutturato esattamente come quello analizzato per il *Crate Parameters*, e come tutti i successivi di richiesta. L'unico parametro che cambia è il SubObjectID, che questa volta è impostato su 300 esadecimale. Il

b) Case di lettura

Sul Front Panel gli indicatori di velocità hanno l'aspetto di contagiri, come mostrato nella figura sottostante. La *Nominal Fan Speed*, essendo una costante del crate, si presenta invece solo come indicatore numerico.



Fig. 4.11 Esempio di indicatore di Fan Speed

2- *Temperature Values*. Per la lettura delle temperature il primo case di scrittura ha ora SubObjectID = 380 *esadecimale*, mentre nel case di lettura i dati sono stati convertiti in modo che potesse essere letto anche il segno.

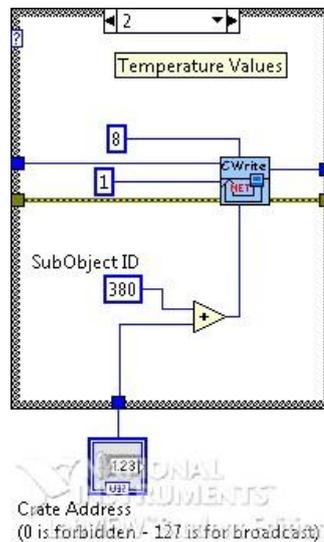
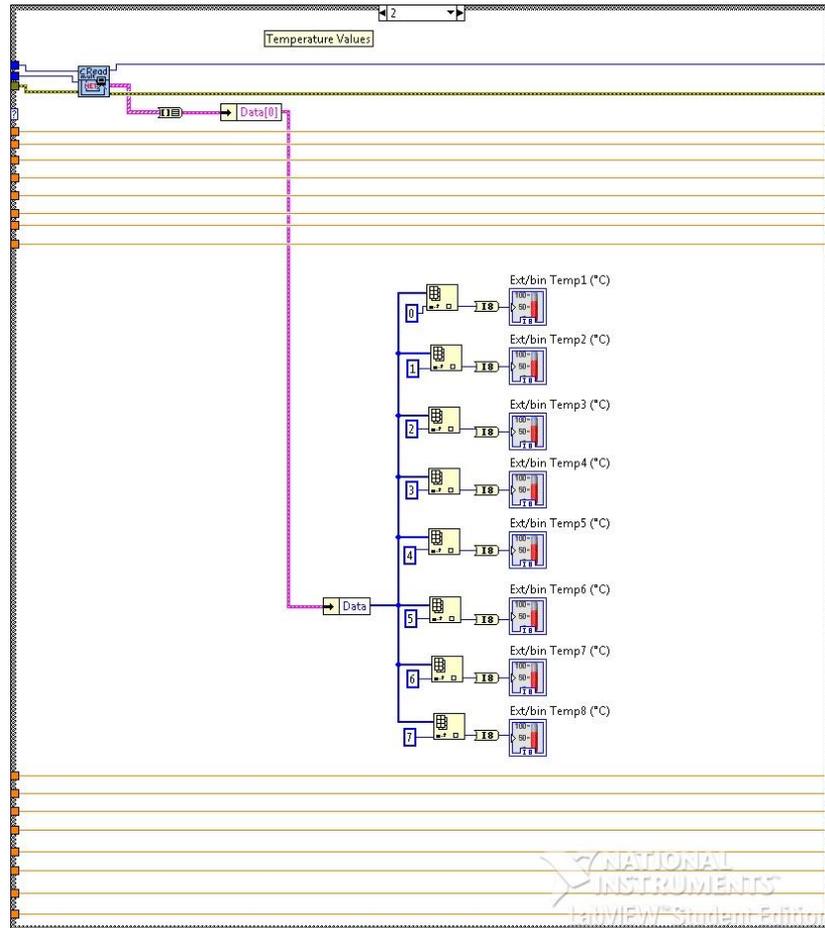


Fig. 4.12 Temperature Values a) Case di richiesta



b) Case di lettura

Sul Front Panel per le temperature è stato aggiunto un termometro, così che la lettura del parametro fosse semplice e intuitiva, oltre a un display numerico che dà il valore in °C. Quando la temperatura indicata è -128°C significa, come detto in precedenza, che il sensore per la rilevazione di quel dato non è presente o non è collegato.

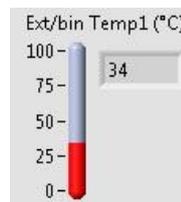


Fig. 4.13 Termometro

3.2.3 *Voltage/Current Channels 0-4 (1-5; 2-6; 3-7)*. La condizione numero 3 per entrambi i case permette di richiedere e ricevere informazioni riguardo la tensione e la corrente dei canali di alimentazione 0 e 4 del crate. Le rimanenti tre condizioni permettono di avere le stesse informazioni sui canali,

rispettivamente, 1 e 5, 2 e 6, 3 e 7, e la loro struttura è pertanto identica a quella che verrà analizzata. Il case di scrittura è articolato come quelli precedenti. In questo caso il SubObjectID è impostato a 100 esadecimale (per gli altri canali: 180 canali 1-4, 200 canali 2-6, 280 canali 3-7). Il case di lettura dei parametri prevede delle operazioni particolari. Infatti, come è già stato anticipato nel capitolo precedente, la funzione *Get measured Voltage and Current of channels 0 ... 7* riporta dei byte che sono invertiti, ovvero il *low byte* si trova in ordine di trasmissione prima dell'*high byte*. Questo significa che per elaborare il valore corretto di tensione o corrente di un canale il byte più significativo debba essere spostato prima di quello meno significativo. Per invertire a due a due i byte sono state svolte tre operazioni: la prima moltiplica l'*high byte* per 100 esadecimale, così che questo guadagni due ordini di grandezza; il risultato di questa operazione viene sommato al valore in uscita del *SubArray 0* che corrisponde al byte 0, ovvero al *low byte*; infine la somma viene divisa per 64 esadecimale, ottenendo così il valore di tensione/corrente in decimale. Ciascun dato di tensione o corrente viene inserito all'interno di un unico grande contenitore, uno per le tensioni, l'altro per le correnti, grazie allo strumento *Bundle*. Ciascuno di essi ha in uscita un *Waveform Chart* sul quale vengono riportati in successione, con un lo stesso clock del ciclo For, uno ad uno i dati acquisiti. Ciò permette di avere tutti i valori dello stesso tipo su un unico grafico così da non solo ridurre il Front Panel, ma anche vedere immediatamente l'andamento di ciascun canale rispetto agli altri.

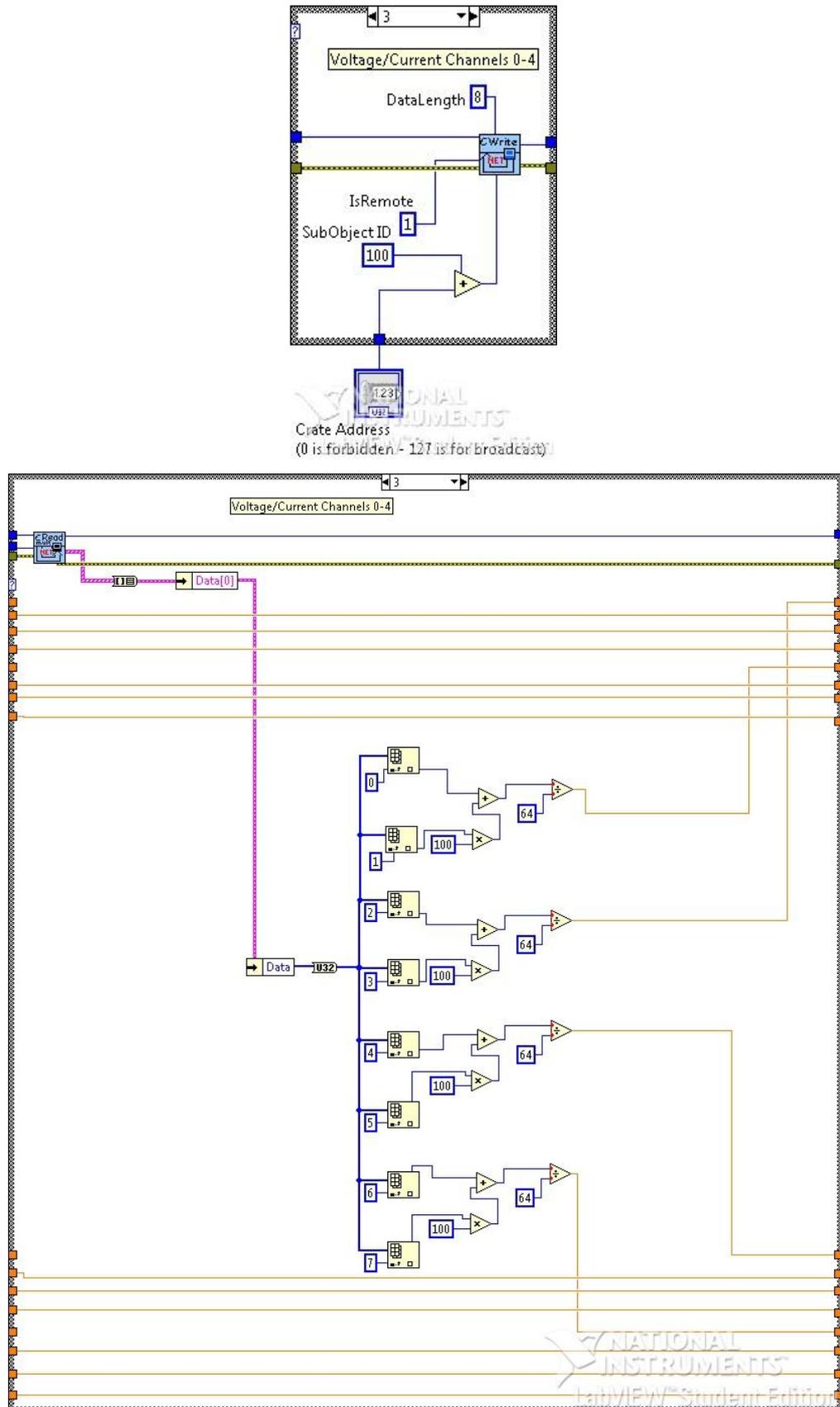


Fig. 4.14 Voltage/Current Channels 0-4 a) Case di richiesta, b) Case di lettura

I grafici Corrente-Tempo (A/s) e Tensione-Tempo (V/s) sul Front Panel si presentano come in figura 4.15. Come si può vedere ciascuna linea di tendenza ha un colore, così da evidenziare i diversi canali. E' inoltre presente per ciascun canale un display numerico per monitorare esattamente il valore di ciascun parametro in un determinato istante.

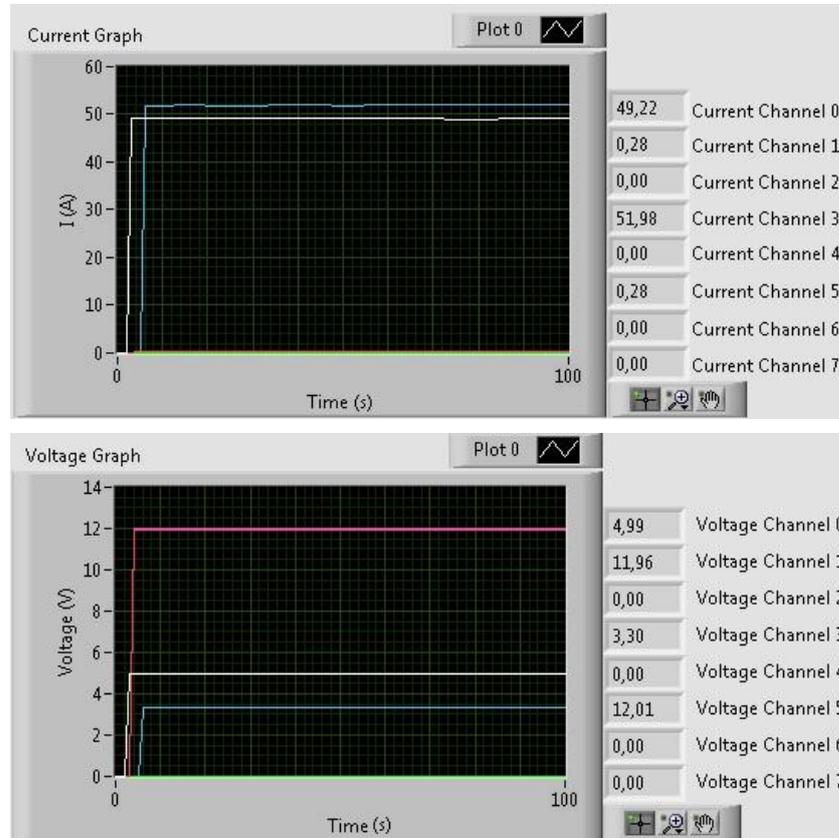


Fig. 4.15 Grafici di corrente (1) e tensione (2)

4.2.3 Specifica dei requisiti software

Requisiti minimi di sistema Windows (sola esecuzione):

- LabVIEW Run-Time Engine 2011 o superiore;
- NI-CAN 14.0 o superiore;
- Interfaccia NI USB-8473 o compatibile;
- Pentium III / Celeron 866 MHz o equivalente;
- RAM 256 MB;
- Risoluzione 1024 x 768 pixel;

- Windows 8.1/8/7/Vista (32 e 64-bit). Windows XP (32-bit). Windows Server 2012 R2 (64-bit). Server 2008 R2 (64-bit). Server 2003 R2 (64-bit);
- Spazio su disco 500 *MB*.

Requisiti minimi di sistema Windows (modifica e sviluppo):

- LabVIEW 2011 o superiore;
- NI-CAN 14.0 o superiore;
- Interfaccia NI USB-8473 o compatibile;
- Processore Pentium 4/M o equivalente;
- Ram 1 *GB* ;
- Risoluzione 1024 x 768 *pixel*;
- Windows 8.1/8/7/Vista (32 e 64-bit). Windows XP (32-bit). Windows Server 2012 R2 (64-bit). Server 2008 R2 (64-bit). Server 2003 R2 (64-bit);
- Spazio su disco 5 *GB* (include i driver di default inclusi in NI Device Drivers DVD).

Requisiti minimi di sistema Mac OS X (sola esecuzione):

- LabVIEW Run-Time Engine 2011 o superiore;
- NI-CAN 14.0 o superiore;
- Interfaccia NI USB-8473 o compatibile;
- Processore Intel-based;
- RAM 256 *MB*;
- Risoluzione dello schermo 1024x768 *pixel*;
- OS X 10.7, 10.8, 10,9 ;
- Spazio su disco 656*MB* – 1.2 *GB*.

Requisiti minimi di sistema Mac OS X (modifica e sviluppo):

- LabVIEW 2011 o superiore;
- NI-CAN 14.0 o superiore;

- Interfaccia NI USB-8473 o compatibile;
- Processore Intel-based;
- RAM 2 GB;
- Risoluzione dello schermo 1024x768 pixel;
- OS X 10.8 o 10.9 ;
- Spazio su disco 1.2 GB per l'installazione completa (esclusi driver).

Requisiti minimi di sistema Linux (sola esecuzione):

- LabVIEW Run-Time Engine 2011 o superiore;
- NI-CAN 14.0 o superiore;
- Interfaccia NI USB-8473 o compatibile;
- Processore Pentium III / Celeron 866 MHz o equivalente;
- RAM 256 MB;
- Risoluzione 1024 x 768 pixel;
- Linux kernel 2.4x, 2.6x, 3.x e GNU C Library (glibc) Versione 2.5.1 per l'architettura Intel x86_64. *LabVIEW Installation Guide* per errore non elenca Linux kernel 3.x in questa lista;
- Spazio su disco 115 MB (32-bit), 131 MB (64-bit).

Requisiti minimi di sistema Linux (modifica e sviluppo):

- LabVIEW 2011 o superiore;
- NI-CAN 14.0 o superiore;
- Interfaccia NI USB-8473 o compatibile;
- Processore Pentium 4/M o equivalente;
- RAM 1 GB;
- Risoluzione 1024 x 768 pixel;
- Red Hat Enterprise Linux Desktop + Workstation 6 o superiore, open SUSE 12.3 o 13.1, Scientific Linux 6 o superiore;
- Spazio su disco 1.2 GB per l'installazione completa di ogni fitness, 1.4 GB per l'installazione completa di entrambe le versioni di LabVIEW da 32- e 64-bit.

Conclusioni e sviluppi futuri

Il software sviluppato per questa tesi ha raggiunto fino a questo momento un livello di ottimizzazione tale da garantire il miglior compromesso tra affidabilità, stabilità ed immediatezza di utilizzo. Tutti i parametri del crate W-Ie-Ne-R in questione sono facilmente comprensibili e la loro interpretazione è semplice e rapida. Avere a disposizione un PC sempre operativo e collegato al crate permetterebbe di accendere o spegnere in ogni momento quest'ultimo, nonché di conoscere il suo stato di funzionamento anche a distanza. Vi sono, tuttavia, diverse migliorie che possono essere apportate al software: si può ridurre il numero di finestre per l'inserimento dell'address del crate, o anche aggiungere tutti gli indicatori dei possibili errori che potrebbero essere riscontrati. Inoltre si potrebbe sviluppare un algoritmo in grado di creare una dipendenza fra più parametri, che si andrebbero così a modificare automaticamente. Un esempio potrebbe essere la regolazione automatica della velocità delle ventole per il raffreddamento in funzione della temperatura rilevata.

Questo programma viene già utilizzato nel laboratorio di progettazione elettronica dell'INFN di Bologna, dove è stato peraltro concepito, sviluppato e messo a punto. Il CERN di Ginevra ha adottato per ognuno dei quattro esperimenti dell'LHC lo stesso crate dell'azienda tedesca W-Ie-Ne-R, ragione per la quale la progettazione di un software di questo tipo trova sicuramente una collocazione all'interno della catena di acquisizione ed elaborazione dati di ciascun esperimento.

In un futuro più remoto sarà anche possibile liberarsi dal vincolo di utilizzo dell'interfaccia NI USB-8473, e quindi anche di un software progettato con linguaggio LabVIEW, nonostante dal suo utilizzo ne derivino diverse facilitazioni. Inoltre per eventuali modifiche del programma è necessaria la licenza LabVIEW, che è estremamente vincolante dal punto di vista dello sviluppo.

Bibliografia

- [1] ATLAS IBL community. ATLAS Insertable B-Layer technical design report 2010.
- [2] CERN. “The Large Hadron Collider” <http://home.web.cern.ch/topics/large-hadron-collider> 06-2015.
- [3] L. Lama, “Development and Testing of the ATLAS IBL ROD preproduction boards”, 2013.
- [4] National Instrument. “NI USB-8473” <http://sine.ni.com/nips/cds/view/p/lang/it/nid/203384> 06-2015
- [5] C. Preti, “Test hardware del primo batch di schede ROD per l’esperimento ATLAS IBL”, 03-2012.
- [6] A. Ruben, A. Köster, “W-IE-NE-R Plein & Baus GmbH” Müllersbaum 20, 51399 Burscheid 25 January 1996.