
OSSERVATORIO ASTROFISICO DI CATANIA

CCDLab: un software per la caratterizzazione elettro-ottica di rivelatori per l'astrofisica

V 1.0



Osservatorio Astrofisico di Catania

Alessandro Grillo⁽¹⁾

(1) INAF - Osservatorio Astrofisico di Catania

Rapporti interni e tecnici
N.01/2012

INAF - Osservatorio Astrofisico di Catania

Via Santa Sofia, 78 I-95123 Catania, Italy Tel.: +39- 095-7332 111 Fax: +39-095-330592

Sede "Mario G.Fracastoro" (Etna) - Tel +39-095-911580 Fax+39-095-916184

www.oact.inaf.it - oacatania@oact.inaf.it



ISTITUTO NAZIONALE DI ASTROFISICA
NATIONAL INSTITUTE FOR ASTROPHYSICS

OSSERVATORIO ASTROFISICO DI CATANIA

Introduzione

Nella caratterizzazione di un grande numero di sensori in un tempo relativamente breve e con margine di errore il più basso possibile, l'automazione del sistema di caratterizzazione (PDE, misure di dark, guadagno, ecc.) diventa un fattore molto importante.

Per raggiungere tale obiettivo ho realizzato il software CCDLab per il controllo dei motori e della strumentazione del banco di caratterizzazione ottica dei rivelatori presso il laboratorio COLD dell'Osservatorio Astrofisico di Catania.

Tale software consente di effettuare misure in serie in maniera totalmente automatica. Esso è stato sviluppato in linguaggio Java e sfrutta le primitive di comunicazione delle porte RS-232 e IEEE-488, presentando un'interfaccia grafica intuitiva e molto semplice da gestire. Inoltre tale software, proprio perché scritto in Java, è indipendente dalla piattaforma, dando all'utente piena libertà di scelta del sistema operativo da utilizzare. Con pochi click del mouse, l'utente è in grado di controllare i motori elettrici del banco ottico ed acquisire i dati prodotti dai vari strumenti collegati ai sensori. Un'altra caratteristica importante è la possibilità di controllare l'intero sistema da remoto.

1 - Il linguaggio Java nella comunicazione seriale e GPIB (IEEE-488)

Il linguaggio di programmazione Java nel corso degli anni si è rivelato molto versatile in svariati ambiti di sviluppo, tra cui quello della programmazione delle interfacce seriali e GPIB. In questi casi, data l'indipendenza di Java dalla piattaforma, è necessaria una API standardizzata con implementazioni che tengano conto della piattaforma in uso.

Sfortunatamente, la Sun non ha prestato molta attenzione alla comunicazione seriale in Java. Infatti esiste sì una API definita dalla Sun per tale scopo, JavaComm, ma tale API non è prevista all'interno della Java Standard Edition. In particolare alla fine del 2005 la Sun fece sparire il supporto per Windows alla JavaComm e questa situazione, unita al fatto che la Sun originariamente non ha provveduto ad un'implementazione di JavaComm per Linux, ha diretto verso lo sviluppo di una libreria gratuita, la RXTX. Essa è disponibile non solo per Linux, ma per diverse piattaforme, Windows compresa. In quest'ultimo caso, unitamente alle classi Java della libreria RXTX, contenute nel file RXTXComm.jar, si utilizza la libreria RXTXSerial.dll contenente le apposite API Win32/64.

Uno degli indiscutibili vantaggi di RXTX sta nell'implementazione di una API pienamente equivalente e compatibile con la JavaComm: la cosa si traduce nella possibilità, in ambiente Linux o Solaris, di usare la libreria che si preferisce cambiando semplicemente un *import* e lasciando il resto del codice sorgente invariato.

L'interfaccia seriale RS-232, sebbene ormai obsoleta e soppiantata da tempo dalla più moderna USB, è ancora ampiamente usata nell'ambito dei controlli elettronici, in particolare nel controllo dei motori elettrici DC e passo-passo. Addirittura in alcuni casi si supplisce alla mancanza della porta seriale nei PC più moderni mediante degli appositi convertitori USB – RS-232, questo per non dover sostituire i vecchi controller seriali dei motori, i quali sono spesso molto costosi.

L'interfaccia GPIB è utilizzata per lo più nell'acquisizione dati di strumentazioni di misura professionali.

OSSERVATORIO ASTROFISICO DI CATANIA

A differenza dell'interfaccia seriale, per la GPIB non è stata fornita una API standardizzata né dalla Sun, né da terze parti. Per accedere ai dispositivi GPIB, in cui è prevista la lettura/scrittura di registri di sistema, cosa estranea all'architettura Java per la sua indipendenza dalla piattaforma, è necessario l'utilizzo di metodi nativi del sistema su cui si lavora, in particolare del meccanismo JNI (Java Native Interface).

Questo meccanismo prevede l'implementazione delle primitive di comunicazione in un linguaggio più prossimo all'hardware utilizzato (ad es. il C) e la conseguente importazione di tali primitive all'interno di Java mediante un *wrapper*, cioè un involucro in grado di interfacciare tali metodi nativi con il linguaggio Java, mediante il quale essi saranno invocati.

2 - Il banco ottico

Il banco ottico sito presso il laboratorio COLD dell'Osservatorio Astrofisico di Catania serve ad effettuare misure di caratterizzazione su diversi tipi di rivelatori.

Esso è composto da cinque moduli collegati per mezzo di flange in modo da costituire un unico sistema. La Fig. 1 mostra lo schema di tale apparato.

L'intervallo di lunghezza d'onda entro il quale il sistema permette di studiare il comportamento di un sensore si estende da 145 nm a 1100 nm.

Vengono utilizzate due sorgenti di radiazione elettromagnetica: una lampada al deuterio che copre l'intervallo spettrale 145-180 nm (ultravioletto da vuoto o VUV) ed una allo xeno per le misure a λ compresa tra 200 nm e 1100 nm (UV, V, IR).

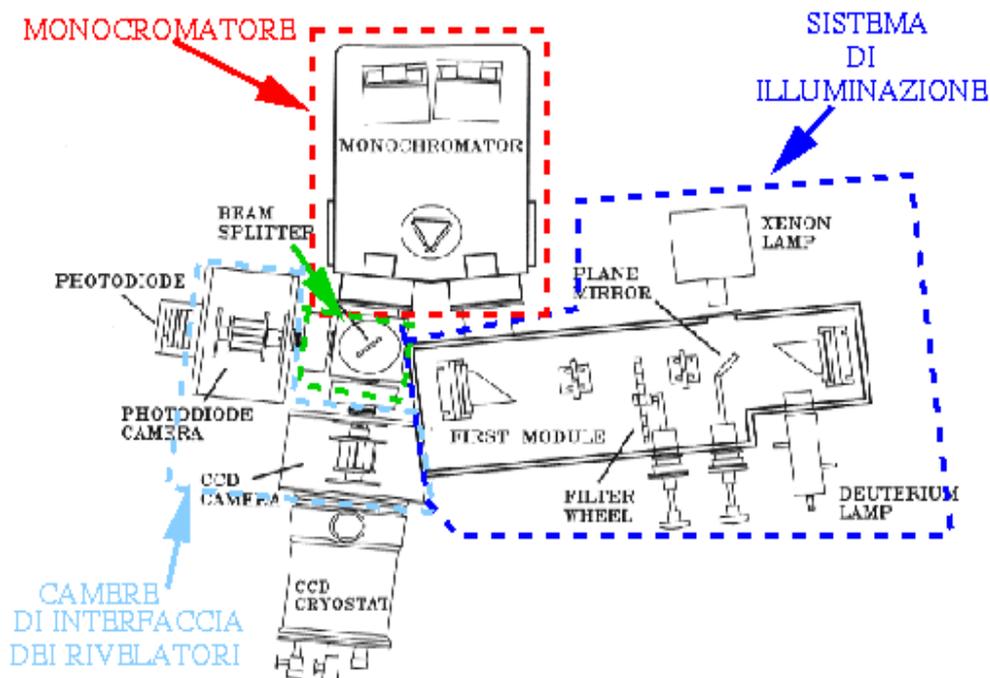


Fig. 1 – Schema del banco ottico.

OSSERVATORIO ASTROFISICO DI CATANIA

Il **primo modulo** ha la funzione di preselezionare la banda di ingresso al monocromatore (secondo modulo) e di bloccare la luce diffusa.

Il fascio emesso dalla lampada al deuterio incide su uno specchio parabolico, che lo rende parallelo, viene collimato da diaframmi, filtrato e quindi focalizzato sulla fenditura d'ingresso del monocromatore per mezzo di un secondo specchio parabolico.

Inoltre, all'uscita della lampada al deuterio, è posta una ruota portafiltri, azionata da un motore *step* gestito mediante il controller PS-35, prodotto dall'azienda tedesca Owis. Quest'ultimo è collegato al computer mediante interfaccia USB, ma viene visto dal sistema come un dispositivo seriale. Esso riceve le istruzioni di comando e, oltre ad eseguire i movimenti richiesti, restituisce dei codici di controllo numerici e alfanumerici, al fine di monitorarne il corretto funzionamento.

I raggi provenienti dalla lampada allo xeno sono paralleli; essi vengono filtrati mediante un sistema di ruote portafiltri, giungono su uno specchio piano inclinato a 45° e da qui sono indirizzati sullo specchio parabolico e diretti al secondo modulo.

Le ruote portafiltri poste all'uscita della lampada allo xeno sono mosse da motori DC, gestiti dal controller DC-500, prodotto anch'esso dalla Owis. Tale controller, molto più datato rispetto al precedente, è connesso direttamente alla porta seriale ed anch'esso, oltre ad eseguire le istruzioni richieste, restituisce, attraverso la stessa porta, i codici di controllo, per il monitoring del funzionamento.

Il **monocromatore (secondo modulo)** contiene due specchi sferici e tre reticoli (riflettenti) con un passo di 1200 tratti/mm, posizionati su una torretta, ciascuno dei quali è ottimizzato per un determinato intervallo di lunghezze d'onda. Il fascio luminoso focalizzato sulla fenditura d'ingresso incide sul primo specchio (collimatore), che lo rende parallelo e lo invia sul reticolo selezionato, ove viene disperso. Di qui, il fascio parallelo viene riflesso sul secondo specchio (focalizzatore), che proietta così lo spettro della radiazione sulla fenditura di uscita. Ruotando il reticolo, si fa scorrere lo spettro sulla fenditura e quindi si seleziona la λ prescelta centrata su di essa.

La larghezza delle fenditure, regolabile mediante viti micrometriche, può essere al massimo di 5 mm. Chiaramente quanto più larga è la fenditura di uscita, tanto più ampia è la parte dello spettro che si seleziona (e peggiore sarà pertanto la risoluzione).

Un altro controller gestisce i movimenti della torretta dei reticoli del monocromatore ed è collegato al PC attraverso la porta seriale. Tale controller è stato fornito insieme al monocromatore. Anch'esso, oltre ad attuare i movimenti, fornisce i codici di controllo per il monitoraggio del funzionamento.

All'uscita del monocromatore la radiazione incide su un **beam-splitter (terzo modulo)**, costituito da uno specchio semiriflettente di MgF_2 inclinato a 45° rispetto alla direzione del fascio incidente.

Quest'ultimo viene così in parte riflesso ed in parte trasmesso verso due camere gemelle (quarto e quinto modulo). Ciò permette di effettuare misure contemporanee su due sensori: uno utilizzato come sensore campione ed uno come sensore da caratterizzare; la contemporaneità delle misure permette di monitorare eventuali variazioni del flusso luminoso. In entrambe le camere il fascio luminoso passa attraverso lenti di MgF_2 . Tali lenti scorrono su di una guida, permettendo la messa a

OSSERVATORIO ASTROFISICO DI CATANIA

fuoco sui rivelatori al variare della lunghezza d'onda. L'intervallo entro cui le lenti possono muoversi è di circa 10 cm e la loro posizione è variata mediante un "focusing controller" che le sposta per step da 0 a 50000; la posizione 0 è quella in cui la lente è alla massima distanza dal rivelatore.

Del *focusing* si occupa lo stesso controller già citato in precedenza, cioè il PS-35 della Owis.

Tutte queste parti di codice, che costituiscono il back-end della suite di comandi, sono stati inglobati in un'interfaccia grafica mediante *NetBeans*, l'ambiente di sviluppo open-source e multi-linguaggio creato dalla *Sun Microsystems* per scrivere, compilare ed eseguire il *debug* ed il *deploy* di programmi.

L'intero progetto è suddiviso in 3 sezioni: *Main*, *Monitor* e *Settings*.

La sezione *Main* contiene il vero e proprio software di controllo motori con la relativa parte grafica (pannelli e pulsanti di comando).

Ecco la schermata del pannello principale:

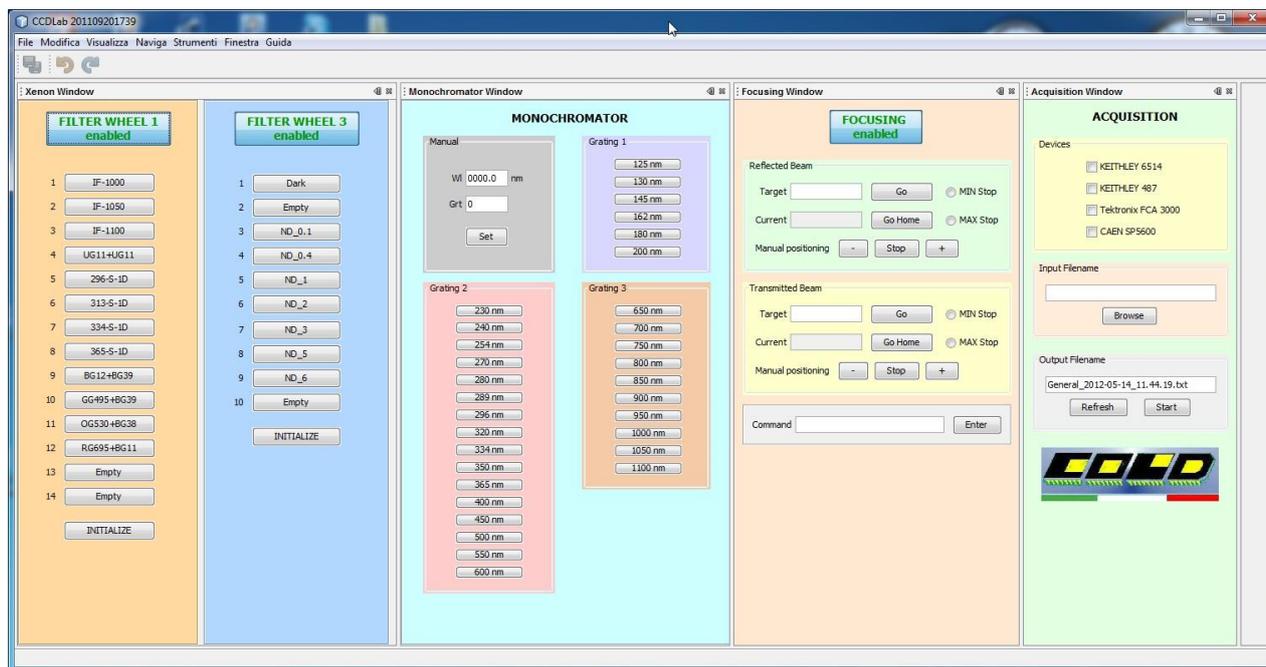


Fig. 2 – Il pannello principale.

Si notano i pulsanti di comando delle ruote portafiltri con le varie posizioni, nonché i pulsanti del monocromatore con le relative lunghezze d'onda da selezionare. E' presente anche un pannello per la selezione manuale, nel caso si debba selezionare una lunghezza d'onda diversa da quella preconfigurata nei vari pulsanti.

OSSERVATORIO ASTROFISICO DI CATANIA

Accanto al pannello del monocromatore si trova quello del *focusing*, che contiene i campi di testo dove si immettono le posizioni (da 0 a 50000 step) delle lenti e i pulsanti di avvio del posizionamento. E' prevista anche la possibilità di spostarsi manualmente senza dover immettere una posizione determinata (pulsanti “-“ e “+”).

Ogni modulo di comando è dotato di una propria finestra di monitoraggio, che permette di controllarne il regolare funzionamento attraverso i log mostrati nell'area di testo; tutto ciò fa parte della sezione Monitor.

Di seguito uno screenshot di un pannello di monitoraggio:

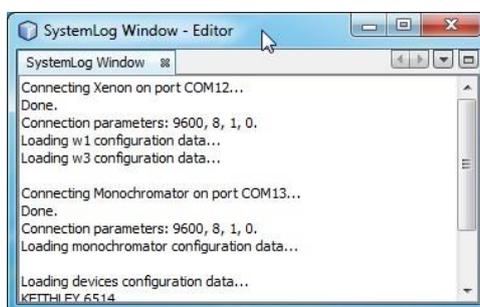


Fig. 3 – Uno dei pannelli di monitoraggio.

La terza e ultima sezione si occupa della configurazione delle varie porte; ogni pannello di configurazione salva i parametri in un file di testo, che viene poi richiamato da ciascun modulo per instaurare la comunicazione con la relativa porta.

Ecco come si presenta uno dei pannelli di settaggio:

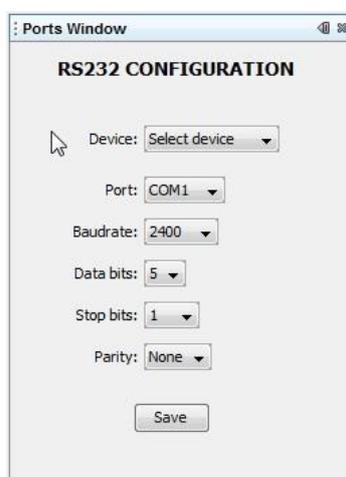


Fig. 4 – Il pannello settaggi RS232.

OSSERVATORIO ASTROFISICO DI CATANIA

3 - L'acquisizione dati

Il modulo *Main* descritto precedentemente, oltre ad occuparsi del controllo dei motori del banco ottico, è in grado, mediante un automatismo, di gestire anche l'acquisizione dei dati prodotti dalla strumentazione di misura. In particolare, dopo il posizionamento delle ruote portafiltri e del monocromatore per selezionare una determinata lunghezza d'onda, inizia la fase di misura e la conseguente acquisizione dei dati utili alla caratterizzazione del sensore.

Di ciò si occupa un apposito modulo che richiama i vari strumenti coinvolti nella misurazione, gestito da una propria classe Java.

Tali classi sono indipendenti tra loro, ma intercomunicanti attraverso un'ulteriore classe che coordina ed elabora i file generati da tali strumenti.

Il risultato finale è un file di testo che contiene le misurazioni di tutti gli strumenti, opportunamente formattato in modo da essere aperto da un foglio di calcolo, ad es. Microsoft Excel.

Gli strumenti di misura hanno solo interfacce di comunicazione di tipo GPIB e per effettuare la connessione fisica tra il computer e la strumentazione è stato utilizzato l'adattatore "Agilent 82357A GPIB/USB Interface Converter", per il quale sono state fornite dalla Agilent le API di comunicazione (VISA) sia per il linguaggio C che per il linguaggio Visual Basic. Di conseguenza ho dovuto realizzare (in C) i metodi nativi JNI che richiassero quelli delle API da un lato, e che si interfacciassero con il *wrapper* Java dall'altro lato.

Ecco di seguito le immagini dei vari strumenti di misura utilizzati e delle relative interfacce di controllo:

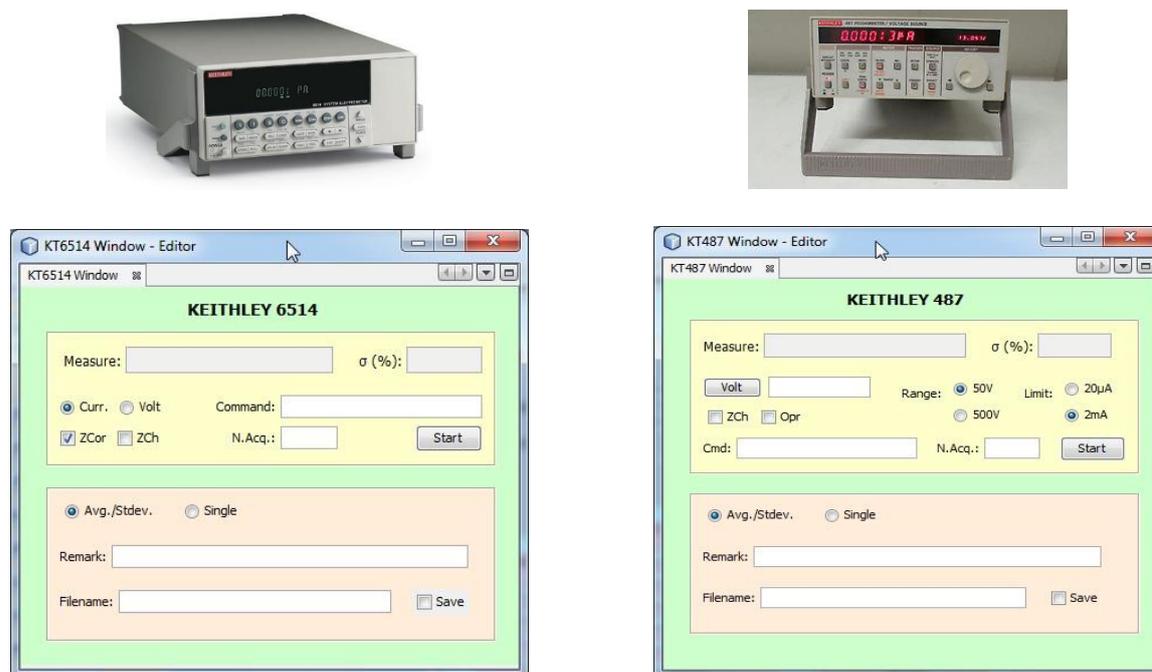


Fig. 5 – I picoamperometri KEITHLEY 6514 e 487 e relative interfacce.

OSSERVATORIO ASTROFISICO DI CATANIA

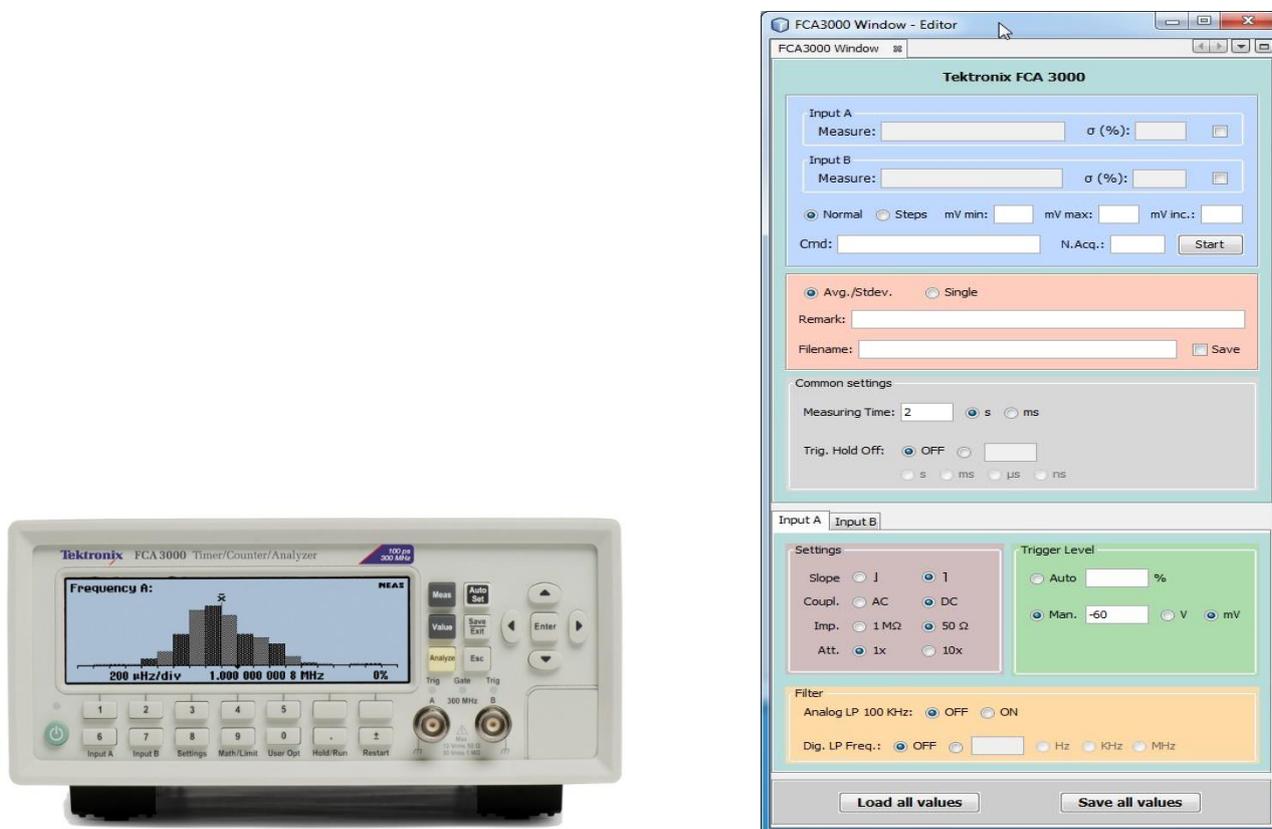


Fig. 6 – Il contatore Tektronix FCA 3000 e la relativa interfaccia.

Le interfacce dei due picoamperometri Keithley 6514 e 487 sono molto simili tra loro e consentono di effettuare misure in serie impostando il numero di acquisizioni sull'apposito campo di testo e premendo il pulsante "Start". Sono presenti anche delle caselle di spunta e dei pulsanti radio per effettuare i vari settaggi previsti in ognuno dei due strumenti. Infine, è possibile salvare su file i risultati acquisiti.

Più complessa, anche se solo apparentemente, appare l'interfaccia del contatore Tektronix FCA 3000. Essendo esso uno strumento dalle enormi funzionalità, si è dovuto pensare ad un'interfaccia capace di gestirlo al meglio e semplice nello stesso tempo.

Tale interfaccia è suddivisa in pannelli: il primo si occupa dell'acquisizione vera e propria, che può essere singola, multipla o per step prestabiliti a diversi intervalli di soglia di tensione. Il secondo pannello gestisce il salvataggio dei dati acquisiti su file, mentre il terzo pannello serve a settare dei parametri comuni ad entrambi i canali di ingresso dello strumento. Il quarto pannello è a sua volta suddiviso in due tab identici, uno per ciascun canale di ingresso; ogni tab riporta i controlli presenti fisicamente sul pannello frontale del contatore e permette di effettuare le medesime regolazioni mediante dei pulsanti radio. Infine, in basso sono presenti due grandi pulsanti: il primo serve ad importare nel programma i settaggi dello strumento, mentre il secondo a salvare sullo strumento i settaggi effettuati dall'utente.