



UNIVERSITÀ  
DEGLI STUDI  
DI PALERMO

---

Facoltà di Ingegneria  
Corso di Laurea in Ingegneria Elettronica

*Realizzazione di un sistema di  
caratterizzazione per memristor*

*Tesi di laurea di:*  
**Alessandro D'Angelo**  
(matr. 0565411)

*Relatore:*  
**Ing. Mauro Mosca**

*Correlatore:*  
**Ing. Fulvio Caruso**

---

**Anno Accademico 2012-2013**

# INDICE

<b>SOMMARIO</b>	pag.1
<b>CAPITOLO 1 CENNI TEORICI SUI MEMRISTOR</b>	pag.2
<b>CAPITOLO 2 DESCRIZIONE DELL'HARDWARE</b>	
2.1 PROGETTAZIONE DI UN ALIMENTATORE DUALE	pag.7
2.2 PROGETTAZIONE DELLA SCHEDA PER IL CONTROLLO DELLA RAMPA DI TENSIONE	pag.10
2.3 REALIZZAZIONE DELL'ALIMENTATORE DUALE E DELLA SCHEDA PER IL PILOTAGGIO DEL POTENZIOMETRO AD5290	pag.13
<b>CAPITOLO 3 DESCRIZIONE DEL SOFTWARE</b>	
3.1 INTRODUZIONE A LABVIEW	pag.17
3.2 MENU' PRINCIPALE	pag.19
3.3 OPERAZIONI PRELIMINARI	pag.20
3.4 PROGRAMMA PER LA GENERAZIONE DI UNA RAMPA DI TENSIONE	pag.23
<b>CAPITOLO 4 CARATTERIZZAZIONE DEI MEMRISTOR CON STRUTTURA A PAD</b>	
4.1 LAYOUT UTILIZZATO PER LA REALIZZAZIONE DEI MEMRISTOR	pag.29
4.2 ESEMPIO DI CARATTERIZZAZIONE DEI MEMRISTOR	pag.31
4.3 CARATTERIZZAZIONE DI MEMRISTOR REALIZZATI CON VO <sub>2</sub> E ZnO	pag.34
<b>CAPITOLO 5 PROGETTAZIONE DEL LAYOUT PER LA REALIZZAZIONE DI MEMRISTOR CON GEOMETRIE PIU' AVANZATE</b>	pag.37
<b>CONCLUSIONI</b>	pag.39
<b>BIBLIOGRAFIA</b>	pag.40

## SOMMARIO

Il seguente lavoro di tesi è stato realizzato con lo scopo di implementare un sistema che fosse in grado di caratterizzare in modo completo i dispositivi memristivi. Questo è stato fatto per potere dare un contributo scientifico alla determinazione delle caratteristiche che permettano di realizzare nel migliore dei modi questi dispositivi. Il *memristor* per l'appunto sembrerebbe essere uno dei componenti fondamentali per le RAM di nuova generazione e proprio per questo motivo il mondo scientifico in questi anni ha incrementato le ricerche in merito.

Nel primo capitolo sono stati fatti alcuni cenni storici; dalla sua scoperta, che avvenne nel 1970 grazie allo scienziato Leo Chua, fino alla sua prima realizzazione, che avvenne molti anni dopo per mano di un team della HP. Inoltre è stato presentato il suo comportamento mnemonico, ovvero è stato mostrato un ipotetico ciclo di scrittura e lettura.

Nel secondo capitolo è stato presentato il *Memristor Data Acquisition System (MDAS)*, ovvero la componente hardware del sistema di caratterizzazione. In questo capitolo vengono mostrati i passi fondamentali che hanno portato alla progettazione e in seguito alla realizzazione dell' *MDAS*.

Nel terzo capitolo invece viene mostrata la componente software, del sistema di caratterizzazione, realizzata utilizzando il programma labVIEW. Viene appunto illustrato il codice utilizzato per implementare le funzioni desiderate e ne viene spiegato nella sua interezza il principio di funzionamento, in modo da rendere il lettore consapevole delle scelte effettuate.

Nel quarto capitolo è stata presentata nei minimi dettagli la procedura da applicare per effettuare la caratterizzazione del dispositivo memristivo, inoltre sono stati inseriti alcuni grafici relativi a delle misure effettuate su *memristor* realizzati utilizzando ossido di vanadio VO<sub>2</sub> o ossido di zinco ZnO.

Infine nel quinto ed ultimo capitolo è mostrata la struttura utilizzata per la realizzazione dei *memristor*. Si è deciso di utilizzare un software per la realizzazione di dispositivi VLSI, grazie al quale è stato possibile realizzare il layout e la geometria del *memristor*

## CAPITOLO 1 CENNI TEORICI SUI MEMRISTOR

Negli ultimi anni le crescenti richieste di mercato hanno portato ad un continuo incremento della capacità delle memorie non volatili. Attualmente si è arrivati quasi ad un punto di stallo poiché non si è più in grado di espandere sensibilmente la capacità di memorizzare dati consentendo contemporaneamente una riduzione delle dimensioni delle memorie. A tal proposito si sono cercati nuovi componenti e tecnologie da usare per implementare memorie dalle dimensioni ridottissime .

Nel 1970 Leon Chua teorizzò l'esistenza di un componente passivo [1], che in seguito si scoprì essere perfetto per gli scopi succitati di memorizzazione dei dati.

Questo componente, che si comporta come una resistenza non lineare, prende il nome di *memristor* (*memory resistor*) proprio per evidenziare il fatto che è possibile usarlo come elemento base di una memoria e sfruttare la variazione della sua resistenza a questo scopo. La caratteristica  $v - i$  del *memristor* è rappresentata in fig.1 e come si può notare essa descrive un ciclo d'isteresi al variare della tensione applicata.

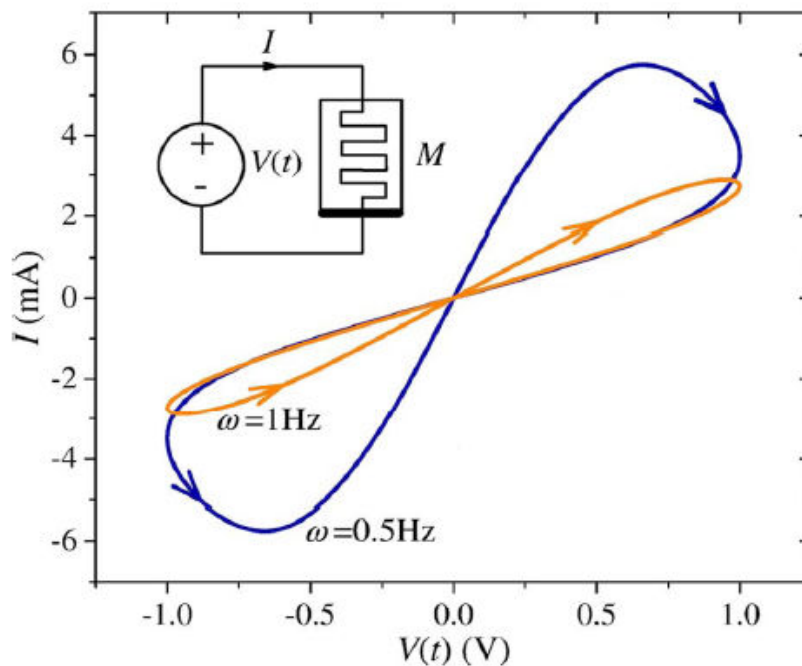


Fig.1 Ciclo di isteresi del *memristor*

Il grafico rappresenta il comportamento del *memristor* ipotizzato da Chua; infatti, per uno stesso

valore di tensione, si possono ottenere due valori di corrente diversi e quindi la resistenza offerta dal componente non è sempre la stessa. Se questa variazione è accettabile si possono discriminare due stati "ON" e "OFF" sulla base dei valori di resistenza del componente e quindi quest'ultimo può essere utilizzato per memorizzare un bit.

Di conseguenza il *memristor* può essere utilizzato come interruttore bistabile per memorie resistive ad accesso casuale (Resistive Random Access Memory, RRAM), anche se il suo funzionamento non è limitato a stati discreti, poichè è infatti possibile determinare uno spettro continuo memristivo, ovvero la memristenza del dispositivo può essere variata in modo controllato da una opportuna polarizzazione. Grazie alla facilità con la quale è possibile ridurli a scale nanometriche [2] i *memristor* possono essere usati come memorie non volatili, consentendo inoltre l'incremento della densità di immagazzinamento di dati minimizzando notevolmente gli spazi (fornendo così un'estensione alla legge di Moore [6]).

Un dispositivo con un comportamento simile a quello a suo tempo proposto da Chua venne realizzato per la prima volta nel 2008 da un gruppo di scienziati della Hewlett-Packard (HP) che costruirono il primo prototipo di *memristor* [3], la fig.2 ne illustra la struttura. Questa era formata da uno strato di platino (Pt) (*bottom contact*) sul quale veniva fatto accrescere un primo film di ossido di titanio con un'insufficienza di ossigeno ( $TiO_{2-x}$ ) e un secondo film di ossido di titanio ( $TiO_2$ ). Successivamente il contatto superiore veniva realizzato nuovamente con il platino (*top contact*) fig.3. Stanley Williams, ricercatore della HP, si accorse che applicando una tensione variabile a tale dispositivo, il grafico  $v - i$  che si veniva a formare rispecchiava solo parzialmente la teoria di Chua infatti, sebbene il risultato fosse un grafico d'isteresi, esso non era del tutto uguale a quello ipotizzato, come si può notare in fig.2.

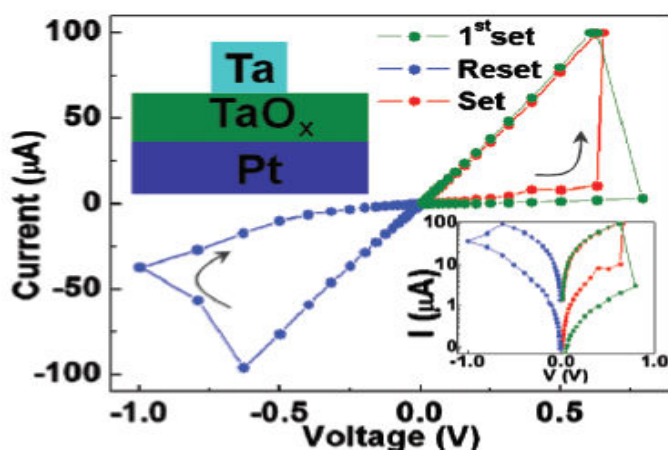


Fig.2 grafico di isteresi reale del *memristor*

Questa differenza nell'andamento del grafico è dovuta al comportamento fisico del dispositivo stesso: infatti mentre la giunzione Pt/TiO<sub>2-x</sub> è sempre ohmica, quella Pt/TiO<sub>2</sub>, quando la tensione aumenta, si comporta da giunzione rettificante, producendo quindi un grafico non lineare; quando invece la tensione cambia di polarità e comincia a diminuire si comporta da giunzione ohmica, quindi produce un grafico lineare (come una resistenza). Questo cambiamento di natura della giunzione è dovuta alla migrazione delle vacanze di ossigeno che dal TiO<sub>2-x</sub> vanno verso il TiO<sub>2</sub> rendendolo drogato di tipo n e facendolo diventare con il platino una giunzione ohmica.

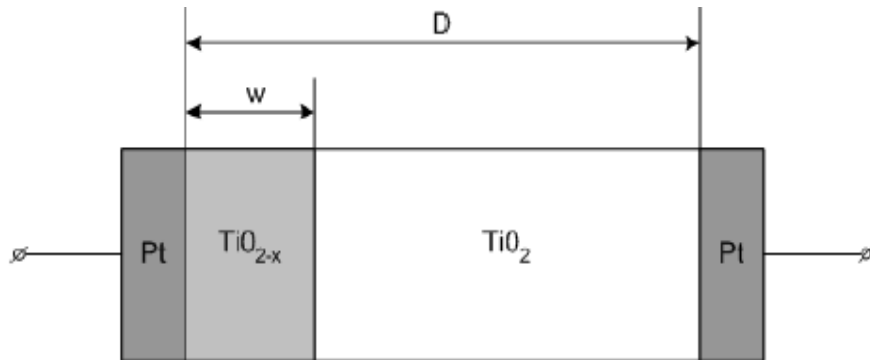


Fig.3 Prototipo di *memristor* realizzato dal team hp

Il metodo più semplice per scrivere una memoria a *memristor* è quello mostrato in fig.4

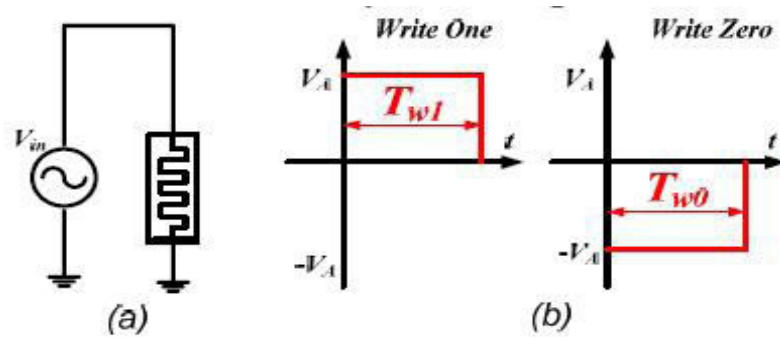


Fig.4 Ciclo di scrittura di una memoria a *memristor*

Chiamiamo  $w(t)$  la zona che dopo la migrazione di vacanze di ossigeno nel *memristor* risulta drogata e supponiamo, ad esempio, di voler partire da un livello logico 0 ( $w(t)=0$ ) e di voler scrivere un bit pari ad 1 ( $w(t)=D$  ovvero la distanza necessaria affinché sia presente un canale conduttivo). Fornendo un impulso di ampiezza  $V_a$  e periodo  $T_{w1}$ , se :

$$T_{wl} > \frac{\phi}{V_a}$$

con :

$$\phi = \beta \frac{D^2}{2\mu v}$$

allora verrà memorizzato nel memristor un 1 logico ovvero: un periodo dell'impulso maggiore del tempo  $T_{wl}$  garantisce una corretta scrittura del valore desiderato, viceversa se l'impulso ha un periodo minore di  $T_{wl}$  il dato non è memorizzato.

L'operazione di lettura invece è rappresentata in fig.5 :

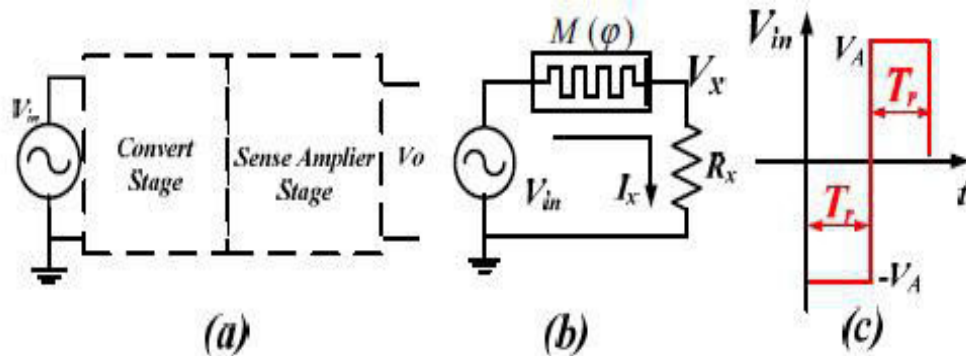


Fig.5 Ciclo di lettura di una memoria a memristor

dallo schema si può notare una resistore  $R_x$  in serie al memristor. Grazie ad esso è possibile convertire lo stato all'interno della memoria in un segnale di tensione, essendo la corrente che scorre nel memristor a portare l'informazione. La tensione al nodo X sarà data da :

$$V_x(t) = V_{in}(t) \frac{R_x}{R_x + M(\Phi_{in} - \Phi_x)}$$

la resistenza  $R_x$  deve essere adeguatamente progettata in quanto un errato dimensionamento potrebbe portare ad una lettura sbagliata del valore logico presente nella memoria. Sapendo che quando la zona drogata si trova ad un valore minore di  $D/2$  la  $V_x$  assume un valore minore della metà di  $V_i$  e che quando  $W(t)$  supera  $D/2$  avviene il viceversa, possiamo porre la  $R_x = R_{off}/2$  in quanto:

- Quando il memristore ha memorizzato uno zero logico,  $M(\varphi)$  è circa uguale a  $R_{off}$ , e l'uscita  $V_x$  essendo minore di  $V_{in}$  comporta proprio la lettura di uno zero.
- Quando invece il memristore ha memorizzato un uno logico,  $M(\varphi)$  è circa uguale a  $R_{on}$ , e l'uscita  $V_x$  è maggiore di  $V_{in}$ , e dunque viene letto un uno.

Il *memristor*, oltre ad essere utilizzato come semplice memoria, potrebbe tornare utile anche per scopi diversi a livello scientifico: Ad esempio da studi recenti si è ipotizzato che esso possa aiutare a simulare le sinapsi (elemento di contatto fra un neurone e un altro) del cervello umano [7]. L'intelligenza artificiale per l'appunto è un traguardo ancora molto lontano e riprodurre il funzionamento del cervello con i calcolatori moderni sembrerebbe un'operazione impossibile. Il limite invalicabile che ad oggi si presenta è insito nella natura stessa dei calcolatori i quali rispettano dei canoni legati all'architettura di Von Neumann: in essi, infatti, vi è una netta separazione tra la parte dedicata alla memoria e quella relativa agli elementi di calcolo. Inoltre nei calcolatori qualunque operazione viene scandita dall'esecuzione in serie di istruzioni basilari che rendono tutto il sistema di tipo sequenziale. Questo comportamento non è minimamente paragonabile con l'attività del cervello umano, nel quale i neuroni svolgono in parallelo e con contemporaneità compiti di calcolo e di accesso alla memoria.

Tuttavia, la recente scoperta del *memristor* ha aperto nuove porte al mondo scientifico poichè esso è ciò che di più simile ad una sinapsi si possa trovare al momento in campo elettronico. Più precisamente, dato che la capacità di memoria di questo nuovo componente è legata alla resistenza da esso offerta, la quale dipende sostanzialmente dalla quantità di carica che lo ha attraversato, esso simula egregiamente una sinapsi, la cui propensione a trasmettere i segnali è legata alla storia delle precedenti eccitazioni.

Il *memristor* inoltre ha il pregio di essere una memoria non volatile, poichè il dato al suo interno non ha bisogno di una sorgente di alimentazione esterna per restare memorizzato. Infine, dato che può assumere un valore di resistenza in un intervallo continuo di valori, esso può essere usato come una memoria analogica in quanto non è limitato all' 1 e allo 0 logico delle memorie usuali.



## CAPITOLO 2 DESCRIZIONE DELL'HARDWARE

In questo capitolo verrà presentato il progetto di un alimentatore duale stabilizzato utilizzato per effettuare le misure su un *memristor*, in seguito verrà descritto tutto il setup di misura utilizzato a tale scopo.

### 2.1 PROGETTAZIONE DI UN ALIMENTATORE DUALE

L'alimentatore è stato realizzato per la caratterizzazione del *memristor*, poichè come si evince dalla letteratura [8] per riuscire ad apprezzare un effetto *memristivo* bisogna pilotare il componente con una rampa di tensioni duali.

Esso è stato progettato prendendo spunto da una *application note* relativa allo stabilizzatore di tensione LM317 della Texas Instruments.

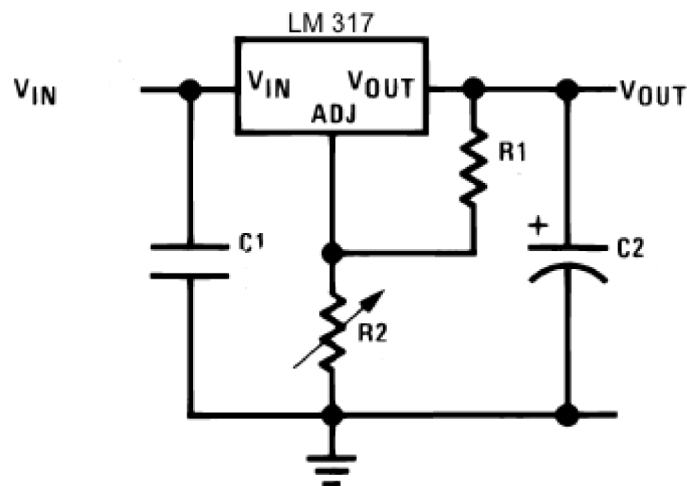


Fig.6 Application note LM317

In questo modo è possibile, a partire da una tensione di ingresso non stabilizzata, ottenere una  $V_{out}$  con minimo ripple e con la quale si può alimentare un qualsiasi circuito elettronico. La tensione di uscita è calcolata in modo teorico con la seguente formula :

$$V_{out} = V_{ref} \left( 1 + \frac{R_2}{R_1} \right) + I_{adj} * R_2$$

essendo  $V_{ref} = 1,25$  V una tensione di riferimento che è presente sempre tra il piedino di output e quello di adjust dell' LM317.

I soli suggerimenti riportati nell'application note non sono sufficienti per il corretto dimensionamento dell'alimentatore. Infatti lo schema elettrico reale che in seguito è stato implementato è quello di fig.7.

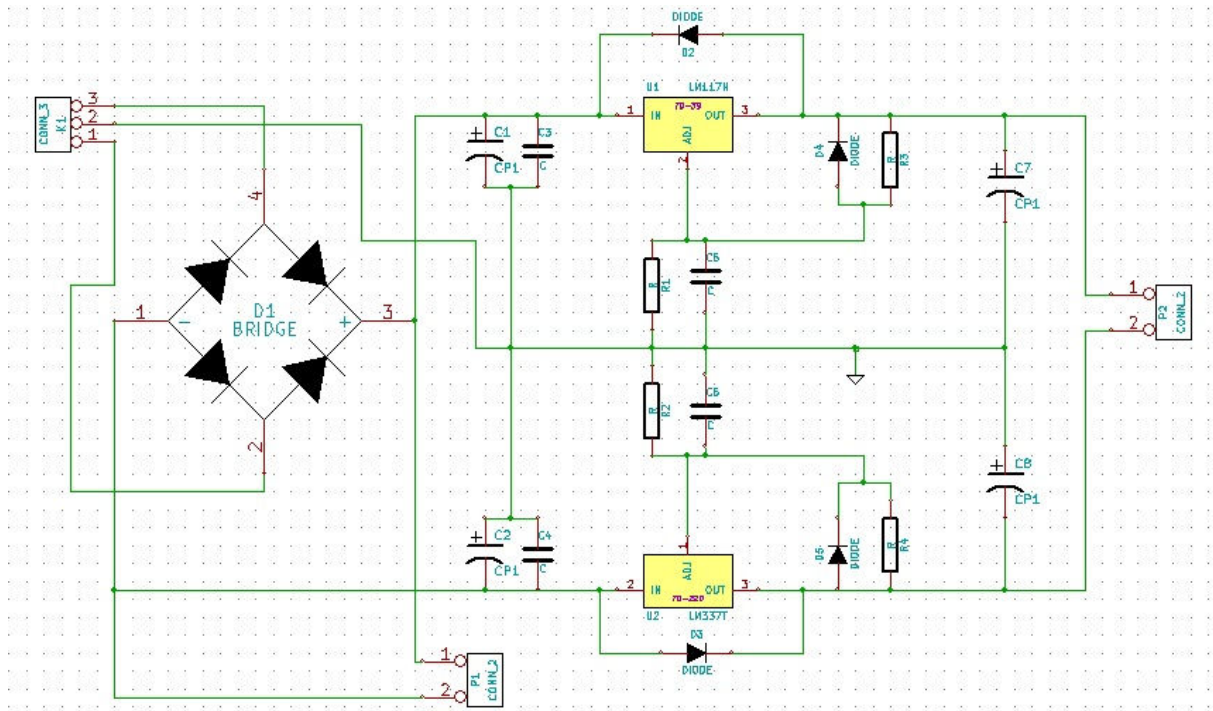


Fig.7 Schema elettrico alimentatore duale

Il seguente schema è stato realizzato con l'ausilio di KiCad, un software per la realizzazione di circuiti elettronici, ed anche la relativa *Printed Circuit Board* (PCB) è stata realizzata con il medesimo programma fig.8

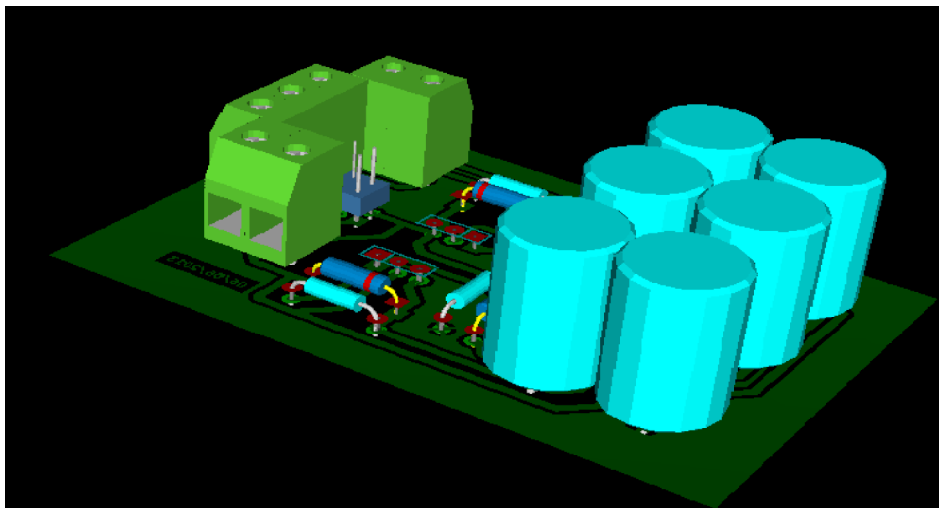


Fig.8 Visualizzazione 3D del PCB dell'alimentatore

Come si può notare dallo schema circuitale, il trasformatore ha il compito, tramite i suoi avvolgimenti con adeguato numero di spire, di portare il livello di tensione da quello di rete (220 V efficaci) a un valore di 15 V efficaci. La tensione però in questo tratto è ancora alternata ovvero oscilla sinusoidalmente tra +15 V e -15 V con una frequenza di 50 Hz. Per questo motivo essa è inviata ad un ponte a diodi di Graetz che ha il compito di raddrizzarla ovvero di far passare soltanto le semionde positive o soltanto quelle negative. Per l'appunto, grazie al trasformatore a presa centrale, al nodo 3 (fig.7) troveremo soltanto le semionde positive mentre al nodo 4 troveremo soltanto le semionde negative. Successivamente i due segnali complementari vengono filtrati tramite due condensatori dimensionati in modo tale da ottenere in uscita di essi una tensione pressoché continua (verificata collegando le uscite dei condensatori all'oscilloscopio). In seguito il segnale è mandato allo stabilizzatore di tensione LM317 (o LM337 nel caso di tensioni negative) che è in grado di fornire in uscita una tensione che assume valori compresi tra 1.25 V ( $V_{ref}$ ) e 37 V con una corrente massima erogabile di 1,5 A.

Come si può notare dalla formula su citata la tensione d'uscita dipende fortemente dalla scelta delle resistenze R1 ed R3 (R2 e R4) che in questo caso sono state dimensionate di valore rispettivamente 1,6 k $\Omega$  e 220  $\Omega$ . Il motivo di tale scelta è dovuta al fatto che il regolatore LM317 ha bisogno, per funzionare in modo corretto, di un *voltage drop*, ossia una caduta di tensione tra ingresso e uscita, di almeno 3 V. Con questi valori di resistenza infatti il valore della tensione di uscita è di circa 11 V anche se in realtà bisogna considerare anche il contributo dovuto alla  $I_{adj}$  ( $\approx 50\mu\text{A}$ ) moltiplicata per la R2 (che però risulta essere molto piccolo,  $\approx 0,5\text{V}$ ). In questo modo tra ingresso e uscita vi è una differenza di tensione di circa 9,5V e il *voltage drop* minimo viene rispettato denotando un corretto funzionamento dell' LM317 (o LM337).

Tra il piedino di output e massa di entrambi i regolatori è stato posto un condensatore elettrolitico di 100  $\mu\text{F}$  per migliorare la risposta al transitorio; inoltre tra il piedino di adjust e massa è stato posto un altro condensatore opzionale che ha il compito di migliorare la reiezione al ripple di tensione. Esso infatti è inserito per evitare che eventuali ripple di tensione vengano amplificati provocando problemi ai dispositivi connessi.

Dal connettore P2 è possibile prelevare la tensione stabilizzata ed impostata ad un valore di circa 11,5V, che in seguito servirà per generare una rampa di tensione in grado di pilotare il *memristor*. Infine è stato previsto un ulteriore connettore (P1) dal quale è possibile prelevare la tensione all'uscita del ponte a diodi, ovvero subito dopo essere stata raddrizzata e prefiltrata. Questo connettore è stato inserito per dare la possibilità di aggiungere un ulteriore circuito stabilizzatore simile a quello di fig.7 dal quale generare, all'occorrenza, dei valori di tensione diversi da quelli previsti dall' alimentatore realizzato.

## 2.2 PROGETTAZIONE DELLA SCHEDA PER IL CONTROLLO DELLA RAMPA DI TENSIONE

Per poter caratterizzare il *memristor* bisogna generare una rampa di tensione in modo che possa essere visibile il ciclo d'isteresi che ne dovrebbe risultare.

Per far ciò è stato utilizzato un potenziometro digitale della Analog Device e nella fattispecie il modello AD5290. Esso è un reostato programmabile a 8 bit in grado quindi di discriminare 256 livelli di tensione in modo sequenziale, la resistenza può essere variata da un valore di 0  $\Omega$  ad uno di 10 k $\Omega$  in maniera lineare. Il package dell' AD5290 è di tipo *Mini Small Outline Package* (MSOP-10) e ,per via delle ridotte dimensioni, è stato necessario costruire un adattatore tale da rendere il footprint simile a quello di un più comune DIP-10. La piedinatura di quest'ultimo è rappresentata in fig.9

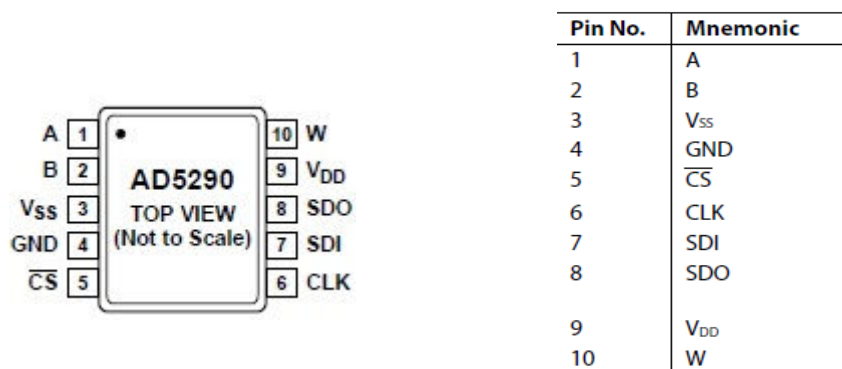


Fig.9 Piedinature del potenziometro AD5290

Affinchè l'AD5290 funzioni correttamente per le operazioni duali deve essere alimentato con valori compresi tra  $\pm 10V$ , e  $\pm 15V$ . La funzionalità di incremento e decremento della resistenza del potenziometro viene gestita attraverso un registro chiamato *wiper*, nel quale è presente il valore binario associato alla posizione spaziale del piedino variabile del reostato. Per spostare il *wiper* (piedino variabile) infatti basta inviare al potenziometro un *bitstream* che rappresenta il valore binario di uno dei 256 livelli possibili. Risulta necessario che questo segnale venga cadenzato da un clock e l'AD5290 deve essere abilitato a ricevere un dato con un impulso di CS. Il principio di funzionamento di tale potenziometro è esplicitato in fig.10

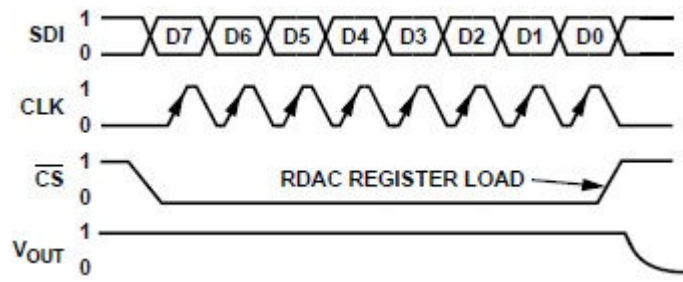


Fig.10 Bite di comando per potenziometro AD5290

Un'immagine virtuale della scheda per il pilotaggio del memristor è riportata in fig.11

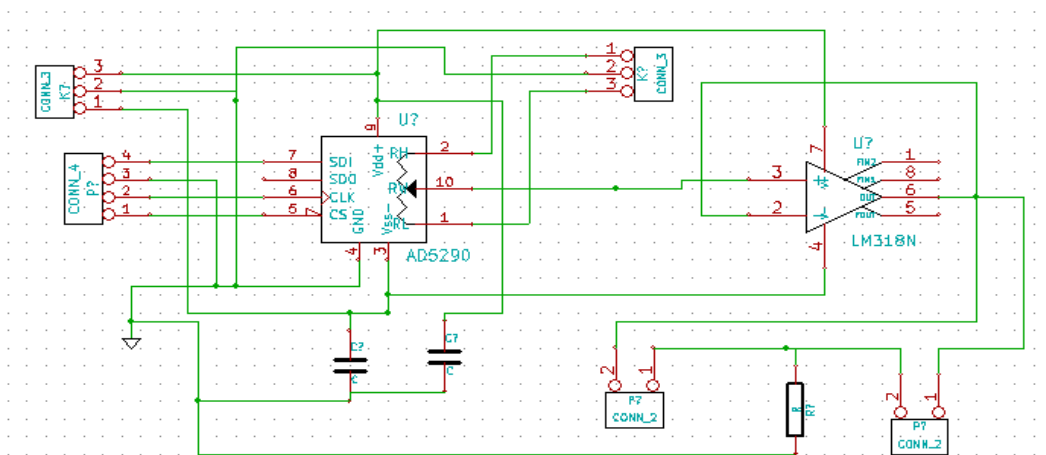
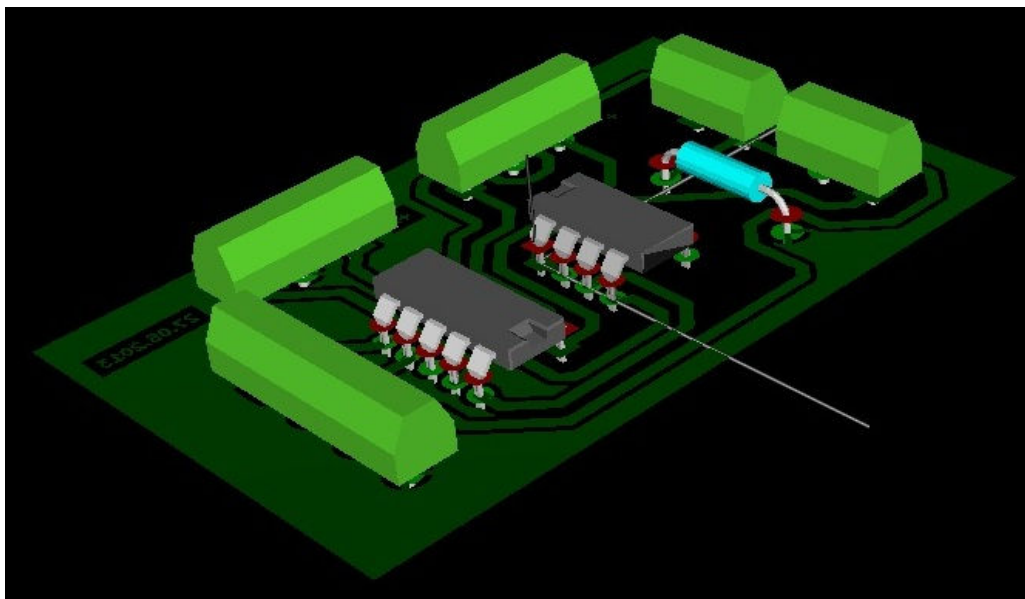


Fig.11 Schema elettrico e visualizzazione 3D della scheda per il potenziometro AD5290

Il potenziometro viene alimentato a +11,5 V -11,5 V attraverso uno dei connettori a tre piedini e inoltre sfruttando le medesime alimentazioni si forniscono ai capi (A, B) della resistenza interna circa 23 V denotando così un passo minimo di circa 90 mV. Questo valore è stato trovato dividendo i 23 V presenti sulla R per i possibili livelli discriminabili con l'AD5290 (256). I segnali di controllo del potenziometro vengono generati tramite software e inviati ad esso attraverso una scheda di acquisizione (DAQ6009) della Texas Instruments (che verrà discussa più in dettaglio nel capitolo 3). Come si nota dalla figura nella scheda oltre al potenziometro è presente anche un amplificatore operazionale (MC33076P1) alimentato anch'esso tra +11,5 V e -11,5 V. Quest'ultimo è stato inserito per fornire la corrente necessaria al pilotaggio del *memristor*, essa si aggira infatti intorno ai 150 mA. L'MC33076P1 per l'appunto è un amplificatore operazionale di potenza in grado di erogare fino a 250 mA.

La resistenza posta nella scheda ha invece il compito di rendere quanto più veritiera possibile la lettura della tensione sul *memristor*. Se infatti il valore di quel resistore non fosse adeguato esso potrebbe creare un effetto di partizione con il *memristor* stesso falsando quindi la relativa misurazione. In questo progetto è stata scelta una resistenza di shunt da 1  $\Omega$  con potenza massima dissipabile di 1 W (per evitare l'eccessivo riscaldamento della stessa). Questa scelta è dovuta al fatto che, a seguito di approfondite ricerche bibliografiche, si è arrivati alla conclusione che la resistenza offerta dal *memristor* si debba aggirare intorno ai 100  $\Omega$  e quindi l'eventuale effetto di partizione che si viene a determinare può essere, con buona approssimazione, trascurato. Sono stati inoltre inseriti due condensatori da 100 nF (non presenti nella visualizzazione 3D di fig.8) tra i piedini di alimentazione e massa in modo tale da disaccoppiare le alimentazioni degli integrati. Nella scheda sono presenti cinque connettori di cui, due sono stati utilizzati per collegare la DAQ all'AD5290, uno è quello dal quale arriva l'alimentazione ed un altro è quello che denota la tensione ai capi del potenziometro. L'ultimo connettore infine lega la scheda al *memristor*. Da esso infatti è possibile, attraverso dei connettori BNC, collegarsi ad una *probe station* tramite la quale in seguito verranno effettuate le misurazioni sul campione.

## 2.3 REALIZZAZIONE DELL'ALIMENTATORE DUALE E DELLA SCHEDA PER IL PILOTAGGIO DEL POTENZIOMETRO AD5290

La componentistica usata per l'alimentatore è di seguito riportata :

- LM317 (per le tensioni positive) Texas Instruments, LM337 (per le tensioni negative) Farchild
- Trasformatore a presa centrale 220V → +15V,-15V
- Raddrizzatore a ponte a diodi
- Condensatori elettrolitici da : 330  $\mu$ F (C1,C2), 100  $\mu$ F (C7,C8), 10  $\mu$ F (C5,C6)
- Condensatori ceramici da 100 nF (C3,C4)
- Resistori da : 220  $\Omega$  (R3,R4), 1,6 k $\Omega$  (R1,R2)
- Connettori serrafilo a vite

I componenti utilizzati per la realizzazione della scheda per il pilotaggio dell'AD5290 sono:

- Potenzenziometro digitale da 10 k  $\Omega$  a 8 bit (AD5290)
- Amplificatore operazionale di potenza (MC33076P1)
- Resistenza di shunt da 1  $\Omega$  ed 1 W
- Connettori serrafilo a vite

Per entrambe le schede il processo di realizzazione è stato il medesimo e per questo motivo ne verrà descritto soltanto uno. Innanzitutto è stato realizzato lo schema elettrico come da progetto, e grazie al programma per il CAD di circuiti elettronici KICad è stato possibile realizzare il master del PCB definitivo della scheda in fig.12. Successivamente esso è stato stampato con una stampante laser su un foglio di carta speciale per circuiti stampati (*Press-n-Peel Blue Transfer Film*) e tramite l'uso di un comune ferro da stiro è stato trasferito il toner presente sul foglio su una basetta ricoperta di rame. Quest'ultima è stata immersa in una soluzione di cloruro ferrico il quale ha corrosso il rame nelle zone non ricoperte dal il toner creando così il circuito stampato definitivo fig.12.

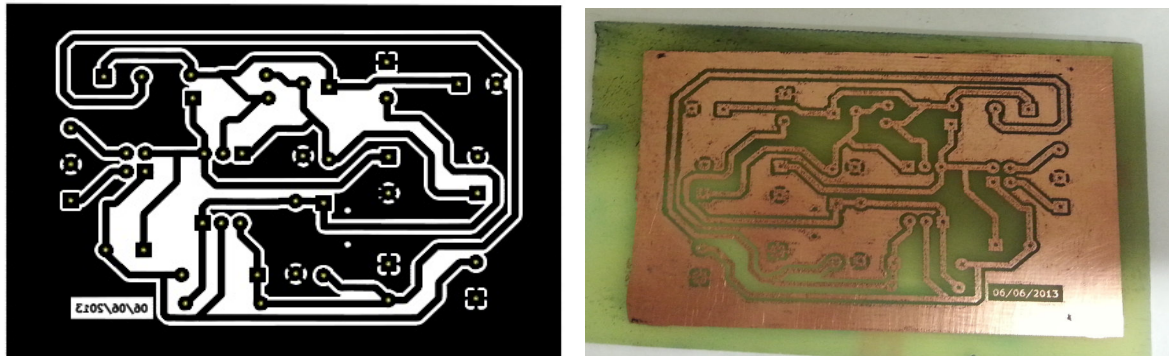


Fig.12 PCB dell'alimentatore e relativa realizzazione

Successivamente lo stampato è stato forato e ad esso sono stati saldati i vari componenti stando attenti a rispettare le piedinature degli stessi in corrispondenza delle piste nel PCB. Il risultato finale è rappresentato in fig.13.

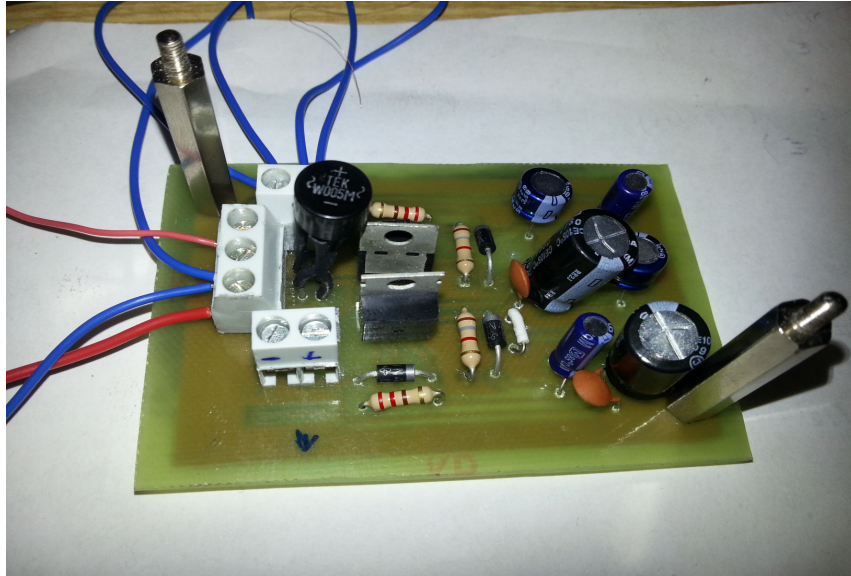


Fig.13 Alimentatore duale terminato

Infine entrambe le schede sono state poste, insieme al trasformatore e a svariati connettori, in una scatola per renderne più maneggevole l'utilizzo fig.14. Il circuito di misura così realizzato è stato denominato *Memristor Data Acquisition System (MDAS)*.

Il setup di misura, illustrato in fig. 15, è composto dall' *MDAS*, dalla DAQ6009, e il relativo PC per controllarla, e ovviamente da una *probe station*. Dopo aver connesso tramite l'apposito connettore la DAQ all' *MDAS* tramite i connettori BNC si collega quest'ultimo alla probe station. Infine dopo aver posto il campione a contatto con le punte è possibile tramite PC far partire una sessione di misura.



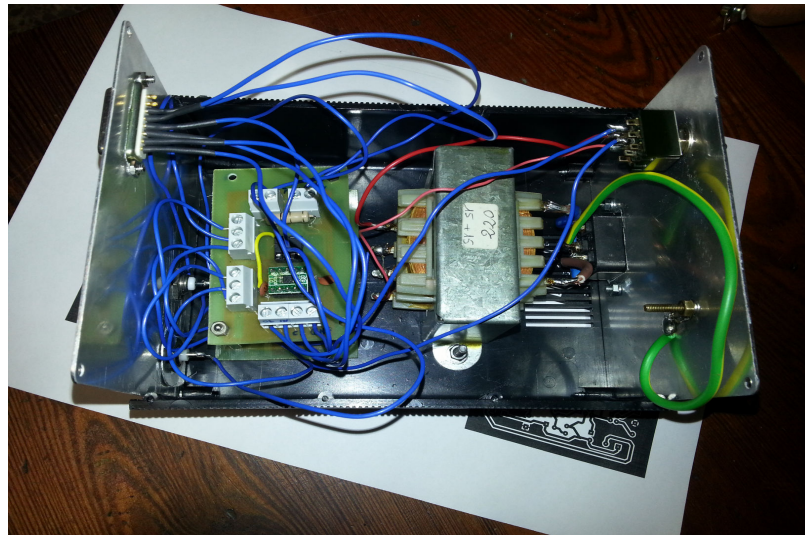


Fig.14 Memristor data acquisition system (MDAS)

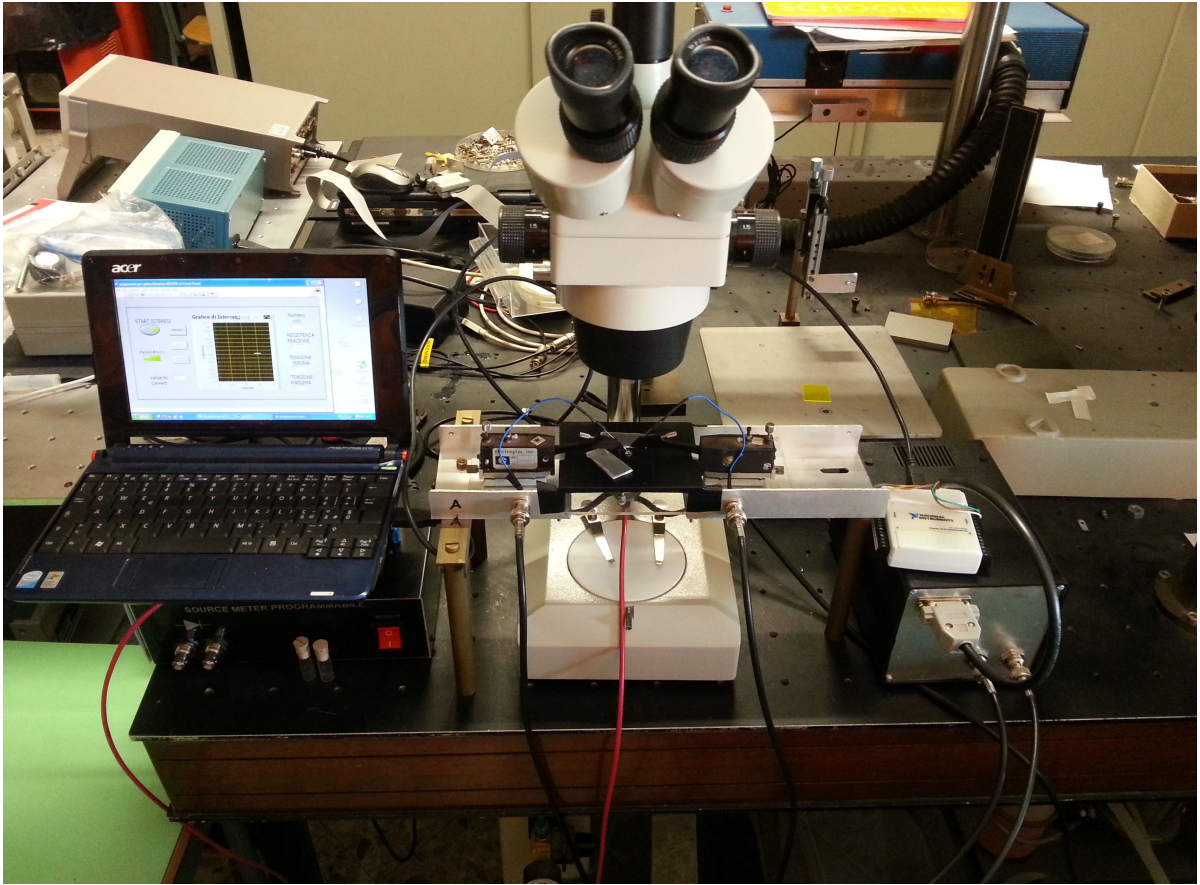


Fig.15 Setup completo di misura

## CAPITOLO 3 DESCRIZIONE DEL SOFTWARE

### 3.1 INTRODUZIONE A LABVIEW

LabVIEW è un ambiente di sviluppo integrato, ossia un software che aiuta il programmatore nello sviluppo di codice per nuove applicazioni. Il linguaggio grafico denominato “G” (*graphic language*) permette di realizzare programmi con un insieme di blocchi raffiguranti un'immagine descrittiva dell'operazione da essi compiuta. I vari blocchi sono connessi mediante dei fili virtuali che assumono un colore diverso in base al dato che vi transita fig.16.



Tipo di collegamento	Scalare	Array 1D	Array 2D	Colore
Numerico				Arancione (virgola mobile), Blu (intero)
Booleano				Verde
Stringa				Rosa

Fig.16 Tabella raffigurante i colori associati alle tipologie di dato

Un programma o sottoprogramma scritto in linguaggio G non esiste sotto forma di testo, come accade nei più tradizionali C, C++ e Pascal, esso esiste solamente come un file binario visualizzabile e compilabile con il solo LabVIEW. Un tipico software viene sviluppato costruendo un programma principale, chiamato *VI* (Virtual Instrument) nel quale è contenuto uno o più *subVI* dedicati allo svolgimento di funzioni più semplici, rendendo così la struttura più snella e scorrevole.

L'ambiente di sviluppo di ogni *VI* è composto da (fig.17) :

- Pannello Frontale (*Front Panel*)
- Diagramma a Blocchi (*Block Diagram*)

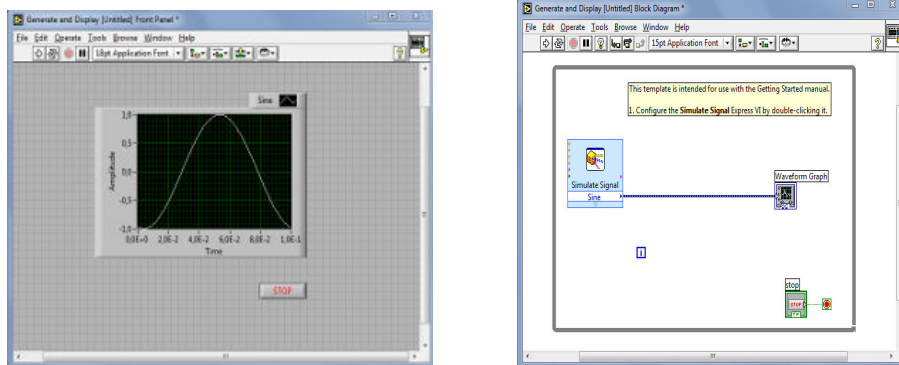


Fig.17 Sulla sinistra il *front panel*, sulla destra il *block diagram*

Il *front panel* è l'interfaccia utente ovvero il luogo nel quale si potranno trovare indicatori e controlli e nel quale si potranno aggiungere grafici per controllare segnali e variabili, mentre il *block diagram* è la finestra nella quale si trova il programma vero e proprio. Queste due finestre sono mutuamente collegate, infatti tutti gli oggetti presenti in una saranno presenti anche nell'altra, con la differenza che, mentre nel *block diagram* viene eseguito il programma nel *front panel* vengono visualizzati i risultati delle operazioni e si possono controllare delle funzionalità anche in fase di esecuzione.

Per inserire degli oggetti nel pannello frontale e nel diagramma a blocchi si utilizzano due tavole chiamate rispettivamente *controls palette* e *functions palette*.

Le strutture *case*, *for* e *while* comunemente utilizzate nei linguaggi di programmazione sono presenti anche in LabVIEW ma in questo caso le troveremo sotto forma di riquadri nei quali sarà possibile aggiungere tutti quei controlli che dovranno sottostare alla struttura stessa (fig.18).

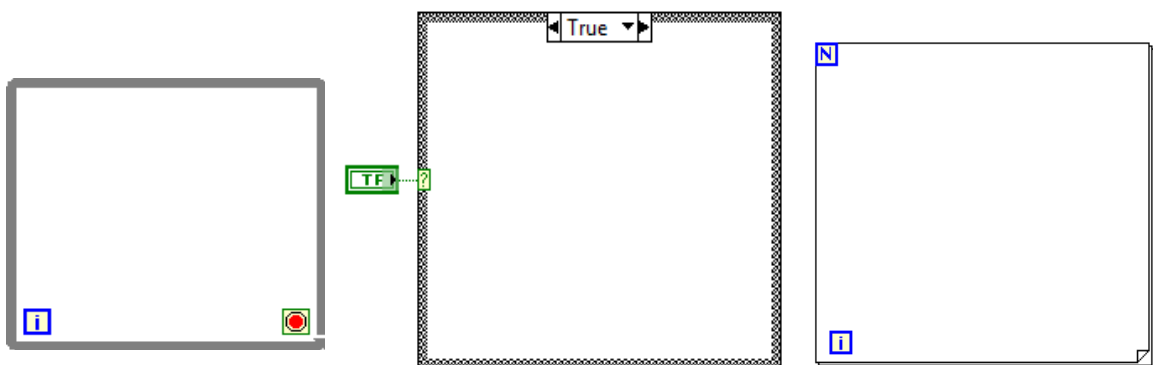


Fig.18 Da sinistra verso destra: ciclo while, struttura case, ciclo for

I driver *DAQmx* sviluppati da NATIONAL INSTRUMENTS permettono di configurare le schede di acquisizione di tipo *DAQ* utilizzando specifici *VI* chiamati *Express VI*. Esse sono presentate sotto forma di icone che il programmatore può utilizzare per interagire con le schede di acquisizione in modo agevole. L'*Express VI* maggiormente usato in questo lavoro di tesi è il *DAQ assistant* che mi ha permesso di comunicare con ma *NI DAQ-USB 6009*


Il software di pilotaggio realizzato in questo lavoro di tesi si articola in tre parti, il menù principale, il sotto-menù operazioni preliminari e il programma che permette di generare la rampa di tensione. Collegando infatti il pc alla *NI DAQ-USB 6009* è possibile tramite il *memristor data acquisition system (MDAS)*, di cui si è parlato nel capitolo 2, pilotare un potenziometro digitale per generare una rampa di tensioni duali.

### 3.2 MENU' PRINCIPALE

Il menù principale è stato progettato in modo tale da essere la prima schermata di interfaccia con cui l'utente finale si trova ad interagire. Da esso infatti è soltanto possibile passare alla finestra delle operazioni preliminari o decidere di far partire la rampa di tensioni. Di seguito è riportato il *front panel* relativo al menù stesso fig.19.



Fig.19 *Front panel* menù principale

Come si può notare in figura, oltre a potere compiere le operazioni preliminari e far partire un isteresi, l'utente può, attraverso il tasto "Accensione Generale", decidere di spegnere l'esecuzione in qualsiasi momento. Più interessante è però il *block diagram*, illustrato in fig.20, nel quale è presente il codice del programma. Il ciclo while presente in figura viene eseguito in modo ciclico nell'attesa che l'utente decida di compiere una delle azioni che sono state previste o che, al termine della misurazione, esso decida di chiudere il programma premendo il tasto "Accensione Generale". L'icona  ha il significato di "continue-if-true" ed essa non permette di uscire dal while fin quando non è stato premuto il tasto a lei connesso ("Accensione Generale"). Il selezionamento del tasto "Operazioni

*Preliminari*” determina la generazione di un segnale booleano di valore “true” che permette al programma di entrare nel riquadro di sinistra e quindi di abilitare il funzionamento della subroutine selezionata.

Allo stesso modo premendo il tasto “*Isteresi*” è possibile richiamare la subroutine ad esso associata.

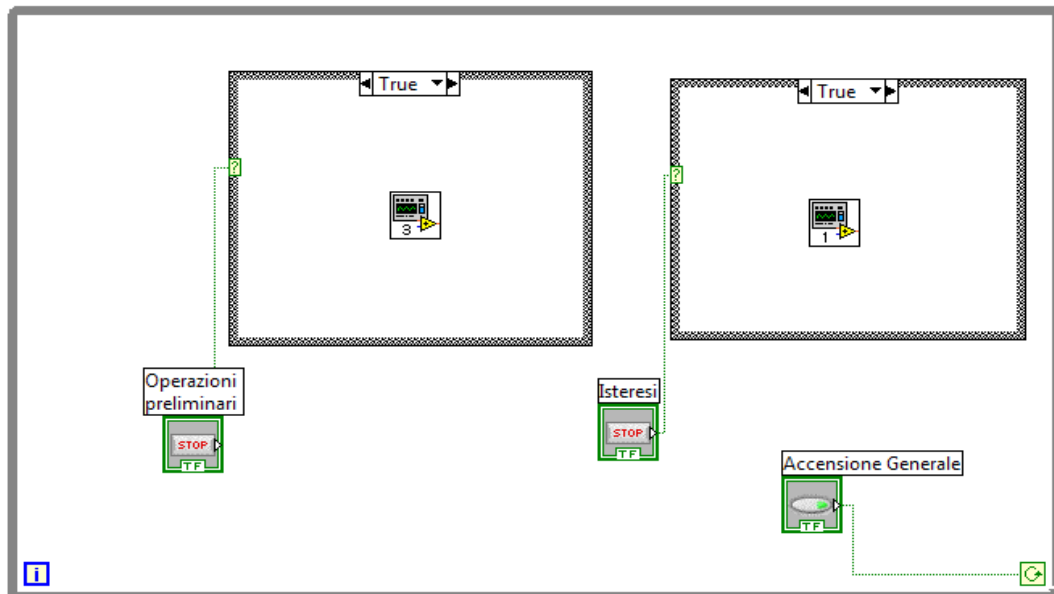


Fig.20 block diagram del menù principale

### 3.3 OPERAZIONI PRELIMINARI

In questa parte del programma l'utente può effettuare tutte quelle operazioni che serviranno per il settaggio dei parametri della rampa di tensione che si vuole generare. Una volta effettuato l'accesso a questa schermata appare, oltre al *front panel*, una finestra per la generazione di un file di tipo .txt: in questo file verranno salvati in formato testuale i punti, del grafico tensione vs. corrente ( $V-I$ ) che si viene a determinare quando viene generata la rampa di tensione.

Come si vede dalla fig.21 i parametri che possono essere impostati dall'utente sono: *cicli*, che indica appunto il numero di spazzolate in tensione che si vogliono far compiere alla rampa, *tensione minima e massima*, che determinano i limiti di tensione superiore ed inferiore della rampa, *resistenza di reazione*, che indica il valore del resistore posto in serie al *memristor* (vedi capitolo 2), e *numero di campioni*, grazie al quale è possibile settare la risoluzione della rampa di tensione.

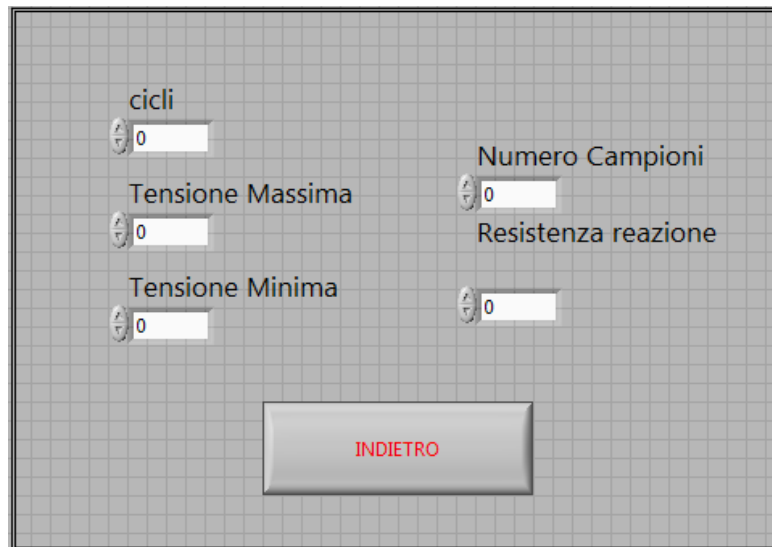


Fig.21 *Front panel* operazioni preliminari

In fig.22 invece è riportato il *block diagram* corrispondente al menù delle operazioni preliminari

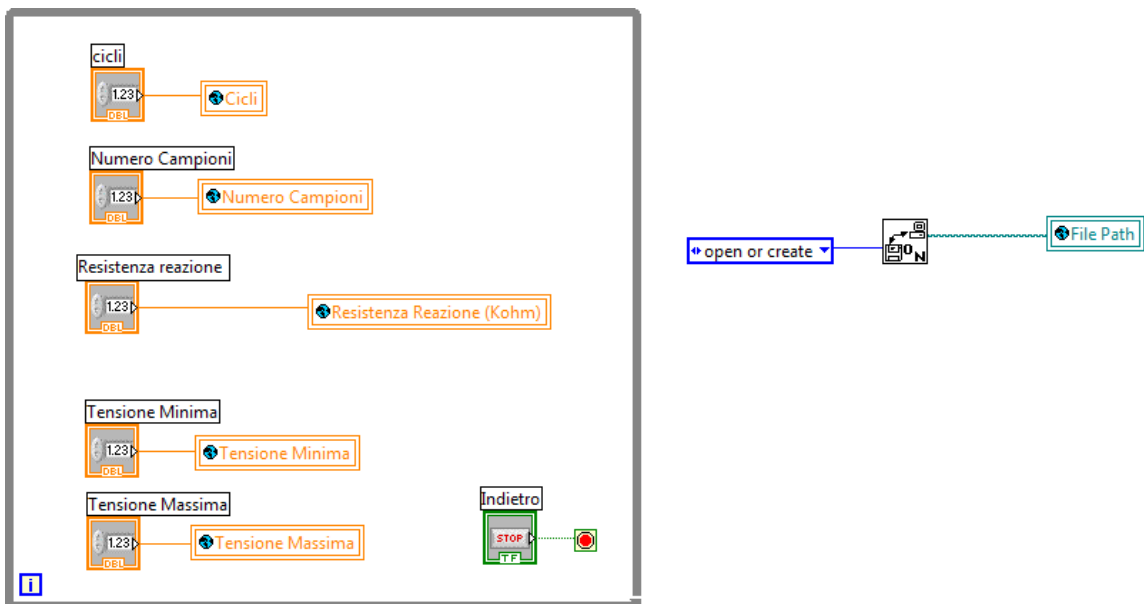




Fig.22 *block diagram* della sezione operazioni preliminari

Dalla figura possiamo notare che sono presenti degli elementi di controllo del tutto speculari a quelli del *front panell* (*numero di campioni, resistenza di reazione* ecc) . Grazie ad essi è infatti possibile trasferire un'informazione dall'utente al programma, che in seguito potrà elaborarla.

All'interno del ciclo while troviamo delle variabili globali che sono quelle con un'icona di questo tipo . Esse vengono utilizzate nel caso in cui il valore al quale vengono inizializzate debba essere usato all'interno di un altro sottoprogramma, nel caso in cui una variabile venga unicamente adoperata in un sottoprogramma si fa uso di una più semplice variabile locale.

Prendiamo come esempio il controllo sul “numero cicli”: se ad esempio viene inserito il valore tre nella relativa icona sul *front panel* il programma reagisce inizializzando la variabile globale “cicli” al valore desiderato (che in questo caso è, appunto, “3”). Tale valore, che resterà memorizzato nella variabile globale fin quando l'operatore non avrà la necessità di cambiarlo, verrà utilizzato in seguito nel sottoprogramma relativo alla generazione della rampa di tensione.

Il ciclo while presente in questo *block diagram* ha il compito di fornire all'utente il tempo necessario per l'inizializzazione dei parametri iniziali, infatti alla fine di tale operazione è possibile, tramite il tasto “indietro”, tornare al menù principale e far partire un ciclo d'isteresi. Le icone a destra della fig.22 sono destinate alla creazione di un file nel quale salvare tutti i risultati della misura: quando, infatti, viene eseguita la schermata “operazioni preliminari” in automatico si apre una finestra dalla quale è possibile selezionare un file di testo esistente o crearne uno nuovo nel quale poi saranno caricati i punti del grafico  $V-I$  che costituiscono il ciclo d'isteresi. L'icona  infine è una variabile globale nella quale viene inserito il percorso del file creato (o selezionato), in questo modo si indica al programma dove dovranno essere inseriti i valori relativi alla misura.



### 3.4 PROGRAMMA PER LA GENERAZIONE DI UNA RAMPA DI TENSIONE

Quando dal menù principale viene selezionato il tasto “Isteresi” il programma esegue la schermata presentata in fig.23.

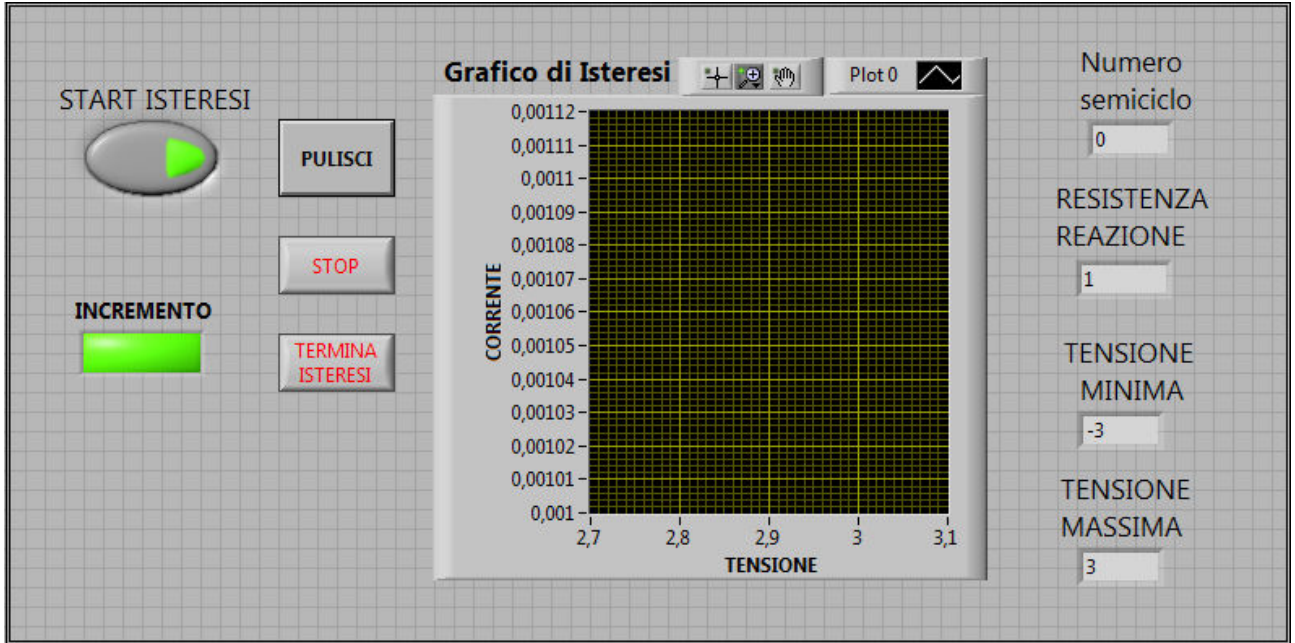


Fig.23 front panel isteresi

Come si vede dalla figura tramite il pulsante “START ISTERESI”, che di default è inizializzato al valore *true*, è possibile far partire il tracciamento del grafico mentre con il tasto “STOP” è possibile interrompere in qualsiasi momento la generazione dello stesso. Con il tasto “TERMINA ISTERESI” invece è possibile una volta terminata la generazione del grafico chiudere definitivamente il programma. Gli indicatori sulla destra della figura visualizzano alcuni dei valori che precedentemente sono stati inseriti nella schermata delle operazioni preliminari e quindi si intuisce che essi sono connessi, nel *block diagram*, con le variabili globali create in precedenza. L'indicatore del “numero semiciclo” in alto a destra è stato inserito per dare contezza ad un osservatore esterno del corretto avanzamento del programma: ad esso infatti è collegato un contatore interno che viene incrementato ogni qual volta il valore della tensione di pilotaggio assume un valore intermedio tra  $V_{max}$  e  $V_{min}$ , ovvero quando è stato tracciato un semiciclo di isteresi. Sulla sinistra possiamo trovare uno *square LED* che risulta essere illuminato quando il valore della tensione della rampa sta aumentando mentre risulta spento se la tensione sta diminuendo (il valore di tensione è direttamente proporzionale al

valore assunto dal potenziometro digitale AD5290). Al centro della schermata è posto il grafico (in cui in ascissa viene plottato il valore di tensione e in ordinata quello della corrente) sul quale verrà tracciato il ciclo d'isteresi. E' importante sottolineare che questo grafico riesce ad "autoscalare" i valori che gli vengono inseriti, nel senso che esso riformatta in continuazione i due assi cartesiani in modo tale da visualizzare sempre le forme d'onda nel modo corretto. Infine con il tasto "PULISCI" è possibile cancellare la traccia presente nel grafico ed inoltre questo comando può essere inserito anche in fase di esecuzione del programma. In fig.24 viene proposto il diagramma a blocchi relativo al *front panel* di fig.23.

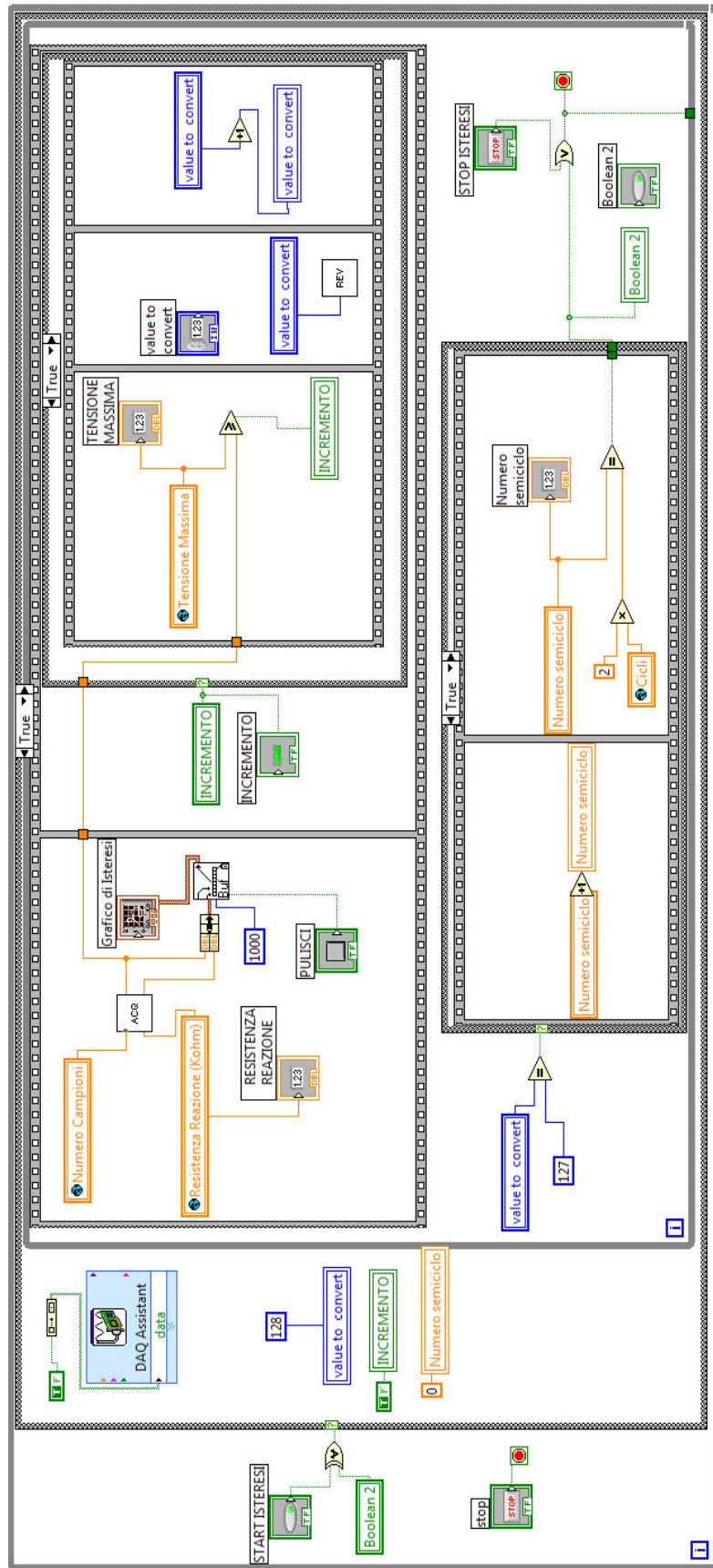



Fig.24 block diagram generatore di rampa

La prima azione da compiere per il corretto funzionamento di un qualsiasi programma, non necessariamente scritto in linguaggio G, è quella di inizializzare ad un valore di default tutte le variabili locali per evitare eventuali valori *undefined*. Proprio per questo, come si può notare dalla figura, all'interno del ciclo *case*, (che viene aperto a seguito della pressione del tasto “START ISTERESI”), sono state inizializzate tutte le variabili globali. Il “Numero semiciclo”, ad esempio, è stato pre-impostato al valore zero mentre il “Value to convert” (che rappresenta uno dei 256 livelli possibili per il potenziometro AD5290) è stato impostato al valore 128 poiché da progetto si è pensato di far partire il potenziometro dal valore corrispondente alla metà della sua resistenza. Infine la variabile “INCREMENTO” è stata inizializzata ad un valore *true* in modo tale che il primo tratto del grafico che si viene a generare sia quello positivo.

Come si vede in figura è stato necessario l'utilizzo di alcune *subVI* grazie alle quali è stato possibile rendere più scorrevole il programma stesso. Una di queste (  ) controlla l'acquisizione dei valori di tensione dal *memristor*; di seguito è rappresentato il suo *block diagram*.

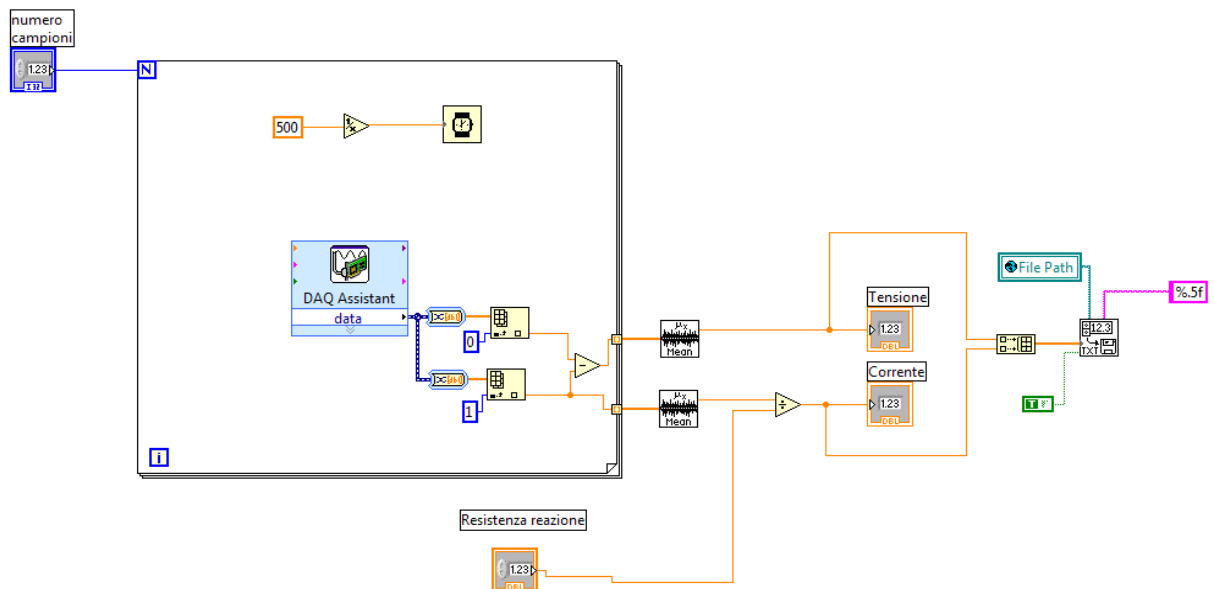
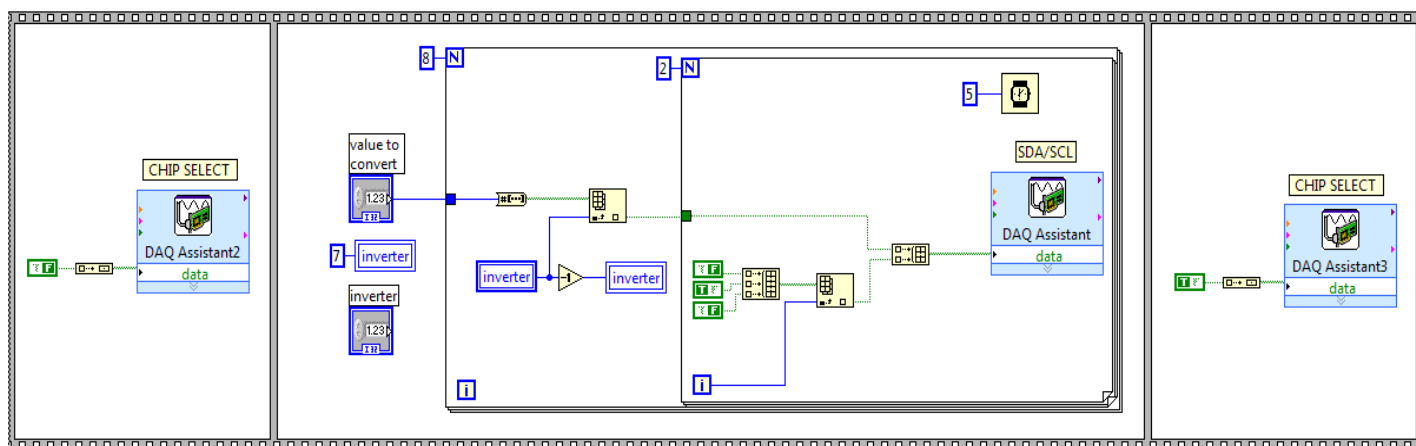


Fig.25 *block diagram* Acquisizione

Nella fig.25 in alto a sinistra troviamo l'elemento di controllo “Numero campioni”, impostato precedentemente nelle operazioni preliminari, grazie al quale è possibile scegliere preventivamente la quantità di campioni che dovrà immagazzinare il sistema prima di emettere un risultato. Supponiamo ad esempio di assegnare al “Numero campioni” (che nella fattispecie determina quante volte si dovrà ripetere il ciclo *for* ) il valore “3”. In questo modo il ciclo *for* verrà ripetuto per tre volte (in un tempo impostato tramite l'icona a forma di orologio) così come l'operazione di acquisizione. Questa

operazione viene effettuata tramite il *DAQ Assistant* le cui uscite sono virtualmente collegate ai capi del *memristor*. Ad ogni ripetizione del ciclo *for* vengono acquisiti due valori di tensione ed in seguito viene fatta la differenza di essi al fine di determinare la d.d.p. ai capi del *memristor*. La corrente viene determinata prendendo il secondo valore di tensione e dividendolo per la resistenza di reazione. Una volta che il ciclo *for* è terminato, sui valori di tensione e corrente acquisiti viene calcolato il valore medio ed essi sono pronti per essere plottati sul grafico *V-I*. Come già illustrato in precedenza, questi vengono anche salvati nel file creato nella schermata delle “Operazioni preliminari”: le icone presenti sulla destra della fig.25 si occupano di tale funzione.

L'altra *subVI* presente nel programma è illustrata in fig.26.



Fig,26 block diagram della *subVI* REV

Il sottoprogramma di figura, denominato REV, è costituito principalmente da una *flat sequence* (struttura più esterna); la cui caratteristica principale è quella di eseguire in modo sequenziale le finestre da sinistra verso destra. In questo caso per prima cosa viene posto il *chip select* dell'AD5290 ad un valore logico basso, poi si passa ad eseguire la finestra centrale nella quale viene mandato al *serial data* il valore che deve assumere il potenziometro: si ricordi che l'AD5290 ha un registro interno chiamato *wiper* nel quale bisogna inserire il valore che si vuole fare assumere alla resistenza interna (vedi capitolo 2); in questo caso il *value to convert* viene trasformato in un valore binario e inviato al *serial data*, cadenzato da un segnale di clock. Infine il *chip select* viene riportato al valore logico alto nella finestra posta all'estrema destra e la *subVI* REV aspetta di essere richiamata nuovamente. Questa struttura è stata implementata seguendo come modello il *datasheet* del potenziometro AD5290 nel quale era riportata la corretta procedura per l'incremento o il decremento del punto di *wiper* fig.10. Dopo che il bitstream è stato inviato al registro di *wiper* tramite il *serial data*, il *value to convert* viene incrementato di una unità e si aspetta la nuova trasmissione di dati.

Come si nota dalla fig.24 la variabile locale “INCREMENTO” viene modificata quando il valore di tensione del *memristor* letto nella *subVI* supera il valore di tensione massima impostato nelle operazioni preliminari ( variabile globale “TENSIONE MASSIMA”). In questo caso la variabile “INCREMENTO” assume il valore *false* e il *value to convert*, dalla successiva acquisizione, comincerà a decrescere denotando un decremento della tensione letta ai capi del *memristor*: quando tale tensione diventa più piccola del valore inserito nella variabile “TENSIONE MINIMA”, “INCREMENTO” passa al valore *true* e la rampa comincia nuovamente a crescere.

Il programma si arresta soltanto in due casi: o quando viene premuto il tasto “STOP” o quando la variabile locale “numero semiciclo” assume un valore pari al doppio di quello inserito dall'utente nella variabile globale “NUMERO CICLI”. Il secondo caso denota un corretto andamento della misura in quanto si verifica soltanto quando si è giunti all'ultimo ciclo d'isteresi.

Infine è importante mostrare come è stata collegata la scheda di acquisizione DAQ 6009 al potenziometro digitale AD5290, ovvero è importante capire quali delle uscite dalla scheda sono state utilizzate come input per il potenziometro. Di seguito è riportata una tabella (tab.1) che indica le uscite, della DAQ, che sono state utilizzate.

<b>USCITE DAQ 6009</b>	<b>INPUT POTENZIOMETRO AD5290</b>
P0.0	CS
P0.1	SDA
P0.2	SCL
GND	GND
AI.0	V0
AI.1	V1

Tab.1 Tabella che mostra i collegamenti tra la DAQ 6009 e il potenziometro digitale AD5290

## CAPITOLO 4 CARATTERIZZAZIONE DEI MEMRISTOR CON STRUTTURA A PAD

### 4.1 LAYOUT UTILIZZATO PER LA REALIZZAZIONE DEI MEMRISTOR

In fig.30 è mostrato il layout che è stato prodotto in questo lavoro di tesi con lo scopo di realizzare dei dispositivi memristivi.

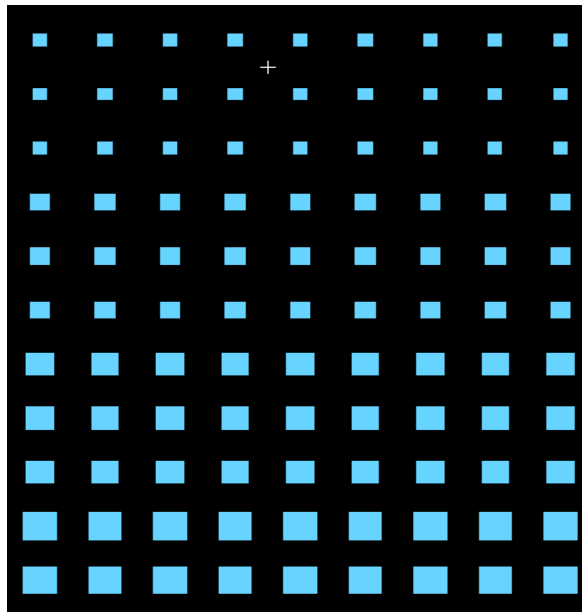


Fig.30 Layout per la struttura a pad (in figura la pad più grande è di dimensioni 150 $\mu$ m X 150 $\mu$ m)

Come si nota dalla fig.30 è stata realizzata una matrice 11x9 di pad con dimensioni variabili, infatti: nelle prime tre righe, partendo dall'alto, troviamo le pad con dimensione di 100 $\mu$ m, nelle tre che seguono in basso vi sono le pad da 150 $\mu$ m, le altre tre file sono relative alle pad con dimensione di 200 $\mu$ m mentre nelle ultime due righe vi sono quelle con dimensione di 250 $\mu$ m e 300 $\mu$ m.

Il file creato con il programma ELECTRIC infine è stato esportato in formato .cif poiché per realizzare la fotolitografia è stato utilizzato un laser che come input ha bisogno di un file con questa estensione.

Dopo aver effettuato la scrittura diretta con il laser e la deposizione dei contatti metallici di Al [10], si è ottenuta la struttura a pad illustrata in fig.31. Si è quindi proceduto alla caratterizzazione elettrica dei dispositivi realizzati.

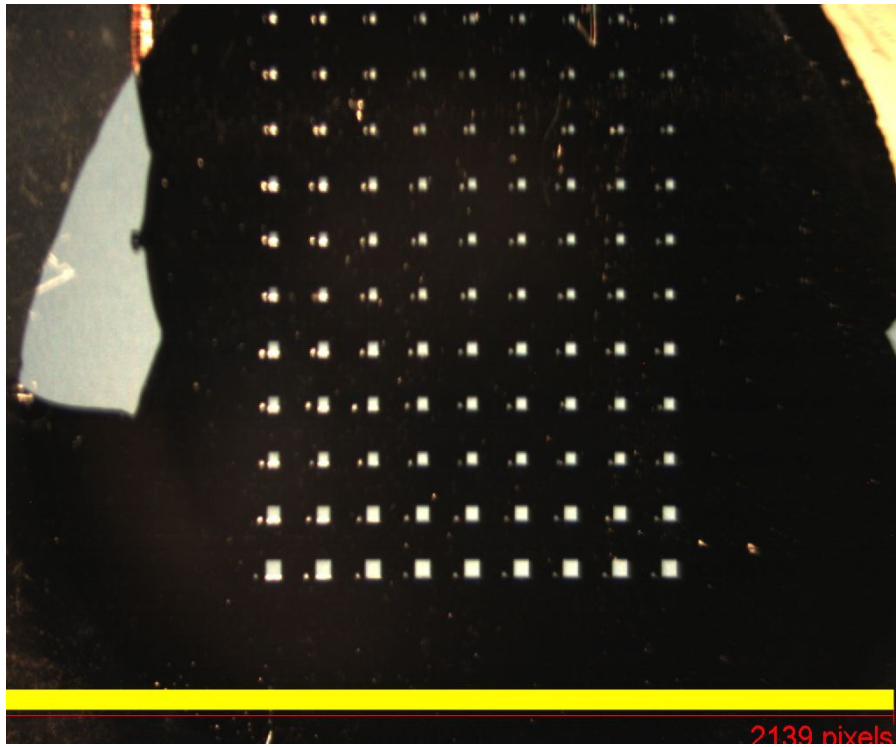


Fig.31 Struttura a pad realizzata mediante scrittura diretta con laser (campione di memristor con ZNO)



## 4.2 ESEMPIO DI CARATTERIZZAZIONE DEI MEMRISTOR

Per la caratterizzazione del *memristor* è stato utilizzato il setup di misura illustrato nel capitolo 2 in fig.15 e il programma in LABVIEW mostrato nel capitolo 3.

Innanzitutto è stato posizionato il campione di *memristor* nella *probe station* e, con l'ausilio di uno stereomicroscopio, è stato possibile posizionare con precisione le punte di *probe* sia sulla pad selezionata, sia sul *bottom contact*, che come detto rappresenta un contatto comune a tutti i dispositivi realizzati (cfr. in fig.32).

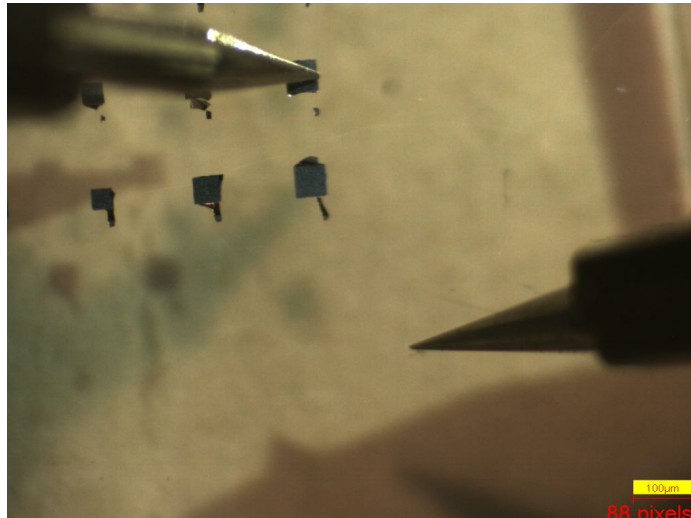


Fig.32 Visione allo stereomicroscopio del *memristor* e delle punte della *probe station*

A questo punto è stato collegato il *memristor data acquisition system (MDAS)* alla *probe station* attraverso dei connettori BNC e in seguito è stata collegata la NI-DAQ 6009 all'MDAS tramite l'apposito cavetto. La scheda di acquisizione è stata collegata al PC tramite un connettore USB e in seguito tramite il programma descritto nel capitolo 3 sono stati impostati i parametri relativi alla generazione della rampa di tensione in questo modo :

- Numero campioni → 50
- Tensione minima → - 4 V, tensione massima → 4 V
- Numero cicli → 10
- Resistenza di reazione → 1  $\Omega$

Dal menù principale è possibile a questo punto far partire un ciclo di isteresi, infatti premendo il

tasto “START ISTERESI” come già visto verrà visualizzata sul desktop la schermata di fig.33.

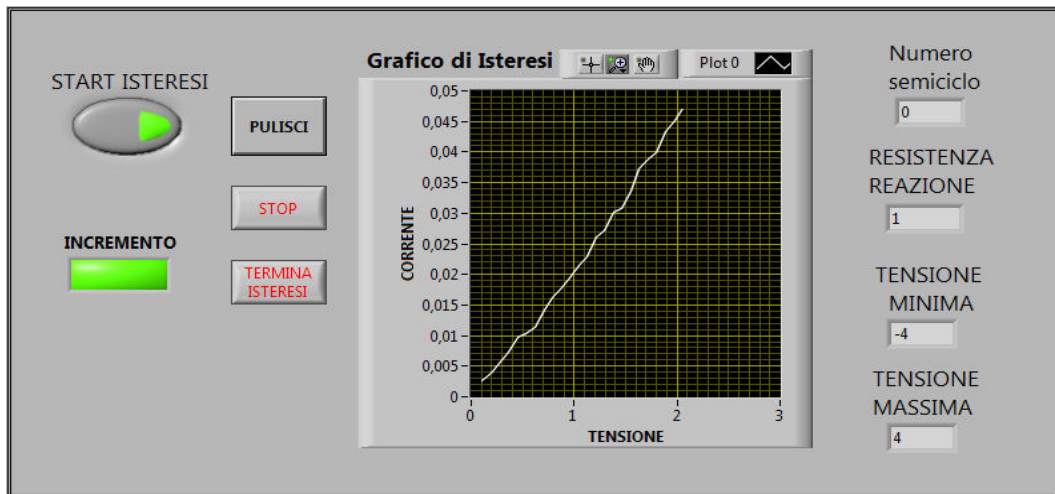


Fig.33 *Front panel* del grafico di isteresi

La fig.34 invece mostra come, dopo aver fatto partire il programma, si viene a formare il ciclo di isteresi.

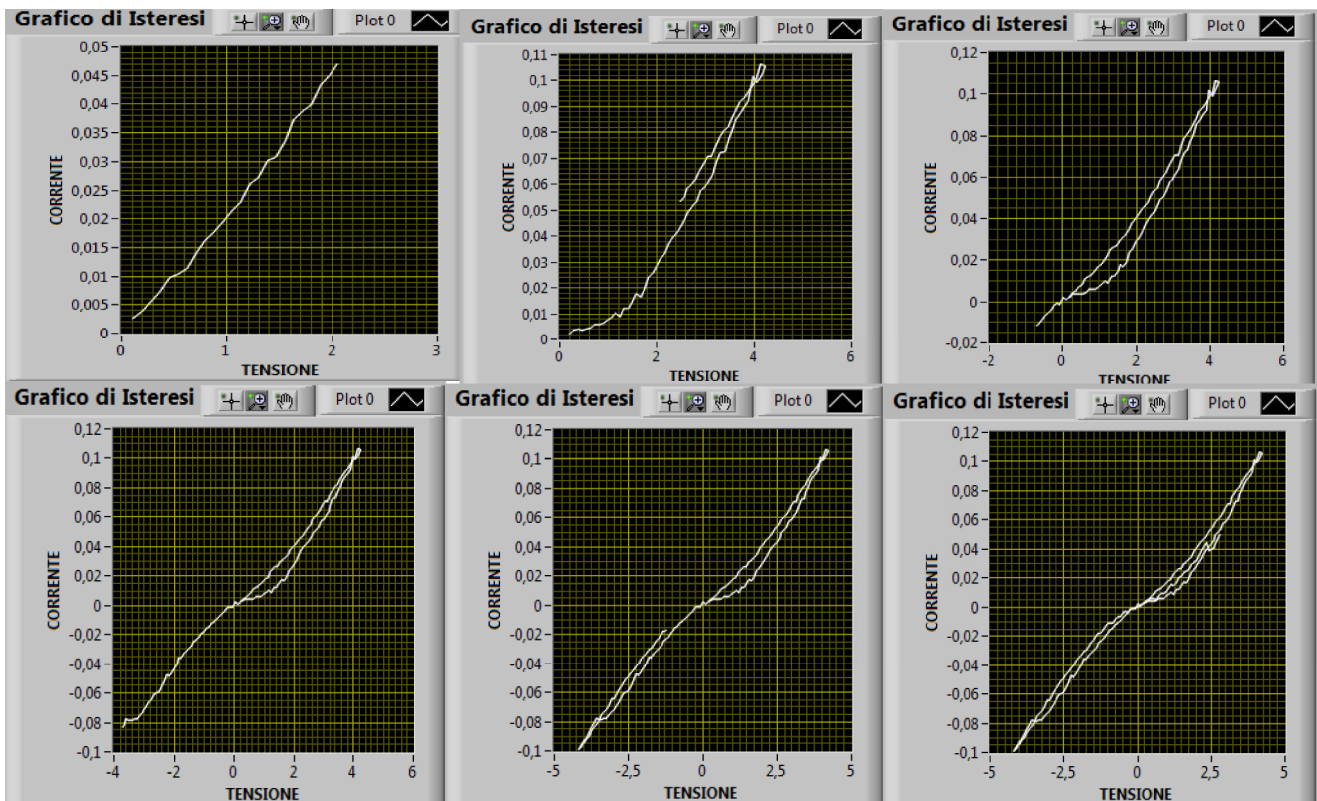


Fig.34 Sequenza di screenshot del grafico d'isteresi

Come illustrato nel capitolo 3, in concomitanza con la generazione del grafico d'isteresi, viene creato un file di testo nel quale sono contenuti tutti i punti del grafico stesso (fig.35).

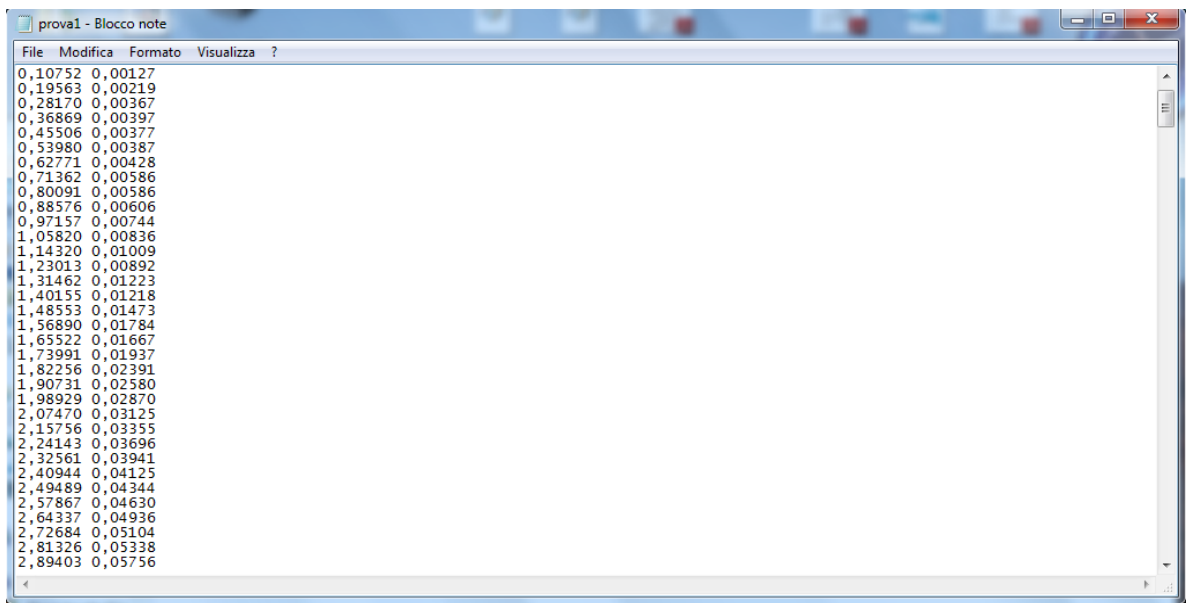


Fig. 35 File di testo contenente i punti del grafico di isteresi

La colonna di sinistra riporta i punti di ascissa (tensione espressa in V) mentre la colonna di destra riporta i dati di ordinata (corrente espressa in A). A questo punto i valori presenti nel file di testo sono stati plottati su un grafico grazie all'ausilio del programma ORIGIN; il risultato finale è illustrato in fig.36.

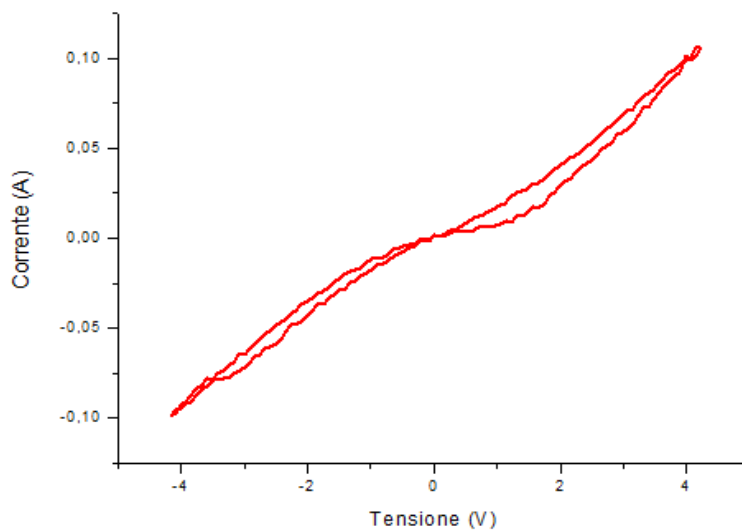


Fig.36 Grafico d'isteresi del campione di memristor realizzato con VO<sub>2</sub> plottato tramite ORIGIN

### 4.3 CARATTERIZZAZIONE DI MEMRISTOR REALIZZATI CON VO<sub>2</sub> E ZnO

Le prime misure sono state effettuate con i campioni realizzati con ossido di zinco (ZnO) su substrato di FTO e nei quali l'*upper contact* era stato realizzato con alluminio. Nello specifico in fig.37 è rappresentato il grafico d'isteresi che si forma effettuando la misura su una pad da 100µm.

La misura inoltre è stata effettuata impostando i parametri della rampa come segue:

- Numero campioni → 20
- Tensione minima → -3 V, tensione massima → 3 V
- Numero cicli → 10
- Resistenza di reazione → 1 Ω

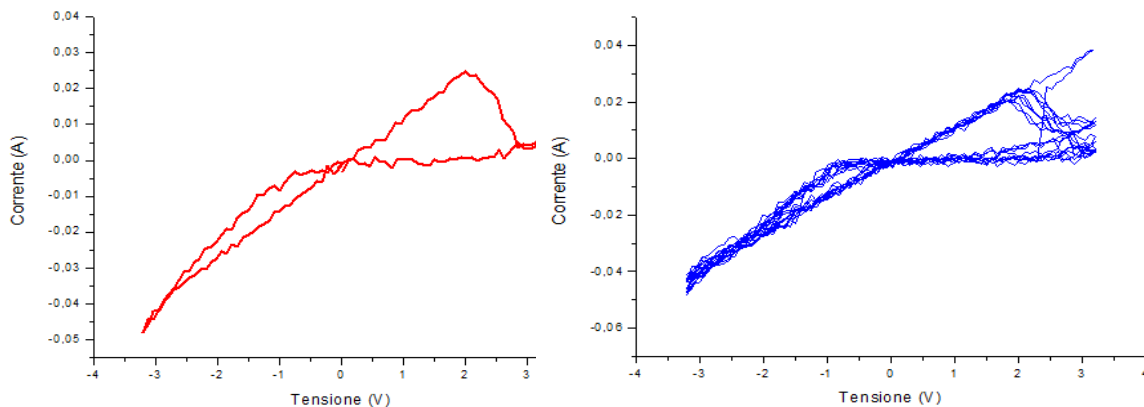


Fig.37 Sinistra: ciclo di forming. Destra: ulteriori cicli. Pad da 300µm di TiO<sub>2</sub>

Come si può notare dalla fig.37 la risposta del *memristor* ad una rampa di tensione è, come ci si aspettava, un ciclo d'isteresi compreso tra i valori di tensione di -3 V e 3 V. A sinistra, in rosso, è riportato il primo ciclo, chiamato di *forming* per denotare il fatto che esso è appunto quello che determina il comportamento memristorico del componente. Da un punto di vista teorico, i dieci cicli successivi dovrebbero differire da quello di *forming*: le dieci curve d'isteresi dovrebbero, infatti, percorrere punti differenti da quelli percorsi dalla curva del primo ciclo. Questa caratteristica nei campioni di ossido di zinco non è molto evidente, anche se confrontando i due grafici di fig.37 effettivamente si nota che questi hanno andamenti leggermente differenti. Cosa più importante è però il modo in cui si sovrappongono i 10 cicli d'isteresi. Essi infatti come si nota dalla figura si allineano quasi perfettamente l'uno sull'altro determinando in questo modo una certa ripetitività della misura.

Questo aspetto è fondamentale in quanto, dovendo essere utilizzati come memorie, essi dovrebbero mostrare un comportamento praticamente uguale anche dopo migliaia di cicli di lettura o scrittura.

Dalla figura 37 si nota che le correnti che scorrono sul *memristor* quando viene applicata una tensione di 2 V sono di circa 25 mA: queste in realtà risultano essere un po' più alte rispetto a quelle generalmente riportate in letteratura che si aggirano intorno ai 10 mA [9]. Per quanto riguarda i campioni di ZnO considerando pad di dimensioni più grande si nota che l'effetto memristorico comincia ad essere meno visibile come mostrato in fig.38.

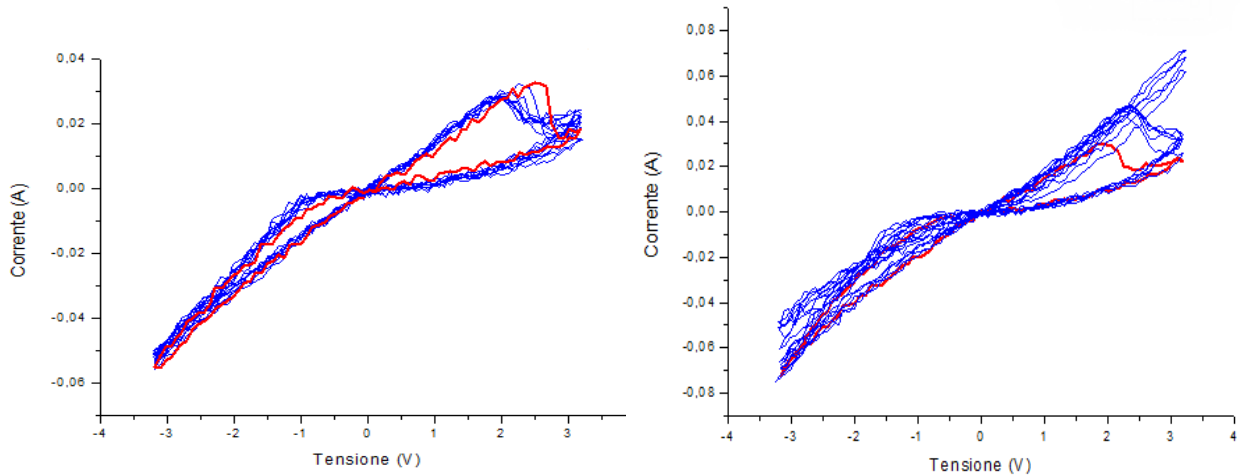


Fig.38 A sinistra ciclo con pad da 150μm, a destra ciclo con pad da 200μm

Sicuramente degli effetti altrettanto visibili sono stati riscontrati nei *memristor* realizzati con ossido di vanadio ( $\text{VO}_2$ ) su un substrato di FTO e nei quali l'*upper contact* è stato realizzato in alluminio. In fig.39 è riportato il grafico relativo alla misurazione, effettuata sempre con i parametri precedenti, su una pad da 250μm.

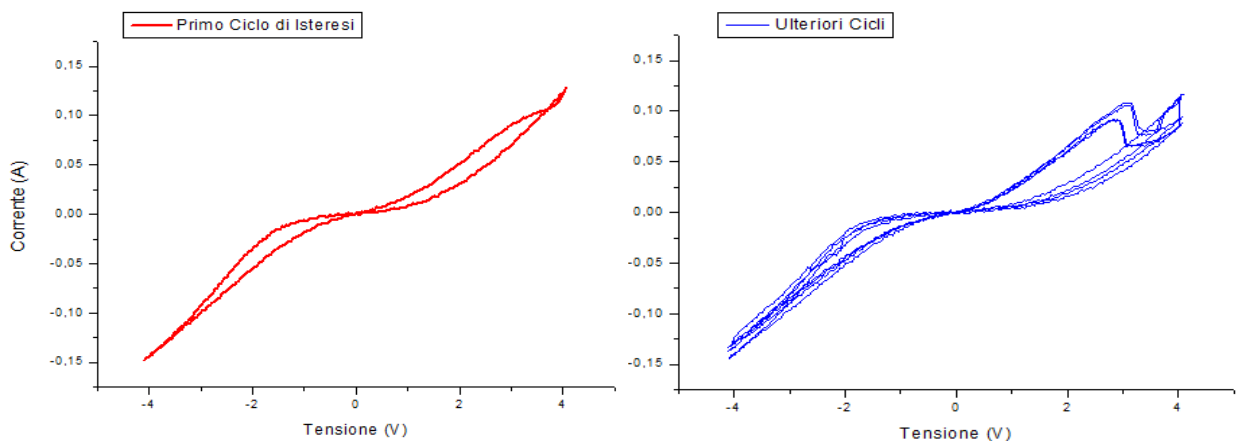


Fig.39 Ciclo di isteresi con pad da 250μm di VO2

Come si nota dalla figura in questo caso il ciclo di *forming*, rappresentato in rosso, è molto diverso rispetto ai cicli successivi. Questo è un fattore fondamentale in quanto come abbiamo già detto in un ciclo di isteresi generico è cosa consueta che il primo ciclo differisca sostanzialmente dagli altri. Inoltre anche in questo caso sono state effettuate 10 spazzolate in tensione tra  $-4\text{ V}$  e  $4\text{ V}$  e il risultato è che tutti i cicli ottenuti sono sostanzialmente sovrapponibili (denotando quindi una buona ripetibilità della misura). Anche in questo caso le correnti in gioco sono un po' più alte rispetto a quelle viste in letteratura, infatti nei grafici di fig.39 quando si arriva alla tensione massima la corrente ha un picco di circa  $120\text{ mA}$ . Per l'ossido di vanadio però i grafici relativi a pad con dimensioni più grandi non presentano gli stessi problemi riscontrati con le pad relative all'ossido di zinco. In questo caso infatti, anche quando effettuiamo delle misurazioni su campioni di dimensioni superiori comunque, l'effetto memristorico è abbastanza visibile e il ciclo di *forming* è notevolmente diverso dagli altri.

In fig.40 sono riportati dei grafici effettuati su pad di dimensioni inferiori.

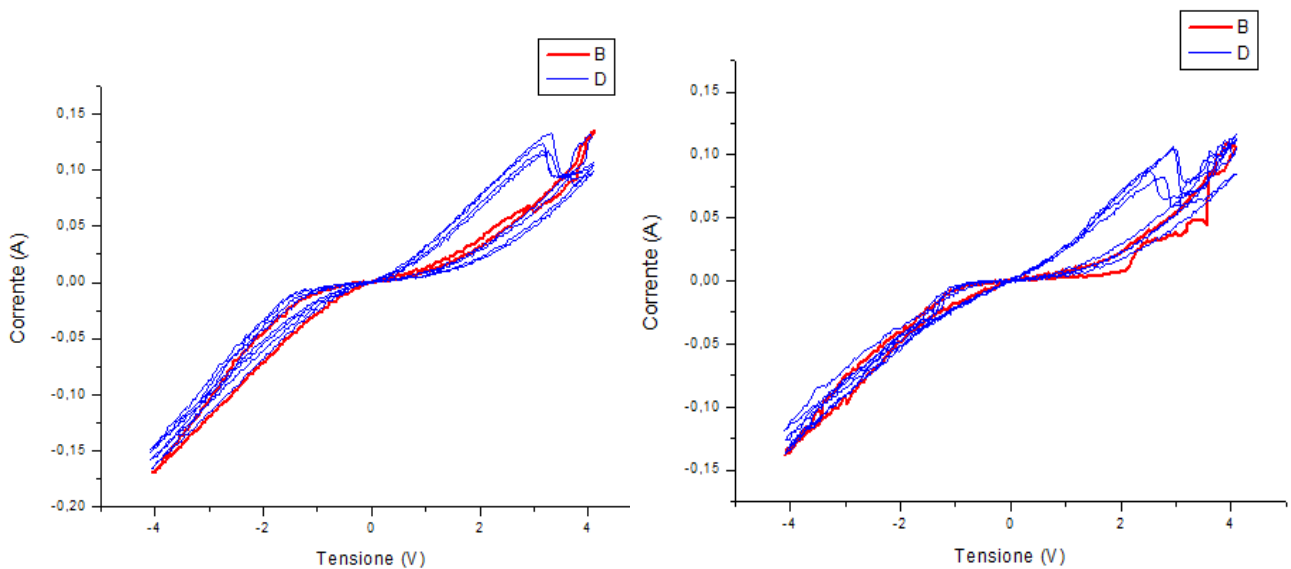


Fig.40 Sinistra: ciclo con pad da  $200\mu\text{m}$ . Destra: ciclo con pad da  $150\mu\text{m}$ . Pad realizzate con  $\text{VO}_2$

Maggiori dettagli sulle misure effettuate possono essere trovati in [10].

## CAPITOLO 5 PROGETTAZIONE DEL LAYOUT PER LA REALIZZAZIONE DI MEMRISTOR CON GEOMETRIE PIU' AVANZATE

Le strutture con le quali è possibile realizzare un *memristor* sono molteplici e proprio per questo motivo si è pensato di realizzare dei file con layout differenti sia in termini di geometrie, sia in termini di dimensioni.

Una delle strutture più utilizzate è quella *crossbar* [4] mostrata in fig.27

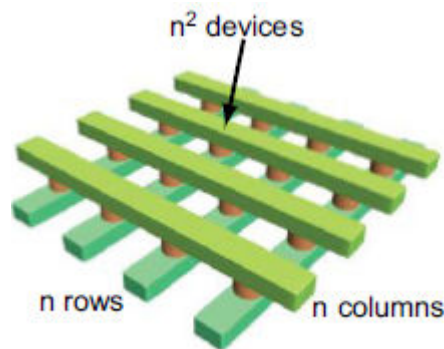


Fig.27 Struttura crossbar [4]

In questo caso il *memristor* si viene a trovare nei punti di incrocio tra le righe e le colonne della struttura, quindi se si realizzano  $n$  righe ed  $n$  colonne si avranno  $n^2$  componenti da poter utilizzare. Questa struttura consente inoltre di realizzare dispositivi con una superficie attiva anche di dimensioni submicrometriche, utilizzando delle pad connesse con ciascuna riga/colonna per effettuare la caratterizzazione elettrica attraverso il set-up descritto nei capitoli precedenti.

Un'altra struttura largamente utilizzata [5] è quella mostrata in fig.28

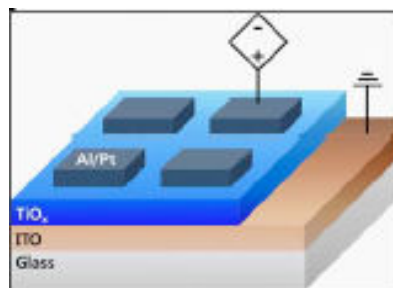


Fig.28 Struttura a pad [5] nel quale è stato realizzato il *bottom contact* in ITO e l'*upper contact* in Al/Pt

In questa struttura denominata a pad il *memristor* si viene a creare nella zona sottostante il contatto metallico che in questo caso è Al/Pt. Infatti con questa geometria uno dei due contatti (*bottom contact*) è il substrato che nella figura è stato realizzato di ITO mentre l'altro contatto (*upper contact*) è quello che si trova sopra lo strato di ossido.

La prima struttura è di più difficile realizzazione in quanto richiede due step di litografia, ma, come già detto permette di raggiungere dimensioni molto ridotte mentre la seconda è più semplice da realizzare ma con essa non si possono realizzare componenti di dimensioni submicrometriche per ovvie ragioni di posizionamento delle punte della probe-station.

I vari layout sono stati realizzati utilizzando un programma per il cad di dispositivi VLSI (ELECTRIC), e il risultato del lavoro è illustrato in fig.29

Sulla destra si trova il layout relativo alla struttura *crossbar* e nella fattispecie sono state realizzate soltanto le colonne poiché utilizzando lo stesso layout in un secondo momento è possibile realizzare anche le righe. Il numero accanto ad ogni struttura indica la dimensione della colonna (e quindi la dimensione che poi assumerà il *memristor*) Si è deciso infatti di realizzare le seguenti dimensioni:

- 0 → 1μm
- 1 → 2μm
- 2 → 3μm
- 3 → 5μm
- 4 → 10μm
- 5 → 15μm
- 6 → 20μm
- 7 → 30μm
- 8 → 40μm
- 9 → 50μm
- 10 → 60μm
- 11 → 80μm
- 12 → 90μm
- 13 → 100μm

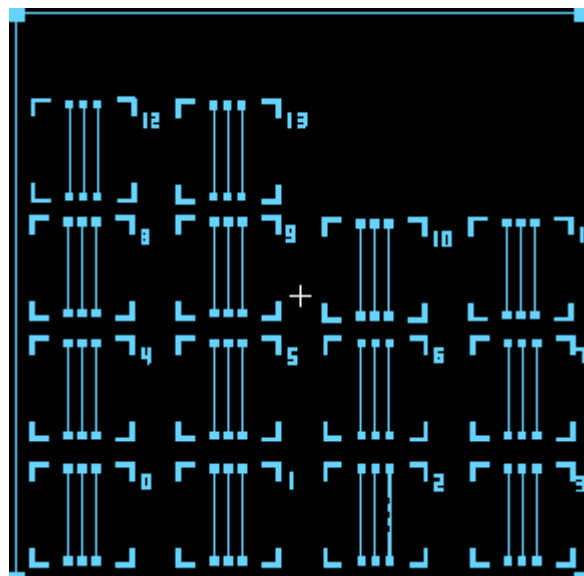


Fig.29 layout per la realizzazione di un *memristor*

Per quanto riguarda le pad sulle quali verranno poste le punte della probe station, le dimensioni scelte sono di 150μm X 150μm per tutte le strutture realizzate.



## CONCLUSIONI

In ultima analisi si può affermare che il sistema di caratterizzazione progettato e realizzato in questo lavoro di tesi ha portato a risultati certamente comparabili con quelli trovati in letteratura. Infatti il ciclo di isteresi che è stato ottenuto sperimentalmente, su diverse tipologie di campioni, è del tutto simile a quello scoperto dal team della HP [3]. Questo porta alla conclusione che effettivamente l'effetto memristorico che ci si aspettava è stato effettivamente riscontrato sperimentalmente.

I risultati ottenuti aprono la strada ad ulteriori studi che possono essere affrontati quali ad esempio: fare uno studio in frequenza del dispositivo, ovvero discriminare in modo univoco la velocità ottimale con la quale è possibile ottenere una memorizzazione certa del dato. Essendo, il *memristor*, principalmente utilizzato come elemento primario di una memoria è interessante verificare la sua vita media, ovvero fare una stima sul numero massimo di cicli di scrittura che è possibile effettuare prima che il dispositivo smetta di funzionare.

Certamente si possono verificare le ipotesi descritte nell'articolo [5], secondo le quali sarebbe possibile incrementare l'isteresi del grafico  $V-I$  pilotando il memristor con un'onda quadra, di tensione, a frequenza elevata; per far ciò basterebbe apportare alcune modifiche sia all'*hardware* che al *software* descritti in questo lavoro di tesi, infatti a partire dal *memristor data acquisition system* e grazie alla scheda di acquisizione DAQ 6009 è possibile generare un'onda quadra di frequenza variabile che permetterebbe una più completa caratterizzazione del *memristor*.

In ultima analisi uno studio molto interessante è quello che evidenzia le differenze che si riscontrano per quanto riguarda l'effetto memristivo al variare della geometria del *memristor* e per far ciò si potrebbe partire ad esempio dalla struttura descritta nel capitolo 5.

## BIBLIOGRAFIA

- [1] Leon O Chua, Memristor-The Missing Circuit Element, IEEE transaction on circuit theory, **5**, 507-519, 1971.
- [2] Jessica Dennis, The Memristor; introduction to nanoelectronics, Rowan university, 3-14, 2009
- [3] Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart & R. Stanley Williams, The missing memristor found, Nature, **453**, 80-83, 2008.
- [4] Wei lu, Kuk-Hwan Kim, Ting Chang, and Siddharth Gaba, Two-Terminal Resistive Switches (Memristors) for Memory and Logic Applications, IEEE, 217-223, 2011
- [5] Seungjae Jung, Jaemin Kong, Sunghoon Song, Kwanghee Lee, Takhee Lee, Hyunsang Hwang, and Sanghun Jeon, Resistive Switching Characteristics of Solution-Processed Transparent TiO<sub>x</sub> for Nonvolatile Memory Application, Journal of The Electrochemical Society, **157**, 1042-1045, 2010
- [6] K. Michelakis, T. Prodromakis, C. Toumazou, Cost-effective fabrication of nanoscale electrode memristors with reproducible electrical response, Micro & Nano Letters, **5**, Iss. 2, 91–94, 2010
- [7] K. Michelakis, T. Prodromakis, C. Toumazou, Switching mechanisms in microscale memristors, Electronics Letters, **46**, No. 1, 1-2, 2010
- [8] J. Joshua Yang, Matthew D. Pickett, Xuema Li, Douglas A., A. Ohlberg, Duncan R. Stewart, R. Stanley Williams, Memristive switching mechanism for metal/oxide/metal nanodevices, Nature Nanotechnology, **3**, 429-433, 2008
- [9] J. Joshua Yang, X. Zhang, John Paul Strachan, Feng Miao, Matthew D. Pickett, Ronald D. Kelley, Medeiros-Ribeiro, R. Stanley Williams, High switching endurance in TaO<sub>x</sub> memristive devices, Applied Physics Letters, **97**, 1-3, 2010
- [10] Tesi di laurea di Vincenzo Costanza, *Realizzazione e progettazione di memristor*, Università degli Studi di Palermo-Facoltà di Ingegneria, Corso di Laurea in Ingegneria Elettronica, A.A 2012/2013

## **RINGRAZIAMENTI**

*Devo i miei più vivi ringraziamenti innanzitutto all' Ing. Mauro Mosca e all' Ing. Roberto Macaluso senza i quali non avrei potuto affrontare il percorso che ha portato allo svolgimento di questo lavoro di tesi; grazie al loro aiuto infatti ho avuto la possibilità di mettere in pratica quanto in questi anni ho appreso. Inoltre mi preme ringraziare il professore Claudio Cali che mi ha dato la possibilità di usufruire delle risorse del laboratorio di film sottili.*

*Secondi, ma non per importanza, sono i ringraziamenti che rivolgo alla mia famiglia che mi ha supportato in questo cammino incoraggiandomi e affrontando con me giorno per giorno le difficoltà incontrate. Un altro vivo ringraziamento va a Roberta, la mia ragazza, che con la sua presenza mi ha dato la carica necessaria per affrontare nel migliore dei modi questo percorso di studio.*

*Un pensiero speciale va ai colleghi del laboratorio, Biagio Seminara e Fulvio Caruso, per la loro disponibilità nell'offrirmi l'aiuto necessario per la realizzazione di questo lavoro di tesi.*

*Infine un grazie di cuore va al mio compagno di avventure Vincenzo Costanza con il quale ho trascorso e condiviso momenti di gioia ma anche momenti più duri; con la sua simpatia infatti è stato in grado di rendere più piacevole questo percorso di studio alla fine del quale non ritrovo più un semplice collega ma un caro amico.*