

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II



FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN

INGEGNERIA DELLE TELECOMUNICAZIONI

(CLASSE DELLE LAUREE IN INGEGNERIA DELL'INFORMAZIONE N.9)

DIPARTIMENTO DI INGEGNERIA ELETTRICA

ELABORATO DI LAUREA

**REALIZZAZIONE DEL SOFTWARE PER IL
CONTROLLO DI ESPERIMENTI DIDATTICI
SUI CIRCUITI**

RELATORE

CH.MO PROF. ING.

MASSIMILIANO DE MAGISTRIS

CORRELATORE

ING. MASSIMO ATTANASIO

CANDIDATO

DARIO IMPROTA

MATR. 841/03

ANNO ACCADEMICO 2007/2008

*Ai miei cari genitori
e a mia nonna*

Ringraziamenti

Desidero ringraziare il Professor Massimiliano de Magistris, l'Ing. Massimo Attanasio e l'Ing. Marco Colandrea per i consigli e le informazioni necessarie per la buona riuscita di questo elaborato.

Infine ringrazio tutti coloro che mi sono stati vicino.

INDICE

INTRODUZIONE.....	7
Capitolo 1 - LA DIDATTICA SPERIMENTALE SUI CIRCUITI.....	9
1.1 La didattica sperimentale in laboratorio, in aula e da remoto	9
1.2 Dispositivi per la didattica sui circuiti	11
1.2.1 Interfaccia USB.....	11
1.2.2 Interfaccia GPIB e USB-GPIB	13
1.2.3 Schede d'acquisizione d'input/output analogiche/digitali	14
1.2.4 Generatori di segnali	16
1.2.5 Oscilloscopi	18
1.3 Alcuni circuiti per la didattica sperimentale	20
1.3.1 Circuiti lineari, non lineari e caotici.....	21
1.3.2 Circuiti del primo ordine	23
1.3.3 Circuito lineare del primo ordine con forzamento sinusoidale.....	24
Capitolo 2 - HARDWARE PER LA DIDATTICA SPERIMENTALE SUI CIRCUITI	26
2.1 La scheda "prototipo"	28
2.1.1 Il protocollo di comunicazione SPI.....	29
2.1.2 Il generatore di segnali	31
2.1.3 Circuito di switch, circuito di acquisizione e multiplexer.....	33
2.2 Descrizione degli integrati per la scheda prototipo	34
2.2.1 Il modulo UM245R.....	34
2.2.2 Il DAC TLV5616CP	36
2.2.3 L'ADC TLC2543IN.....	37

2.2.4	Il Port Expander MCP23S17	37	
2.2.5	Il Transistor Array ULN2003AN.....	39	
2.2.6	Il relè MT2-C93432	39	
2.2.7	Il Direct Digital Synthesis AD9834	40	
2.2.8	Il DAC MCP4922	42	
2.2.9	L'operazionale sottrattore TL082CP.....	43	
2.2.10	Il moltiplicatore AD835AN.....	44	
2.2.11	L'amplificatore di tensione LM6172IN	45	
2.2.12	L'amplificatore di corrente THS6012I.....	45	
2.3	La scheda d'acquisizione veloce PCI-5102	47	
2.4	Gli esperimenti	48	
2.4.1	Circuito caotico RLD	48	
2.4.2	Circuito caotico di Chua.....	49	
Capitolo 3 - REALIZZAZIONE DEL SOFTWARE DI CONTROLLO, PER IL			
LABORATORIO DI CIRCUITI			54
3.1	Introduzione al Labview	54	
3.1.1	Cos'è il Labview?	54	
3.1.2	Utilizzo principale di LabView.....	56	
3.1.3	Virtual Instrument (VI)	57	
3.1.4	Sub - VI	58	
3.2	Descrizione del software di controllo per il prototipo	59	
3.2.1	Software di controllo: le diverse versioni.....	59	
3.2.2	Versione del software per l'utilizzo in aula o in laboratorio	60	

3.2.3	Versione del software per l'utilizzo da remoto.....	75
3.3	Virtual Instrument per il testing della scheda prototipo.....	80
3.3.1	Sub - VI: "TestUM245R.vi"	80
3.3.2	Sub - VI: "Test MCP23S17.vi"	80
3.4	Sub - VIs di libreria per il software di controllo	81
3.4.1	Files di libreria del modulo UM245R	82
3.4.2	Files di libreria del Port Expander MCP23S17	84
3.4.3	File di libreria del DAC TLV5616.....	86
3.4.4	Files di libreria del Generatore di segnali	87
3.4.5	Altri sub-VI per il funzionamento del software.....	95
Capitolo 4 - REMOTIZZAZIONE WEB DEGLI ESPERIMENTI.....		97
4.1	Cenni sul linguaggio HTML	98
4.2	Il sito web per il laboratorio remoto di circuiti	99
4.2.1	Struttura del sito	99
4.2.2	Realizzazione delle pagine in HTML	100
4.2.3	I tag HTML utilizzati per la realizzazione del sito	101
4.3	Remotizzazione del software di controllo	102
4.3.1	Procedura per la remotizzazione	102
CONCLUSIONI.....		107
BIBLIOGRAFIA.....		108

INTRODUZIONE

Nello studio dei circuiti elettrici ed elettronici l'attività sperimentale è di notevole importanza per comprendere molti argomenti. Difatti la possibilità di affiancare alle ore di studio teorico interessanti esperimenti in laboratorio, che mostrano in pratica ciò che si è sviluppato in teoria, ne permettono una migliore comprensione.

E' questo il motivo per cui negli anni, presso il laboratorio di circuiti e diagnostica elettrica e magnetica, sono stati realizzati da laureandi ed ingegneri, un insieme di sistemi hardware e software per la didattica sperimentale sui circuiti.

Tali apparati di piccole dimensioni e interfacciati direttamente col calcolatore, hanno gradualmente sostituito la strumentazione convenzionale, e consentono di eseguire esperienze, sia in aula che da remoto, su circuiti fisicamente realizzati. Il laboratorio remoto di circuiti, accessibile da utenti remoti via internet tramite web browser, costituisce un laboratorio didattico virtuale finalizzato ad esercitazioni sperimentali e all'apprendimento interattivo, mediante l'utilizzo di un modello client-server e di un software personalizzato basato sul linguaggio LabView. Il laboratorio remoto di circuiti è costituito da un server, un'interfaccia web, una scheda d'acquisizione commerciale, un circuito d'acquisizione, un circuito di switch che controlla un multiplexer a relè, e degli esperimenti, che attualmente sono un circuito caotico di Chua ed un circuito a risonanza non lineare (RLD). L'utente che interagisce con questi circuiti, ha la possibilità di variarne alcuni parametri osservandone gli effetti su interfacce grafiche realizzate in ambiente Labview, che ricordano molto quelle degli oscilloscopi tradizionali.

L'insieme dei circuiti sperimentali, di controllo e di gestione sviluppati è stato recentemente integrato in un'unica scheda realizzando un prototipo.

In questo contesto si colloca il presente elaborato, cui obiettivo è stato quello di realizzare il software, in ambiente Labview 7.1, per il controllo di questo prototipo, in modo da rendere fruibili gli esperimenti sia in aula, sia in laboratorio e sia in remoto.

A tal fine sono stati messi a punto i sub-VI per il corretto funzionamento dei circuiti, ed è stata realizzata un'interfaccia grafica, che permette all'utente di scegliere quale tra gli esperimenti proposti desidera eseguire.

Sono state implementate due versioni del software descritte entrambe nel terzo capitolo: una versione è indicata per l'utilizzo in aula o in laboratorio, ed implementa tutte le funzioni disponibili, ovvero: controllo del circuito di Chua, del circuito RLD, del generatore di segnali e scrittura nel DAC del circuito di acquisizione.

L'altra, più specializzata, è stata realizzata per l'utilizzo in remoto, ed offre la possibilità di selezionare solo gli esperimenti d'interesse didattico, ovvero il circuito di Chua e quello RLD; tale versione necessita inoltre di alcune funzioni dedicate, per preservare il corretto funzionamento del software anche nel caso di un utilizzo "non previsto".

Durante le fasi realizzative del prototipo, sono stati inoltre implementati alcuni programmi specifici per eseguirne il testing passo passo.

Nel quarto capitolo viene descritta la procedura di remotizzazione web del software attraverso il tool di pubblicazione incluso in Labview, ed il sito web realizzato per consentire l'accesso via internet agli esperimenti.

Capitolo 1

LA DIDATTICA SPERIMENTALE SUI CIRCUITI

Nello studio dei circuiti elettrici/elettronici l'impiego di attività sperimentali è importante per comprendere molti argomenti. Difatti la possibilità di poter affiancare alle ore di studio teorico interessanti esperimenti in laboratorio, che mostrano in pratica ciò che si è sviluppato in teoria, ne permettono una migliore comprensione. Questo è l'ambito in cui si muove la didattica sperimentale nei circuiti, volta ad affiancare a concetti base dell'elettronica anche un riscontro pratico grazie alla realizzazione di sistemi e apparati sperimentali. Essi permettono di mostrare esperimenti più o meno semplici come complemento allo studio teorico, e di consentire l'acquisizione da parte degli studenti di una maggiore dimestichezza, anche pratica, con gli strumenti elettronici [1]. studio

1.1 La didattica sperimentale in laboratorio, in aula e da remoto

Le attività sperimentali possono essere realizzate nei laboratori che l'Università mette a disposizione degli studenti, purtroppo la dimensione odierna degli stessi è incompatibile col numero degli studenti presenti nelle aule e non consente di usufruire a pieno ed in condizioni ottimali degli strumenti convenzionali. Tale problematica determina la necessità di realizzare in altro modo le attività sperimentali; l'impossibilità di realizzare queste attività in laboratorio, ha portato a voler spostare gli esperimenti in aula rinunciando tuttavia alla strumentazione più convenzionale: peso e dimensioni infatti, non ne agevolano il trasporto al di fuori del laboratorio. Di qui la necessità di passare

a strumenti, realizzati mediante apparati sperimentali, di dimensioni più piccole e di minor peso, quindi facilmente trasportabili, meglio ancora se gli strumenti di misura sono realizzati e/o interfacciabili direttamente col calcolatore poiché in tal modo è possibile anche realizzare una proiezione su schermo dell'evoluzione dell'attività sperimentale e controllarli direttamente dal computer senza doversi concentrare su più oggetti contemporaneamente.

La diffusione di massa delle reti di telecomunicazioni e la sempre maggiore disponibilità di banda, ha aperto poi nuovi scenari alla didattica sperimentale. Sfruttando l'hardware prototipale realizzato per la didattica in laboratorio e in aula, è stato predisposto un vero laboratorio remoto per la didattica sperimentale sui circuiti: un sistema hardware e software che consente agli utenti di interagire, attraverso la rete internet, con processi fisici dislocati in altri posti, permettendo così la fruizione di veri esperimenti da luoghi diversi dal laboratorio; esso rappresenta il risultato di un'attività di progettazione e sviluppo portata avanti con gli anni da laureandi e ingegneri, ed è stato realizzato per rendere possibile l'esecuzione da remoto, di esperimenti fisicamente presenti presso il laboratorio di circuiti. Vediamo più nel dettaglio come è strutturato, considerando lo schema seguente:

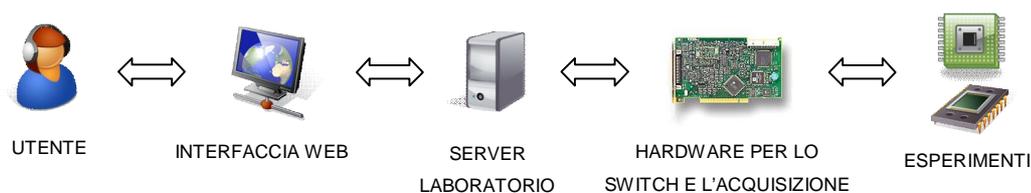


Figura 1.1 - Struttura del laboratorio remoto di circuiti

L'interfaccia web ed il server sono elementi caratteristici dell'architettura base di un laboratorio remoto, in quanto il loro scopo principale è la definizione del modo in cui un utente si interfaccia con esso; in particolare l'interfaccia web mostra all'utente il pannello frontale della strumentazione utilizzata, realizzato tramite l'ambiente di sviluppo LabVIEW 7.1, e garantisce il controllo remoto dell'esperimento e la visualizzazione a video dei risultati. Attualmente gli esperimenti disponibili per fini didattici, sono il circuito caotico di Chua e quello RLD.

E' da osservare che l'utilizzo di Internet, consente una gestione più efficiente delle risorse del laboratorio che diventano fruibili durante tutto l'arco della giornata e da qualsiasi computer collegato alla rete. Di conseguenza con l'utilizzo di questi strumenti gli studenti sono maggiormente motivati ad effettuare esperienze pratiche, acquisendo una più completa conoscenza della materia. Inoltre la piena accessibilità a differenza dei laboratori tradizionali, consente di effettuare esperienze ed esercizi pratici anche da casa a qualunque orario, anche durante le ore in cui i laboratori tradizionali restano chiusi; ciò comporta un incremento dell'efficienza ed una maggiore conoscenza dell'utilizzo degli strumenti. Inoltre tale laboratorio può essere utilizzato senza difficoltà anche da studenti con handicap motori.

1.2 Dispositivi per la didattica sui circuiti

1.2.1 Interfaccia USB

L'Universal Serial BUS (USB) è uno standard di comunicazione tra PC e periferiche. La nascita dell'Universal Serial BUS risale al 1995. Durante la conferenza annuale WinHEC95 (Windows Hardware Engineering Conference) un consorzio di produttori composto da Microsoft, IBM, Compaq, Digital, NEC e Northern Telecom ha posto le basi per la definizione di questo standard di comunicazione. Per favorirne la rapida diffusione, lo standard USB è stato messo a disposizione di tutti i produttori di hardware liberamente e gratuitamente. Secondo i maggiori produttori di hardware e software come Apple e Microsoft, l'USB possiede tutte le caratteristiche per imporsi come l'interfaccia del futuro; con essa, infatti, è possibile collegare tastiere, mouse, modem, stampanti, scanner, webcam, joystick, altoparlanti e altre periferiche utilizzando un solo tipo di connessione.

Grazie all'interfaccia USB è possibile collegare al PC, per mezzo di un singolo cavo composto da una coppia di conduttori, una grande varietà di dispositivi. Utilizzando degli speciali moltiplicatori di porte (HUB) è possibile, inoltre, creare una struttura su

sette livelli, alla quale per mezzo degli HUB è possibile, a sua volta, collegare e far lavorare contemporaneamente un elevato numero di periferiche.

Lo schema del connettore della porta di comunicazione USB è riportato in figura.



Figura 1.2 - Connettore USB

Lo Standard USB 1.0 supporta collegamenti 1.5 Mbit/s, la seguente velocità è adeguata per standard lenti come mouse, tastiere. Lo standard 1.1 aggiunge la “Speed” che innalza la velocità a 12Mbit/sec. La maggiore novità per la USB è l’innalzamento della velocità di trasferimento a 480 Mbit/sec.

Questa velocità così elevata consente all’USB di sovrintendere allo sviluppo dello standard USB ha rinominato USB 1.1 come USB 2.0 Full Speed e USB 2.0 come USB 2.0 *High Speed*. Tanti sono i vantaggi per l’uso di una periferica USB:

- Basso costo rispetto alla velocità di trasferimento di 480 Mbit/sec;
- Risparmio di corrente. Secondo le specifiche PC98 e PC99 vengono alimentate solo le periferiche utilizzate;
- Una periferica USB può essere collegata in qualsiasi momento anche a computer acceso. Il sistema operativo è in grado di riconoscere quando una nuova periferica viene collegata (Plug and Play);
- Tramite standard per la telefonia TAPI, possiamo collegare sia modem standard che ISDN;

- Standard di interfaccia veloce nella comunicazione con la periferica;
- Semplicità d'uso da parte gli utenti.

1.2.2 Interfaccia GPIB e USB-GPIB

Il *GPIB* (General Purpose Interface Bus) è un bus per l'interconnessione ed il controllo di apparecchiature elettroniche reso standard dalla IEEE e chiamati IEEE 488.1.

La seguente interfaccia nacque per permettere gli utenti di collegarsi con i calcolatori, dato che uno strumento di misura è dotato di un interfaccia GPIB.



Figura 1.3 – Connettore GPIB

L'IEEE 488.1 permette a 15 periferiche di condividere un Bus elettrico composto da 8 bit in parallelo con una connessione di tipo “daisy chain”. Un dispositivo più lento determina la velocità totale di comunicazione sul Bus nel controllo e nel trasferimento dati. Il massimo data rate è di circa 1 Mbyte nello standard originale, mentre nelle versioni più recenti è di circa 8 Mbyte; da questa analisi risulta evidente che lo standard GPIB è molto più performante dell'USB.

I Personal Computer presentano incompatibilità con la strumentazione dotata della seguente interfaccia è quindi bisogna dotare fisicamente il calcolatore di un'interfaccia GPIB, inserendo un'apposita scheda di espansione.

Il problema sorge con il personal computer portatile in cui la presenza di tali schede di espansione non è prevista poiché la tendenza tecnologica è quella di ridurre l'espansione interna. Da qui l'utilizzo dell'USB/GPIB nella didattica, ovvero di un'interfaccia che permetta di collegare PC portatili che sicuramente sono dotati di porte USB a strumentazione come Oscilloscopi o Generatori di segnale dotati di porte

GPIB; grazie a questa soluzione è possibile sfruttare le prestazioni dello standard IEEE 488 e le qualità dell'USB [3].



Figura 1.4 - Esempio di connessione con interfaccia USB e PC portatile

1.2.3 Schede d'acquisizione input/output analogiche/digitali

La scheda di acquisizione input/output analogica/digitale è un dispositivo hardware che consente la raccolta automatizzata di segnali analogici e digitali. Il suo compito è quello di trasferire nel modo più adatto e veloce i segnali esterni al PC, in modo tale da poterli elaborare e analizzare. In particolare, essa si adopera a “catturare” i segnali analogici e digitali uscenti da una qualsiasi sorgente e a salvarlo in un file riconoscibile e utilizzabile dal PC, preservandone naturalmente tutte le caratteristiche. Allo stesso modo questo dispositivo permette, nella maggior parte dei casi, il processo contrario. Inoltre, una scheda di acquisizione può presentare anche delle uscite controllate tramite degli appositi algoritmi. Il suo impiego, nel laboratorio, è relativo alla possibilità di fornire al PC fisso o portatile i dati da esaminare, poiché si collega al personal computer tramite l'interfaccia USB.

Esempi di schede di acquisizione sono mostrate in figura 1.5.

Attualmente sul mercato sono disponibili diverse tipologie di schede di acquisizione aventi prestazioni, caratteristiche di funzionamento e costi differenti, secondo le proprie esigenze; quelle attualmente a disposizione nel laboratorio di circuiti sono la scheda **PCI-6040E** e la **PCI-5102**, entrambe prodotte dalla National Instruments.



Figura 1.5 - Tipi di schede d'acquisizione

La **PCI-6040E** è mostrata in figura 1.6.



Family	Bus	Analog Inputs	Input Resolution	Max Sampling Rate	Input Range	Analog Outputs	Output Resolution	Output Rate	Output Range	Digital I/O	Counter/Timers	Triggers
NI 6040E	PCI, PXI	16 SE/8 DI	12 bits	500 kS/s	± 0.05 to ± 10 V	2	12 bits	1 MS/s	± 10 V	8	2, 24-bit	Analog, digital

Figura 1.6 - NI PCI-6040E

E' una scheda di acquisizione multifunzione, le cui prestazioni in termini di precisione e di velocità di acquisizione, risultano determinate dall' insieme calcolatore-scheda di acquisizione-periferica. In particolare, essa presenta 16 ingressi analogici Single Ended, 2 uscite analogiche, 8 ingressi/uscite digitali, una tensione nominale di ingresso da ± 0.05 V a ± 10 V. L' acquisizione può essere condotta su singolo canale per volta (single acquisition) o su più canali contemporaneamente (multipleacquisition). Nel

caso di acquisizione multipla, è possibile stabilire quali sono i canali che interessano e l'ordine con cui devono essere acquisiti.

La massima frequenza di campionamento, nel caso di acquisizione singola, è di 500Ksample/sec (Sample: unità di misura che indica il numero di campioni), mentre nel caso di acquisizione multipla su N canali, è di $(250/N)$ Ksample/sec per canale. Come si vede la frequenza massima del campionatore , scende a 250Ksample/sec nel caso di acquisizione multipla; la frequenza effettiva di campionamento per ogni canale è pari alla frequenza del campionatore diviso il numero di canali selezionati. Inoltre, è dotata di una memoria Buffer di tipo FiFo (First In First Out, cioè il primo elemento in ingresso è anche il primo elemento ad uscire) in cui vengono temporaneamente immagazzinati 1024 campioni. Il buffer allocato in memoria è di tipo circolare costituito da due puntatori, uno per la lettura ed uno per la scrittura. Per cui in operazioni di acquisizioni dati ad alta velocità, la scrittura avviene con un ritmo più veloce della lettura provocando, dopo un certo tempo, il riempimento del buffer . Inoltre secondo le specifiche lo Stream to disk, cioè quella modalità di acquisizione che dipendentemente dall' hardware del sistema che ospita la scheda, acquisisce e salva direttamente sulla memoria di massa, avviene senza saturazione per velocità di campionamento inferiori a 250Ksample/sec (system dependent).

Sebbene le prestazioni di questa scheda siano ottimali, presenta il problema del costo eccessivo e di un limitato numero di ingressi/uscite digitali. In questo momento i suoi 8 bit digitali sarebbero completamente occupati per il controllo del circuito di Chua, di conseguenza il laboratorio resterebbe limitato ad un solo esperimento.

Data l'esigenza di voler collegare anche il circuito RLD si è deciso negli anni, di realizzare in sostituzione, un circuito di switch con un numero di bit i/o superiore. Per i dettagli su tale circuito si rimanda al capitolo 2.

1.2.4 Generatori di segnali

Un generatore di segnale è un dispositivo che genera forme d'onda di frequenza, forma ed ampiezza di molteplici tipi, note e regolabili, in altre parole è uno strumento in grado

di emettere da una o più uscite un segnale periodico variabile in ampiezza ed in frequenza in genere di tipo onda quadra, onda sinusoidale, onda triangolare o a dente di sega. Questi segnali possono essere opportunamente modificati regolando la frequenza, l'ampiezza e l'offset. Il generatore di segnale è, in alcuni modelli, in grado di generare il rumore bianco; viene utilizzato a scopi di misurazione o di controllo; infatti le "sollecitazioni" che produce sono quelle necessarie per verificare il corretto funzionamento di dispositivi ed apparati elettronici, ossia per effettuare prove, testare dispositivi o semplicemente alimentare un circuito. In laboratorio è facile reperire un generatore di segnali adeguato all'esperimento in quanto fa parte della strumentazione di base di ogni laboratorio che si rispetti. I generatori da laboratorio, sebbene abbiano prestazioni di tutto rispetto in quanto a capacità di generazione e precisione della forma d'onda generata, presentano i difetti del costo sostenuto e delle dimensioni e peso non proprio trascurabili.

Quindi la situazione è più complessa qualora si volessero effettuare gli esperimenti in altre situazioni, esterne ai laboratori, come una dimostrazione didattica in aula, proprio per questo ultimo motivo.

Inoltre, nella maggior parte dei casi, i generatori di segnali disponibili sul mercato sono dotati di una interfaccia parallela GPIB per il collegamento a personal computer che, quindi, avviene mediante una interfaccia USB to GPIB. Mentre solo recentemente sono stati introdotti generatori di segnali che prevedano un'interfaccia USB per interagire con un personal computer, avendo, però, costi elevati.

Col tempo si è allora pensato alla realizzazione di un generatore di segnale che, oltre a presentare le caratteristiche necessarie di frequenza e ampiezza in relazione ai segnali generati, fosse portatile e totalmente controllabile da computer via USB, in modo tale da sostituire il convertitore USB/GPIB con un semplice cavo USB. Ovviamente le sue prestazioni non possono competere con quelle di uno strumento commerciale, ma sono comunque sufficienti a soddisfare le più comuni necessità sperimentali. Per i dettagli su tale generatore, si rimanda ai capitoli successivi.

1.2.5 Oscilloscopi

L'oscilloscopio è uno strumento di misura elettronico che consente di visualizzare, su un grafico bidimensionale, l'andamento temporale dei segnali elettrici e di misurare abbastanza semplicemente tensioni, correnti, potenze ed energie elettriche. L'asse orizzontale del grafico solitamente rappresenta il tempo, rendendo l'oscilloscopio adatto ad analizzare grandezze periodiche. L'asse verticale rappresenta la tensione. La frequenza massima dei segnali visualizzabili, così come la risoluzione temporale, ovvero la più rapida variazione rilevabile, dipende dalla banda passante dello strumento, a sua volta dipendente dalla qualità e in ultima analisi dal costo. Si spazia dalle decine di megahertz adatti per lavorare con segnali audio e televisivi, ai costosi modelli digitali da diversi Gigahertz. Può essere considerato uno strumento universale, collegandogli appropriati trasduttori, si può analizzare qualsiasi fenomeno fisico, anche eventi casuali e non ripetitivi. Un oscilloscopio classico è tipicamente costituito da una scatola rettangolare su cui è presente uno schermo e numerose manopole e pulsanti di comando. Allo schermo è sovrapposto un reticolo allo scopo di favorire la lettura dei dati. Ogni intervallo del reticolo è chiamato *divisione*, sull'asse orizzontale le divisioni sono solitamente 10, sull'asse verticale variano da 6 in su, dipende dalla geometria dello schermo. Questa modalità costruttiva è rimasta invariata per decenni fino all'avvento degli oscilloscopi digitali dotati di microprocessore, e di conseguenza in grado di poter installare a bordo veri e propri sistemi operativi, i quali uniti alla tecnologia degli schermi sensibili al tocco, hanno cambiato radicalmente l'aspetto e l'operatività di questo strumento. Il segnale da misurare viene introdotto attraverso un apposito connettore, solitamente di tipo coassiale BNC adatto per frequenze relativamente basse, o tipo N, utilizzato quando le frequenze in gioco sono molto elevate. Al connettore d'ingresso possono essere collegate le sonde, accessori particolari usati per prelevare segnali dai circuiti studiati.

La scala verticale è espressa in volt per divisione, e può essere regolata da decine a millesimi di volt, sono comunque disponibili modelli in grado di effettuare misure dell'ordine dei microvolt. L'altezza iniziale del grafico (offset) può comunque essere decisa dall'utente, così come è possibile escludere la componente in corrente continua

presente nel segnale in esame, nonché scegliere l'impedenza di ingresso. In questo modo si ottiene la visualizzazione di un grafico di tensione in funzione del tempo. Se il segnale è periodico, è possibile ottenere una traccia stabile regolando la base dei tempi in modo da coincidere con la frequenza del segnale o un suo sottomultiplo.

La base dei tempi, non essendo sincronizzata con il segnale in analisi, impedisce di avere una traccia stabile e ferma, questa fluttuerà lentamente da destra a sinistra o viceversa. Per ottenere una traccia stabile gli oscilloscopi moderni dispongono di una funzione chiamata *trigger* (innesco), questo circuito inventato nel 1946 dai due fondatori della società Tektronix, fa partire la scansione solo in corrispondenza del verificarsi di un evento sul segnale in ingresso, per esempio il superamento di una soglia di tensione positiva o negativa. Dopo avere completato la scansione da sinistra a destra, l'oscilloscopio rimane in attesa di un nuovo evento. In questo modo la visualizzazione rimane sincronizzata al segnale e la traccia è perfettamente stabile. La soglia di sensibilità del trigger, così come altri parametri è regolabile dall'utente.

Molti oscilloscopi permettono di escludere la base dei tempi e fornire all'asse orizzontale un segnale esterno: è la modalità X-Y, utile per visualizzare le relazioni di fase tra due segnali in ambito radiotelevisivo. Applicando due segnali sinusoidali in rapporto armonico di frequenza agli ingressi, sullo schermo vengono visualizzate particolari figure, chiamate figure di Lissajous, da cui è possibile inferire i rapporti di fase e di frequenza tra essi. Alcuni oscilloscopi hanno sullo schermo dei cursori che possono essere spostati ed utilizzati per misurare con precisione intervalli di tempo o differenze di potenziale. Molti oscilloscopi hanno due o più ingressi verticali, consentendo di visualizzare diversi segnali contemporaneamente. Le regolazioni verticali sono separate mentre la base dei tempi ed il trigger è comune.

Esistono modelli con doppio trigger, che permettono di visualizzare un segnale con una base dei tempi diversa. In questo modo è possibile avere la modalità *zoom*, in cui una porzione del segnale mostrato su una traccia può essere mostrato espanso sull'altra traccia. Per analizzare eventi non ripetitivi alcuni oscilloscopi sono dotati di *memoria di traccia*, un sistema che mantiene visualizzata l'ultima traccia apparsa. In alcuni modelli digitali la scansione può durare ore, e la traccia visualizzata scorre sullo schermo da

destra a sinistra come avverrebbe in un registratore su striscia di carta. Normalmente ogni oscilloscopio è dotato di un circuito di calibrazione, il quale genera un segnale continuo, costituito da un'onda quadra di ampiezza e frequenza nota. Collegando l'ingresso di misura all'uscita di calibrazione è possibile controllare il funzionamento e fare pratica con lo strumento, ma soprattutto calibrare periodicamente le sonde. L'oscilloscopio in dotazione presso il laboratorio di circuiti è il TDS 220 della Tektronix ed ha le seguenti caratteristiche principali: larghezza di banda è di 100MHz, la frequenza di campionamento è di 1 GS/s su ciascun canale, i canali sono due e sono identici con trigger esterno. Lo strumento ha una buona sensibilità e parte da 2 mV/div a 5V/div in tutte le modalità e a 20MHz a 10mV/div in rivelazione di picco.

Sebbene tale strumento sia facilmente trasportabile, ai fini della didattica in aula e da remoto, è stato opportuno sostituirlo con un oscilloscopio "virtuale", per limitare al minimo il numero di apparati da dover trasportare e per avere su PC tutti i controlli in un'unica interfaccia: l'interfaccia, già realizzata per un progetto precedente, utilizza un PC, la scheda d'acquisizione analogica NI-5102 ed un software di controllo sviluppato in ambiente LabView 7.1.

1.3 Alcuni circuiti per la didattica sperimentale

I circuiti più appropriati per il laboratorio di Circuiti devono avere le seguenti caratteristiche:

- Essere piccoli, in modo da poter essere facilmente trasportabili;
- Essere semplici, la complessità circuitale è in genere portatrice di problemi di realizzazione e di stabilità di funzionamento;
- Essere robusti nel loro comportamento, in modo che non alterino il loro comportamento al variare di parametri esterni come temperatura o tensione di alimentazione per garantire sempre il successo delle dimostrazioni;

- Non devono richiedere tensioni di alimentazioni elevate, in modo da non essere potenzialmente pericolosi e da richiedere alimentatori semplici;
- Non devono funzionare a frequenze elevate, per garantire l'usabilità con strumentazione semplice e di prestazioni non eccessive;
- Devono mostrare un ampio spettro di dinamiche, per mostrare una larga parte dei fenomeni caratteristici dei sistemi caotici, biforcazioni successive ben distinte, caos, finestre nel caos;
- Devono essere facilmente controllabili, nel senso che sia facile variare i parametri di funzionamento per mostrare i vari regimi dinamici e come essi si presentino al variare dei parametri circuitali;

A valle di tutte queste esigenze, sono già stati realizzati due circuiti: il circuito RLD e il circuito di Chua che saranno descritti nel capitolo 2. Sono stati realizzati questi due circuiti non lineari, in quanto è molto interessante studiare e osservare la loro dinamica caotica, ciò non toglie che possono essere realizzati circuiti più semplici come quelli lineari[8].

1.3.1 Circuiti lineari, non lineari e caotici

Nell'ambito della teoria dei circuiti, una grossa distinzione può essere fatta tra i circuiti lineari e quelli non lineari: un circuito si dice lineare se è composto da soli componenti lineari (resistori, condensatori, induttori ecc.); in tal caso esso è descrivibile mediante leggi fisiche di tipo lineare, cioè in generale, da equazioni differenziali lineari a coefficienti costanti [3].

Nella formazione di un ingegnere lo studio dei sistemi lineari gioca un ruolo fondamentale rispetto a quello dei sistemi non lineari, l'analisi di questi ultimi è limitata, nella maggior parte dei casi, alla determinazione del punto di funzionamento statico e dei parametri differenziali per la linearizzazione intorno al suddetto punto di funzionamento. Questo comportamento è dovuto sostanzialmente al fatto che mentre si ha disponibilità di strumenti analitici potenti applicabili al caso generale dei sistemi

lineari, per i sistemi non lineari non esistono in pratica metodi generali di soluzione. La determinazione della soluzione in maniera analitica nel caso dei sistemi non lineari è possibile solo per particolari problemi in maniera molto limitata [4].

Ricordiamo che si definisce “soluzione di regime” una soluzione di un circuito che, per $t \rightarrow \infty$ risulti indipendente dalle condizioni iniziali; il concetto di “comportamento asintotico” è l’estensione di quello di soluzione di regime quando esso non è applicabile. I possibili comportamenti asintotici che si presentano nei sistemi lineari e non lineari, in un buon numero di casi, presentano importanti proprietà di regolarità, vedremo, però, che esistono delle dinamiche che non possono essere inquadrare in queste categorie e a cui faremo corrispondere il concetto di caos.

Una buona parte delle soluzioni di regime dei sistemi presentano alcune proprietà di regolarità che possono essere classificate secondo le seguenti tipologie:

1. soluzioni costanti
2. soluzioni sinusoidali
3. soluzioni armoniche, formate da sinusoidi a frequenze multiple di quelle dell’eventuale forzamento.
4. soluzioni subarmoniche, formate da sinusoidi a frequenze anche sottomultiple di quella del forzamento.
5. soluzioni quasi periodiche, formate da sinusoidi a frequenze incommensurabili¹.

Visti nello spazio delle fasi gli attrattori ottenuti sono nel primo caso un punto isolato detto anche punto fisso, il secondo, il terzo ed il quarto caso appaiono come delle traiettorie chiuse e sono detti globalmente cicli limiti. Il quinto caso è particolare rispetto ai precedenti, visto nello spazio delle fasi questa soluzione appare avvolgersi su di una copia diffeomorfa di un toro di dimensione inferiore rispetto alla dimensione dello spazio delle fasi (di dimensione pari al numero delle frequenze). Le prime due

¹ Due numeri reali sono incommensurabili se il loro rapporto non è un numero razionale.

categorie e la quarta sono caratteristiche dei sistemi lineari nel senso che esauriscono completamente le possibili dinamiche delle loro soluzioni di regime. Nel caso dei sistemi non lineari generici possiamo riscontrare ogni tipologia delle suddette soluzioni; in più si nota lo sviluppo di un nuovo tipo di soluzione, non più periodica, che nello spazio delle fasi appare come una traiettoria che resta confinata in una regione limitata dello spazio, ma si avvolge su se stessa (senza mai intersecarsi), questo genere di soluzioni si definiscono soluzioni caotiche ed un sistema che le dovesse presentare lo si definisce a sua volta caotico.

Di seguito vedremo alcuni tra i più importanti circuiti lineari e non lineari ai fini didattici.

1.3.2 Circuiti del primo ordine

La carica e la scarica di un condensatore/induttore consentono di mostrare come evolve un semplice circuito lineare dinamico RC o RL, mostrati in figura, con induttore e condensatore lineari e tempo invarianti.

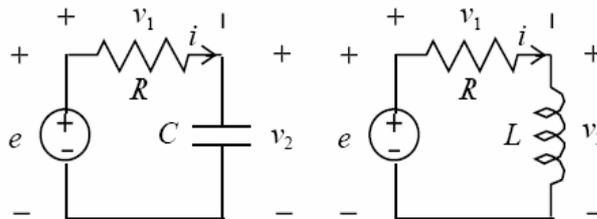


Figura 1.7 - Circuito RC ed RL

In particolare la carica ci permette di analizzare l'evoluzione forzata in cui le grandezze di stato del circuito all'istante iniziale sono tutte nulle e, quindi, l'energia immagazzinata nei bipoli dinamici è uguale a zero. In evoluzione forzata le intensità di corrente del circuito vengono prodotte solo attraverso l'azione dei generatori.

Invece la scarica ci permette di analizzare l'evoluzione libera che si ha quando un circuito dinamico è privo di generatori o i generatori che contiene sono tutti spenti. In un circuito in evoluzione libera le grandezze di stato dello stesso all'istante iniziale sono diverse da zero, quindi è l'energia immagazzinata all'istante iniziale nei bipoli dinamici

che produce le correnti del circuito . E' possibile realizzare l'esperimento utilizzando il generatore e dimensionando opportunamente i componenti, ovvero un resistore ed un condensatore, nel caso del circuito RC, o un resistore ed un induttore, nel caso del circuito RL.

Nel caso del circuito RC l'esperimento consiste nel considerare il seguente circuito:

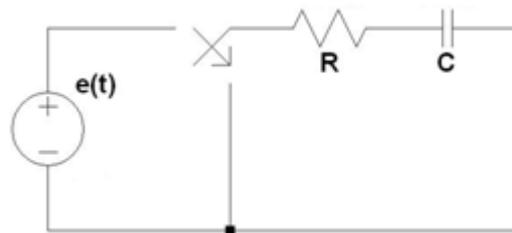


Figura 1.8 - Circuito per la carica/scarica del condensatore

L'esperimento consiste nel far caricare la capacità quando l'impulso è attivo e quindi calcolare il suo tempo di "carica", ed infine vedere la scarica del condensatore quando l'impulso è andato a zero calcolando il tempo di scarica. I risultati sono riportati in figura.

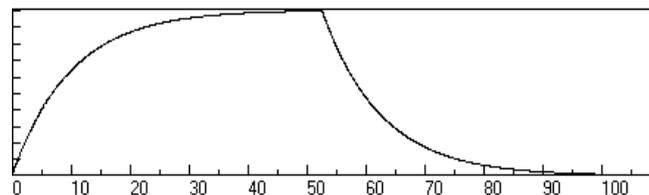


Figura 1.9 - Carica e scarica condensatore

1.3.3 Circuito lineare del primo ordine con forzamento sinusoidale

Si consideri, ora, il caso in cui la tensione impressa dal generatore di tensione in figura 1.7, sia sinusoidale con pulsazione ω ($\omega = 2\pi f$ dove f è la frequenza), ampiezza massima E e fase iniziale ϕ ,

$$e(t) = E \cos(\omega t + \phi)$$

Vogliamo ora determinare la soluzione dell'equazione differenziale:

$$RC \frac{dv_2}{dt} + v_2 = e(t)$$

che è in generale, appunto, non omogenea.

La soluzione effettiva del circuito, nel caso di generatore sinusoidale, è allora:

$$v(t) = (V_0 - V \cos \alpha) e^{-t/\tau} + \frac{E}{\sqrt{(\omega\tau)^2 + 1}} \cos[\omega t + \varphi - \arctg(\omega\tau)] \quad (1)$$

Il termine transitorio dipende dalla condizione iniziale, mentre quello di regime non è dipendente dalla condizione iniziale.

Val la pena di osservare che, qualunque sia la condizione iniziale si ha che:

$$v(t) \rightarrow \frac{E}{\sqrt{(\omega\tau)^2 + 1}} \cos[\omega t + \varphi - \arctg(\omega\tau)] \quad \text{per } t \rightarrow \infty$$

Il primo termine della (1) è il termine transitorio e tende esponenzialmente a zero, qualunque sia il valore iniziale della tensione, con la costante di tempo τ . Dopo che il termine transitorio si è estinto, la soluzione del circuito è sinusoidale nel tempo, con la stessa pulsazione del generatore. Per tale ragione alla soluzione particolare $v(t)$ con $t \rightarrow \infty$ si dà il nome di soluzione di regime sinusoidale.

In figura 1.10 viene riportato il grafico della tensione $v(t)$ in evoluzione forzata (in funzione del tempo).

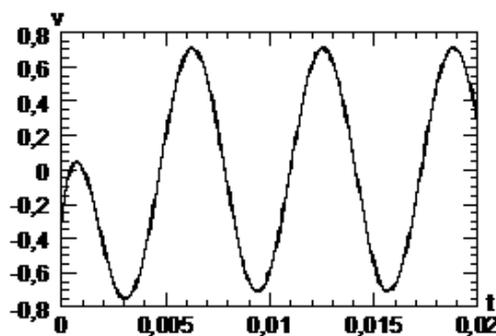


Figura 1.10- Evoluzione forzata della tensione sul condensatore

Capitolo 2

HARDWARE PER LA DIDATTICA SPERIMENTALE SUI CIRCUITI

In questo capitolo daremo dei cenni sull'hardware a disposizione per didattica sperimentale sui circuiti e per il quale è stato realizzato il software, descritto nel capitolo 3. Dal lavoro svolto da laureati e laureandi sono nati negli anni, una serie di circuiti che costituiscono la parte hardware del laboratorio "portatile", il cui scopo è finalizzato alla didattica sperimentale in aula e da remoto[8]. L'hardware in questione è costituito da una scheda prototipo che integra, un generatore di segnale, una circuito di acquisizione, un circuito di switch che pilota un multiplexer a relè, e da circuiti dimostrativi implementati, che attualmente sono due circuiti non lineari con dinamiche caotiche, in particolare il circuito di Chua ed un circuito a risonanza non lineare (RLD).

Completa l'hardware in dotazione per la didattica sperimentale sui circuiti, la scheda d'acquisizione veloce PCI-5102 prodotta dalle National Instruments, che rappresenta anche l'unico componente commerciale. Interfacciando l'hardware ad un personal computer, desktop o portatile, provvisto di scheda 5102 e porta USB, è possibile controllare il generatore di segnali o interagire con gli esperimenti, mediante l'utilizzo di specifici programmi realizzati in ambiente Labview. Tali programmi realizzano delle interfacce grafiche che ricordano molto quelle degli oscilloscopi tradizionali e consentono di variare alcuni parametri dei circuiti e osservarne a video gli effetti.

In figura 2.1 è mostrato lo schema a blocchi dell'hardware in questione.

L'integrato con interfaccia USB si occupa dell'invio dei comandi dal PC, verso il generatore di segnali, il circuito di acquisizione e il circuito di switch della scheda prototipo, tramite il BUS SPI.

Il generatore di segnali consente di realizzare forme d'onda sinusoidali, triangolari e quadre in un ampio range di frequenza e con una precisione che, nell'ambito in cui è

utilizzato (generare un segnale in ingresso al circuito RLD), non è eccessivamente lontana da quella di un generatore di riferimento.

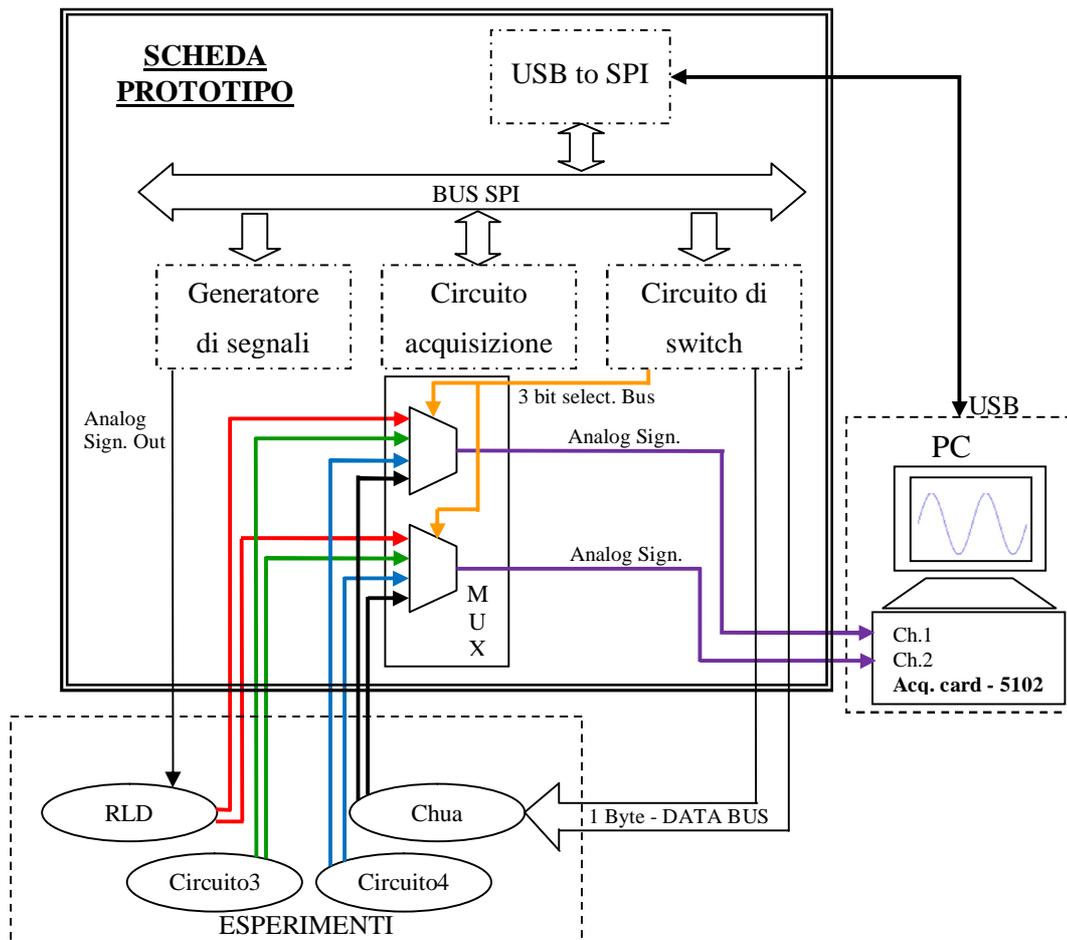


Figura 2.1 - Schema a blocchi dell'hardware per la didattica sui circuiti

Il circuito di acquisizione è composto da un DAC e un ADC che in futuro, una volta sviluppati i driver necessari al loro funzionamento, consentiranno l'input/output di segnali costanti.

Il circuito di switch consente di variare digitalmente la resistenza del circuito di Chua, in modo da poter osservare come varia la sua dinamica caotica al variare di tale valore, e di pilotare il multiplexer che effettua la selezione dell'esperimento d'interesse; in base alla scelta effettuata, viene messo in comunicazione tale esperimento con la scheda

d'acquisizione PCI-5102; per ciascun esperimento vengono acquisiti due segnali, essi vengono campionati e forniti al calcolatore, il quale successivamente li elabora mediante programmi labview. I risultati di tali elaborazioni, sono mostrati dagli stessi programmi, sotto forma di dati e diagrammi.

Di seguito verranno descritti in dettaglio, ciascuno dei componenti hardware succitati.

2.1 La scheda “prototipo”

Questo circuito controllato via USB, integra come già indicato: un generatore di segnale, un circuito di acquisizione ed un circuito di switch che pilota un multiplexer a relè.

La scelta d'integrare tutto su un'unica scheda, prosegue quel percorso di sostituzione della strumentazione convenzionale con apparati sperimentali più piccoli e maneggevoli, consentendone un trasporto più agevole per le attività didattiche in aula.

Per la realizzazione hardware di questo circuito sono stati utilizzati i seguenti circuiti integrati:

- Modulo USB (UM245R);
- Port Expander (MCP23S17);
- DAC (TLV5616CP);
- ADC (TLC2543IN);
- Transistor Array (ULN2003AN);
- Relè (MT2-C93432);
- DDS (AD9834);
- DAC (MCP4922);
- Operazionale sottrattore (TL082CP);
- Moltiplicatore (AD835AN);
- Amplificatore di tensione (LM6172IN);

- Amplificatore di corrente (THS6012I);

2.1.1 Il protocollo di comunicazione SPI

E' una semplice interfaccia che permette la comunicazione tra controllori e circuiti integrati periferici o l'interconnessione fra due o più controllori.

Il Bus SPI usa un protocollo sincrono, cioè la trasmissione e la ricezione dei dati è guidata da un segnale di clock (SCKL) generato dal controllore. L'interfaccia SPI consente di collegare diversi dispositivi, il controllore seleziona uno di essi con il segnale CS (Chip Select, a volte anche denotato come SS, Slave Select).

La SPI usa un modello Master-Slave (figura 2.2), il dispositivo Master fornisce il segnale di clock e determina lo stato del chip select, cioè attiva lo Slave se vuole comunicare con esso, quindi CS e SCLK sono output. Il dispositivo Slave riceve il clock e il chip select dal Master, per cui CS e SCLK sono input. Ciò significa che c'è un solo Master mentre il numero di Slave è limitato dal numero di chip select.

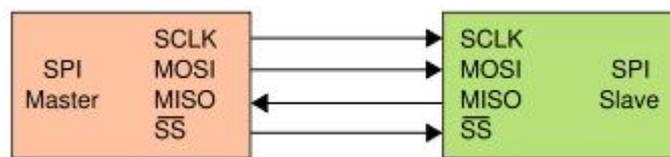


Figura 2.2 - Bus SPI con un master ed uno slave

Il Bus SPI consiste di quattro linee di segnale:

- SCKL, Serial Clock, è generato dal master per sincronizzare i dati di input e output trasmessi, quindi è un input per tutti i dispositivi Slave.
- MOSI, Master Out Slave In: su questo link, viaggiano i dati dal Master allo Slave; Il trasferimento dei dati avviene solo in questa direzione.
- MISO, Master In Slave Out, è generato dallo Slave, quindi il segnale MISO è definito come input dal Master e come output dallo Slave. Il trasferimento dei dati è solo in una direzione, da uno Slave al Master.
- SS, Slave Select, è generato dal Master per attivare gli Slave.

Nel caso in cui il dispositivo Master comunica con più dispositivi Slave, come mostrato in figura 2.3, il Master dovrà avere per ogni Slave un segnale di CS indipendente, inoltre dovrà generare una frequenza di clock più o meno uguale alla frequenza massima che i dispositivi Slave supportano. Per quanto riguarda i dispositivi Slave, non attivati dal Master, dovranno ignorare il segnale di clock e i segnali MOSI e MISO.

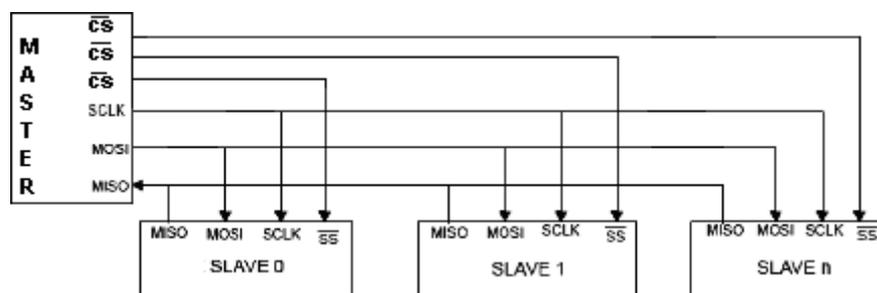


Figura 2.3: SPI Bus con un Master e tre Slave

Per la realizzazione del circuito in oggetto, sono state utilizzate tutti e quattro i segnali che costituiscono il Bus SPI, in particolare, il segnale MOSI per il trasferimento dei dati in uscita, inviati dal PC, tramite il modulo USB che funge da Master, agli integrati periferici, ovvero al port expander, ai due DAC, all' ADC e al DDS che rappresentano gli Slave; il segnale MISO, per il trasferimento dei dati d' input per il PC, inviati da uno degli Slave al Master; i segnali di Chip Select per l' attivazione dei componenti Slave da parte del Master ed infine il segnale di clock per le tempificazioni nelle operazioni di lettura e scrittura nei vari integrati.

In figura 2.4 è rappresentato lo schema a blocchi del circuito realizzato, dove si vedono facilmente le parti che lo compongono, ovvero, quella relativa al circuito di acquisizione, allo switch ed, infine, al generatore di segnale.

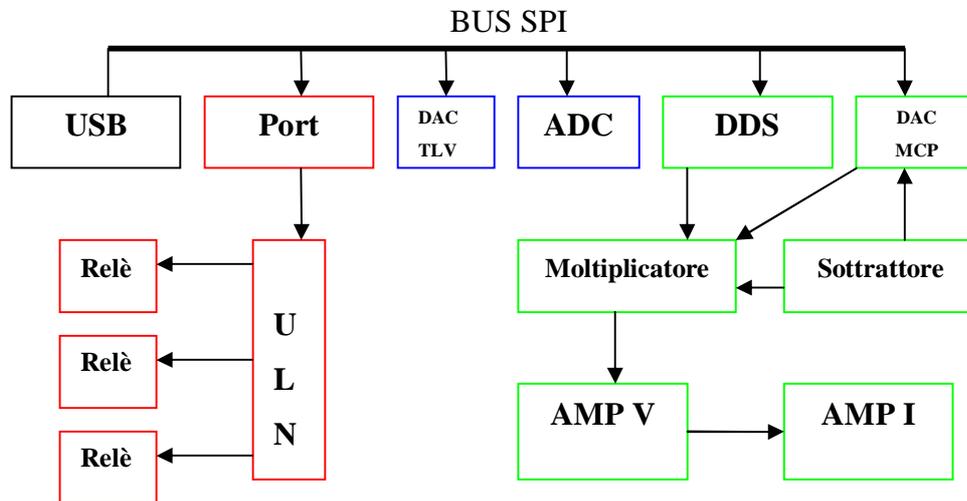


Figura 2.4 - Circuito di acquisizione, generatore di segnali, circuito di switch

2.1.2 Il generatore di segnali

Analizzeremo adesso ciascun componente della scheda prototipo, partendo dal generatore di segnali.

Le caratteristiche del generatore di segnali realizzato possono essere così riassunte:

- generazione di onde sinusoidali, quadre e triangolari
- regolazione della frequenza
- regolazione dell'ampiezza
- regolazione dell'offset continuo.

Da un punto di vista hardware, è composto dai seguenti blocchi:

- circuito di controllo e d'interfaccia
- oscillatore d'onda a frequenza variabile
- amplificatore a guadagno programmato
- generatore di offset
- amplificatore d'uscita.

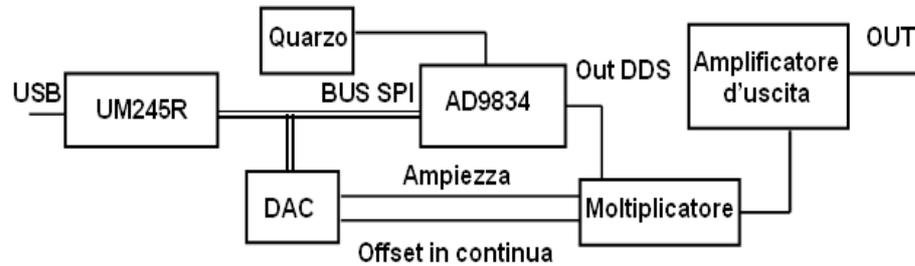


Figura 2.5 - Schema a blocchi del generatore di segnale portatile

Il modulo UM245R si occupa dell'invio dei comandi dal PC verso il generatore tramite il BUS SPI, l'oscillatore quarzato fornisce il segnale di clock ai due circuiti digitali ovvero il DDS e il DAC che, opportunamente pilotati forniscono rispettivamente la forma d'onda alla frequenza voluta e i valori di ampiezza ed offset che questa deve avere. Questi tre segnali così ottenuti vengono manipolati da un moltiplicatore analogico che fornisce in uscita la versione non amplificata del segnale d'uscita, sarà compito infine dello stadio d'uscita amplificare tale segnale.

Grazie allo sviluppo dei circuiti integrati è stato possibile sviluppare un valido generatore di segnale con un numero relativamente basso di componenti. Una sua realizzazione con componenti discreti avrebbe richiesto l'uso di molti amplificatori operazionali, non sarebbe stato facilmente controllabile dal personal computer e sarebbe stata certamente critica la sua messa a punto.

Il sintetizzatore DDS è la parte del circuito che è effettivamente dedicata alla generazione del segnale, è realizzato con un circuito integrato dedicato allo scopo. Il DDS (Direct Digital Synthesis), crea digitalmente forme d'onda arbitrarie e frequenze da una singola, fissata sorgente di frequenza. In pratica un flusso di dati digitali che rappresentano la forma campionata del segnale analogico viene applicato alla porta digitale di un convertitore D/A che viene fatto funzionare con una frequenza di clock costante. I dati digitali vengono a questo punto convertiti in una serie di valori di tensione che approssimano la forma del segnale analogico. Dopo aver filtrato questi valori di tensione si riproduce la forma d'onda del segnale analogico. Un circuito DDS basilare consiste di un controllore elettronico, una RAM, una frequenza di riferimento, un contatore ed un DAC.

L'unico produttore DDS è l'Analog Devices, che ne produce una vasta gamma tra cui è stato scelto l'AD9834 perché ha prestazioni sufficienti alle nostre necessità, prevedendo una frequenza d'output che varia da 0Hz a 25MHz [7].

Le caratteristiche di questo generatore di segnali possono essere così riassunte

Caratteristiche	Specifiche
Forme d'onda	sinusoidale, triangolare, quadra
Ampiezza	$100\text{mV} \div 10\text{V}^*$
Offset	$-10\text{V} \div +10\text{V}^*$
Range di frequenza	$0.5 \div 2\text{MHz}^{**}$
Impedenza d'uscita	50Ω
Interfaccia PC	USB
Alimentazione	Rete AC 50/60Hz oppure $\pm 15\text{V}$

Figura 2.6 - Specifiche tecniche del generatore di segnali portatile

* non è possibile impostare ai massimi valori ampiezza e offset contemporaneamente per problemi di saturazione dell'amplificatore finale. ** valori ricavati impostando una forma d'onda di tipo sinusoidale.

2.1.3 Circuito di switch, circuito di acquisizione e multiplexer

Per realizzare l'interfacciamento tra il PC ed il resistore a controllo digitale del circuito di Chua è stato realizzato una **circuito di switch** con interfaccia USBtoSPI. Esso dispone di 16 bit digitali, configurabili come ingressi/uscite indipendentemente, può così gestire in contemporanea, gli 8 terminali digitali d'ingresso/uscita, adoperati per generare il segnale digitale in ingresso alla interfaccia digitale del circuito di Chua e per controllare un doppio **multiplexer**, tramite il quale è possibile selezionare uno tra gli esperimenti disponibili. Il **circuito di acquisizione** è composto invece da un DAC ed un ADC che in futuro, una volta sviluppati i driver necessari al loro funzionamento, consentiranno l'input/output di segnali costanti.

Le caratteristiche di tale circuito possono essere così riassunte:

- 1 uscita analogica a 12 bit
- 11 ingressi analogici a 12 bit
- possibilità di ulteriore espansione nel numero dei segnali aggiungendo ulteriori circuiti integrati a basso costo
- funzionamento "statico" del circuito a frequenze prossime allo zero.

Per la realizzazione hardware di tali circuiti e del multiplexer a relè, sono stati utilizzati i seguenti componenti:

- Circuito di controllo e di interfaccia;
- Port expander;
- Convertitore D/A;
- Convertitore A/D;
- High-voltage Darlington transistor array;
- Relè[8].

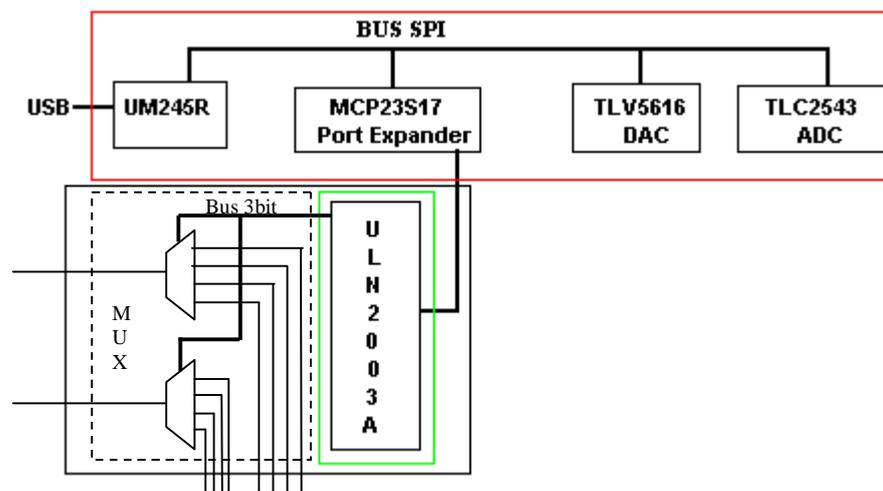


Figura 2.7 - Schema a blocchi del circuito I/O A/D e del multiplexer a relè

2.2 Descrizione degli integrati per la scheda prototipo

2.2.1 Il modulo UM245R

Il dispositivo UM245R, prodotto dalla FTDI, ha come funzione principale, quella di permettere la trasmissione dei dati tra un PC e gli altri circuiti integrati.

L'UM245R è dotato di 24 pin come mostrato in figura 2.8.



Figura 2.8 - Modulo USB UM245R

Per la realizzazione del circuito in oggetto sono serviti solo 10 pin, in particolare:

- DB0, DB1, ..., DB7 = generico bit di input/output
- GND = massa
- VCC = tensione di alimentazione (+5V)

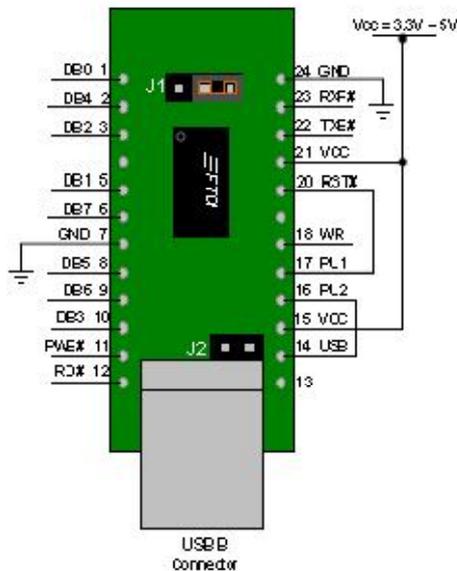


Figura 2.9 - Piedinatura del modulo UM245R

I generici bit input/output sono stati configurati come segue:

- DB0 = MOSI;
- DB1 = SCLK;
- DB2 = SS;
- DB3 = SS;
- DB4 = SS;

- DB5 = SS;
- DB6 = SS;
- DB7 = MISO[8].

2.2.2 II DAC TLV5616CP

Il Digital (to) Analog Converter (DAC), in italiano Convertitore Digitale-Analogico, è un componente elettronico in grado di produrre una determinata differenza di potenziale in funzione di un valore numerico che viene caricato.

L'integrato utilizzato nel circuito è il TLV5616CP, un DAC a 12-bit con una interfaccia SPI, alimentato nel range (2.7-5.5)V. La codifica di questo integrato è binaria diretta e la tensione in uscita è data dalla seguente relazione: $2 \times V_{REF} \times (\text{CODE} / 2^n)$ Volt, dove V_{REF} è la tensione di riferimento, Code è il valore in ingresso digitale, che può variare da 0 a $2^n - 1$, in binario [2].

Il TLV5616CP è dotato di 8 pin come mostrato in figura; per la realizzazione del circuito in oggetto sono serviti tutti i pin.

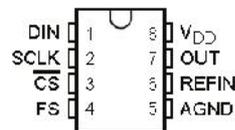


Figura 2.10 - Piedinatura dell'integrato TLV5616CP

- DIN = dati in ingresso digitali seriali;
- SCLK = segnale di clock;
- CS = chip select;
- FS = frame select;
- AGND = massa;
- REFIN = tensione analogica di riferimento in ingresso;
- OUT = uscita analogica;
- VDD = tensione di alimentazione (+5V)[8].

2.2.3 L'ADC TLC2543IN

L'Analog to Digital Converter (ADC, è un circuito elettronico in grado di convertire una grandezza continua (ad es. una tensione) in una serie di valori discreti.

L'ADC utilizzato nella realizzazione del circuito è il TLC2543IN; esso ha una risoluzione di 12-bit con interfaccia SPI. In aggiunta al convertitore ad alta velocità il dispositivo ha nel chip, un multiplexer a 14 canali che gli consente di selezionare uno degli ingressi o una delle tre tensioni all'interno del test-self [7].

Il TLC2543IN è dotato di 20 pin come mostrato in figura.

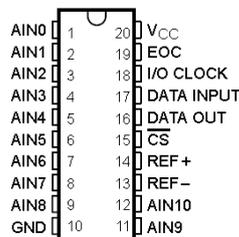


Figura 2.11 - Piedinatura dell'integrato TLC2543IN

- AIN0, ..., AIN10 = ingressi analogici i quali vengono “multiplexati” all'interno;
- GND = massa;
- REF- = tensione di riferimento negativa;
- REF+ = tensione di riferimento positiva;
- $\overline{\text{CS}}$ = chip select;
- DATA OUT = dati in uscita seriali;
- DATA INPUT = dati in ingresso seriali;
- I/O CLOCK = input/output clock;
- EOC = fine conversione;

VCC = tensione di alimentazione (+5V)[8].

2.2.4 Il Port Expander MCP23S17

L'integrato MCP23S17, prodotto dalla Microchip, è un Port Expander bidirezionale a 16-bit I/O, con interfaccia seriale (SPI), utilizzato per realizzare il circuito di switch. Questi 16-bit sono divisi in due gruppi da 8-bit associati rispettivamente a due registri

corrispondenti a due porte: PORT-A e PORT-B, in modo tale da poter configurare l'integrato in modalità 8-bit o 16-bit, dei quali, inoltre, è possibile configurare singolarmente i bit come ingressi oppure uscite semplicemente scrivendo nei registri IODIRA e IODIRB, di cui è dotato.

Grazie al protocollo SPI, il Master (UM245R) ha la possibilità di estendere i suoi 8-bit i/o ai 16-bit i/o del Port Expander, favorendo in tal modo il pilotaggio di un maggior numero di Slave. Il Port Expander MCP23S17 è dotato di 28 pin come mostrato in figura.

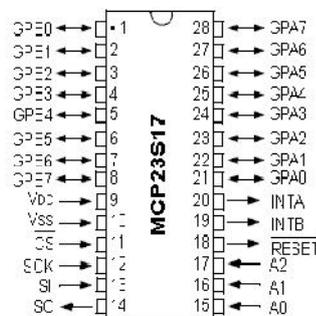


Figura 2.12 - Piedinatura dell'integrato MCP23S17

- GPB0, ..., GPB7 = pin I/O bidirezionali;
- VDD = tensione di alimentazione;
- VSS = massa;
- CS = chip select;
- SCK = clock;
- SI = dati in ingresso seriali;
- SO = dati in uscita seriali;
- A0, A1, A2 = pin di indirizzo hardware;
- RESET = reset hardware;
- INTB = interruzione dati in uscita per il PORT-B;
- INTA = interruzione dati in uscita per il PORT-A;
- GPA0, ..., GPA7 = pin I/O bidirezionali[8].

2.2.5 Il Transistor Array ULN2003AN

Il Transistor Array ULN2003AN consente di avere in uscita una tensione e una corrente elevata (rispetto agli standard delle porte logiche), grazie all'utilizzo di un array di transistori Darlington. Poiché consente di trasformare le uscite digitali del Port Expander (associate a 0V e 5V), in uscite (ovvero tensioni e correnti) compatibili per i relè, è stato utilizzato nella realizzazione del circuito per il pilotaggio di tre relè necessari per realizzare un doppio multiplexer 4:1.

Il Transistor Array ULN2003AN è dotato di 16 pin come mostrato in figura.

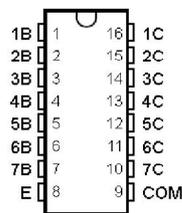


Figura 2.13 - Piedinatura dell'integrato ULN2003AN

- 1B, ..., 7B = ingressi;
- E = massa;
- COM = comune dei diodi;
- 1C, ..., 7C = uscite[8].

2.2.6 Il relè MT2-C93432

Il relè è un interruttore pilotato da un segnale.

Nella realizzazione del circuito in oggetto sono stati inseriti tre relè MT2-C93432 (figura 2.14) per realizzare un doppio multiplexer 4:1, in cui ogni relè in base al segnale ricevuto in ingresso seleziona uno tra quattro esperimenti che possono essere collegati al circuito realizzato.

Il relè MT2-C93432 è dotato di 8 pin esterni come mostrato in figura 2.14, in particolare:

- Pin 1, Pin 16 = terminale della bobina;

- Pin 4, Pin 8 = COM (terminale comune del deviatore);
- Pin 6, Pin 11 = NC (normalmente chiuso);
- Pin 8, Pin 9 = NA (normalmente aperto)[8].

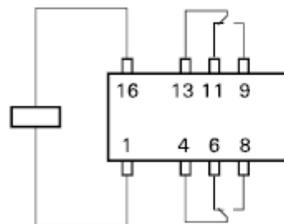


Figura 2.14 - Piedinatura dei relè MT2-C93432

2.2.7 Il Direct Digital Synthesis AD9834

Il sintetizzatore AD9834, prodotto dall' Analog Devices, è un circuito integrato DDS costituito da un accumulatore di fase a 28 bit, una SIN ROM, un DAC e un comparatore. Esso è un IC a bassa potenza con frequenza massima di clock di 50MHz capace di generare uscite sinusoidali, triangolari e, essendo dotato di un comparatore interno, anche onde quadre. I due registri interni di frequenza di 28 bit permettono di ottenere una risoluzione di 0.2Hz con un clock rate di 50MHz , e di 0.004Hz con un clock rate di 1MHz . Il DDS riceve i dati tramite il bus SPI che opera ad un clock rate fino a 40MHz , ed è alimentato nel range $(2.7-5.5)\text{V}$.

L' AD9834 è dotato di 20 pin come mostrato in figura 2.15:

- FSADJUST = Full-Scale Adjust Control;
- REFOUT = Voltage Reference Output, tensione di riferimento interna di (1.20V) resa disponibile a tale pin;
- COMP = DAC Bias Pin, per disaccoppiare la tensione di polarizzazione del DAC;

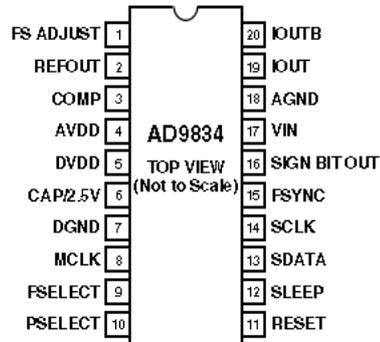


Figura 2.15 - Piedinatura dell'integrato AD9834

- AVDD = tensione di alimentazione per la sezione analogica;
- DVDD = tensione di alimentazione per la sezione digitale;
- CAP/2.5V = ulteriore tensione di alimentazione per la sezione digitale;
- DGND = massa per la sezione digitale;
- MCLK = Digital Clock Input, le frequenze d'output sono espresse come una frazione binaria della frequenza di MCLK;
- FSELECT = Frequency Select Input, per controllare quale registro di frequenza è usato nell'accumulatore di fase;
- PSELECT = Phase Select Input, per controllare quale registro di fase, è addizionato all'output dell'accumulatore di fase;
- RESET = Active High Digital Input, per resettare gli appropriati registri interni a "0";
- SLEEP = Active High Digital Input, quando questo pin è "1", il DAC si spegne perché non usato;
- SDATA = dati in ingresso seriali;
- SCLK = Serial Clock Input;
- FSYNC = Active Low Control Input, segnale di sincronizzazione della frame per i dati in ingresso; quando è "0", la logica interna è informata che una nuova parola è stata caricata nell'integrato;
- SIGNBITOUT = uscita del comparatore;
- VIN = ingresso del comparatore;

- AGND = massa per la sezione analogica;
- IOUT, IOUTB = Current Output, uscite complementari del DDS, infatti le due uscite sono in controfase, in tal modo facendone la differenza la componente continua si elimina[8].

2.2.8 Il DAC MCP4922

Il DAC utilizzato nella realizzazione della parte relativa al generatore di segnale è l'MCP4922: esso è alimentato nel range 2.7V-5.5V ed è un DAC a 12-bit con un'interfaccia SPI; la tensione di output ideale è data da una relazione lineare che la lega al codice scritto [12].

L'MCP4922 è costituito da due DAC, DAC-A e DAC-B: è stato utilizzato uno dei canali del DAC per ottenere la tensione di controllo dell'ampiezza del segnale generato e l'altro per ottenere la tensione di controllo dell'offset. Queste due tensioni vengono elaborate e passate al moltiplicatore analogico che realizza effettivamente la regolazione dell'ampiezza e dell'offset del segnale generato.

L'MCP4922 è dotato di 14 pin come mostrato in figura 2.16:

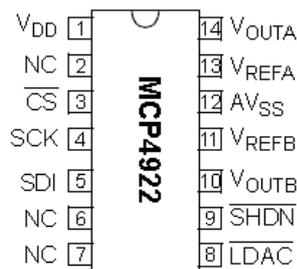


Figura 2.16 – Piedinatura dell'integrato MCP4922

- VDD = tensione di alimentazione;
- NC = non connesso;
- CS = Chip Select;
- SCK = clock;
- SDI = dati in ingresso seriali;

- **LDAC** = Latch DAC Input, per aggiornare contemporaneamente i due DAC;
- **SHDN** = Hardware Shutdown Input, per spegnere il DAC quando non usato;
- **VOUTB** = uscita del DAC-B;
- **VREFB** = tensione di riferimento per il DAC-B;
- **AVSS** = massa;
- **VREFA** = tensione di riferimento per il DAC-A;
- **VOUTA** = uscita del DAC-A[8].

2.2.9 L'operazionale sottrattore TL082CP

L'integrato TL082CP contiene due amplificatori operazionali a JFET dal basso costo e buona velocità. Esso è stato utilizzato per realizzare un circuito sottrattore, infatti, lo scopo è quello di traslare lo swing di uscita del DAC-B, che gestisce l'offset, verso il basso di un valore pari proprio a VREFB in modo da ottenere uno swing dell'offset simmetrico rispetto a 0V. Nella fattispecie VREF è impostata a un valore pari a 1.024V (per ragioni legate alla codifica binaria della tensione) grazie ad un regolatore di tensione con regolazione d'uscita, mentre l'output del DAC-B varia fra 0V e 2V, per ottenere la differenza fra queste due tensioni è sufficiente costruire il circuito in figura[8].

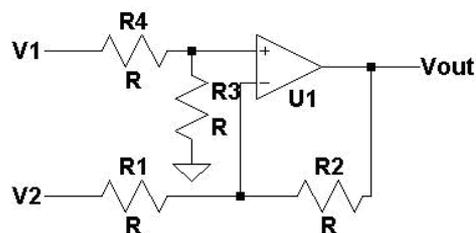


Figura 2.17 - Circuito sottrattore

Il TL082CP è dotato di 8 pin come mostrato in figura 2.18.

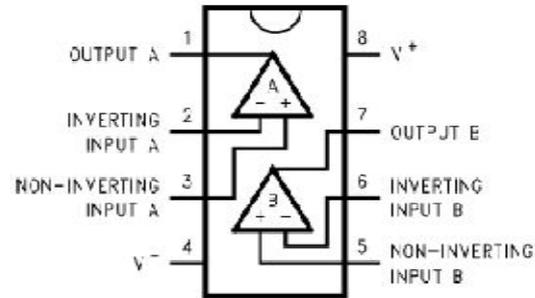


Figura 2.18 - Piedinatura dell'integrato TL082CP

2.2.10 Il moltiplicatore AD835AN

L'AD835AN è un moltiplicatore analogico con uscita in tensione a quattro quadranti. Si tratta di un circuito in grado di moltiplicare istante per istante il valore di due tensioni. Nella sua modalità di base, fornisce il prodotto lineare delle sue tensioni d'input X e Y con una larghezza di banda di uscita a $-3dB$ di $250MHz$. Gli input differenziali di moltiplicazione (X, Y) e l'input sommatore (Z) sono ad alta impedenza, consentendo di non essere influenzati dai circuiti a monte. La tensione di uscita a bassa impedenza consente di non essere influenzata dai circuiti a valle, può fornire fino a $\pm 2.5V$ e può pilotare carichi fino a 25Ω .

L'AD835AN è dotato di 8 pin come mostrato in figura:

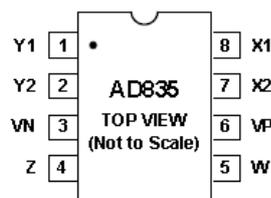


Figura 2.19 - Piedinatura dell'integrato AD835AN

- Y1 = tensione d'input;
- Y2 = tensione d'input;
- VN = tensione di alimentazione negativa;
- Z = tensione d'input;
- W = tensione d'output;
- VP = tensione di alimentazione positiva;

- X2 = tensione d'input;
- X1 = tensione d'input[8].

2.2.11 L'amplificatore di tensione LM6172IN

L'integrato LM6172IN è un doppio amplificatore operazionale ad alta velocità.

E' stato utilizzato per realizzare un blocco amplificatore a doppio stadio costituito da due configurazioni invertenti in cascata. Un singolo stadio è costituito da un operazionale in configurazione invertente, come mostrato in figura 2.20, in cui il guadagno di tensione è dato essenzialmente da:

$$V_{out} = - V_{in} \cdot (R_f/R_{in})$$

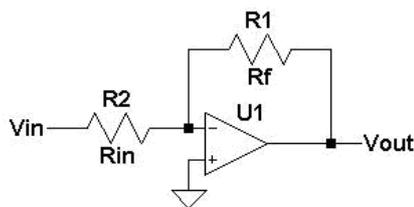


Figura 2.20 - Amplificatore di tensione invertente

Dal momento che si vuole ottenere un'amplificazione di tensione totale pari a 10V/V (20dB) è sufficiente replicare la struttura appena vista e collegare i due stadi in cascata. La piedinatura di questo integrato è identica a quella del TL082CP (figura 2.18)[8].

2.2.12 L'amplificatore di corrente THS6012I

L'integrato THS6012I contiene due driver ad alta velocità capaci di fornire 400mA di corrente d'uscita in un carico da 25Ω. Tali driver possono essere configurati in maniera differenziale in modo da pilotare un segnale da 50V_{pp} su di una linea a bassa impedenza. I driver sono amplificatori di corrente progettati per gli alti slew-rate necessari per supportare una bassa distorsione armonica totale nelle applicazioni xDSL. Il THS6012I è dotato di 20 pin come mostrato in figura 2.21.

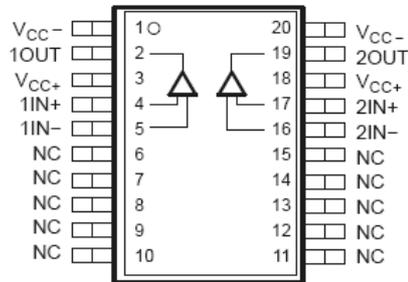


Figura 2.21 - Piedinatura dell'integrato THS6012I

- V_{CC-} = tensione di alimentazione negativa;
- 1OUT = uscita del primo driver;
- V_{CC+} = tensione di alimentazione positiva;
- 1IN+ = ingresso differenziale positivo del primo driver;
- 1IN- = ingresso differenziale negativo del primo driver;
- NC = terminale per la dissipazione termica;
- 2IN- = ingresso differenziale negativo del secondo driver;
- 2IN+ = ingresso differenziale positivo del secondo driver;
- 2OUT = uscita del secondo driver.

L'integrato è stato montato, come in figura 2.22, con i due driver in parallelo configurati come buffer di tensione in quanto non c'è necessità di ulteriori amplificazioni di tensione, le resistenze sono state scelte in base ai suggerimenti forniti dal datasheet[8].

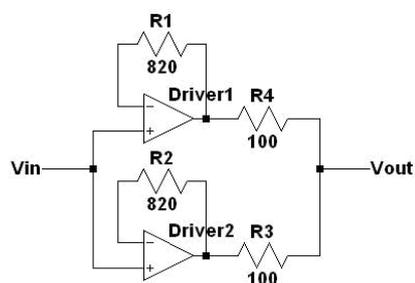


Figura 2.22 - Configurazione dell'integrato

Il circuito ottenuto connettendo i vari integrati, è mostrato in figura 2.23.

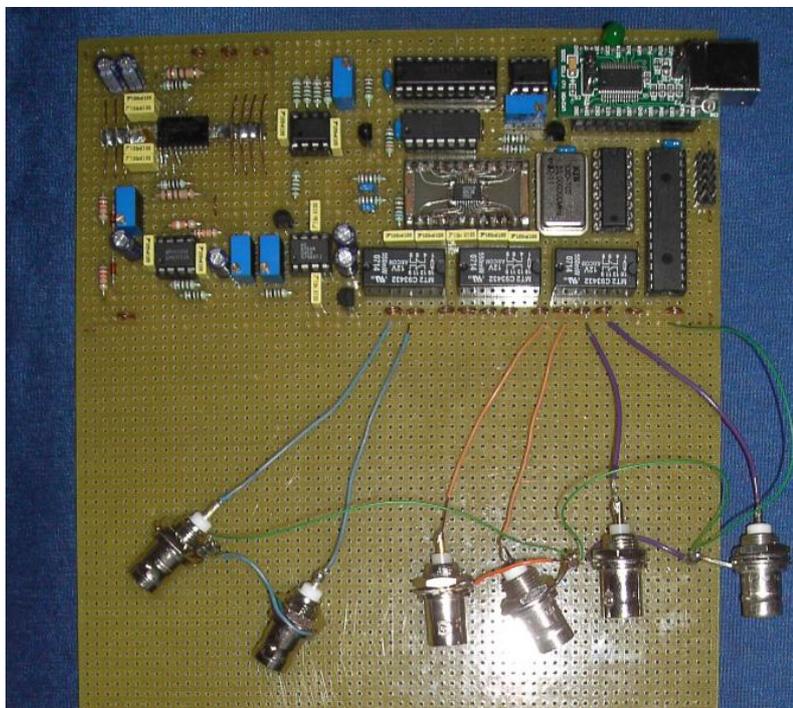


Figura 2.23 - Prototipo realizzato su basetta millefori

2.3 La scheda d'acquisizione veloce PCI-5102²

Fa parte delle schede di acquisizione “High speed digitizers”. In particolare, è in grado di acquisire ad una frequenza di campionamento massima pari a 20Mcampioni/sec con risoluzione a 8 bit da 2 canali ed una banda passante analogica di 15MHz, permettendo quindi di realizzare strumenti in grado di lavorare fino ad alcuni MHz senza grandi problemi. Inoltre, è dotata di una memoria Buffer di tipo FiFo, in cui vengono temporaneamente immagazzinati 663Kcampioni. Presenta una tensione nominale d'ingresso di $\pm 5V$, è dotata di un front-end analogico di amplificazione a guadagno programmabile che permette di adattare la sensibilità per ogni canale tra un minimo di $\pm 50mV$ ad un massimo di $\pm 5 V$. Con i driver della scheda NI vengono forniti una

² Esiste anche la versione con interfaccia PCMCIA per l'installazione su PC portatili, necessaria per poter utilizzare gli esperimenti in aula.

applicazione di test che realizza un semplice oscilloscopio ed una serie di programmi d'esempio sul funzionamento della scheda e sui passi di programma necessari per dialogarvi. Sebbene il costo di questa scheda sia eccessivo, presenta il vantaggio di essere molto veloce e tramite i suoi driver è stato possibile realizzare il pannello frontale di un oscilloscopio digitale [4].



Product	Bus	Channels	Sampling Rate	Bandwidth	Memory Total	Resolution
NI 5102	PCI, PXI, PCMCIA, USB, ISA	2	20 MS/s	15 MHz	663 kS to 16 MS	8 bits for PCI/PXI

Figura 2.24 - NI PCI-5102

2.4 Gli esperimenti

2.4.1 Circuito caotico RLD

Il circuito RLD, mostrato in figura 2.25, è composto dalla serie di una induttanza, una resistenza ed un diodo ed è alimentato da un generatore sinusoidale ad ampiezza e frequenza variabile. Il solo componente non lineare è il diodo mentre resistenza ed induttanza sono da considerarsi funzionanti in regione lineare (si considera l' induttanza funzionante non in saturazione).

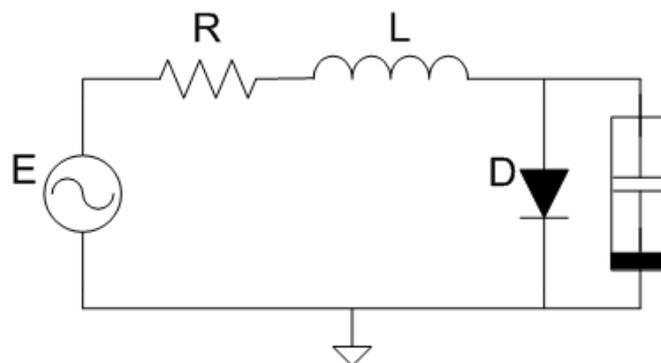


Figura 2.25 - Circuito caotico RLD

Del diodo è certamente importante la capacità inversa infatti i fenomeni più significativi si notano ad una frequenza del generatore prossima a quella della risonanza serie della induttanza e della capacità del diodo. Per questo motivo è necessario scegliere o un diodo di potenza lento o un varicap, non vanno invece bene i diodi di commutazione di piccolo segnale o i diodi veloci perché hanno capacità inversa e tempo di recupero inverso troppo piccolo costringendo il circuito a lavorare a frequenze troppo elevate per avere fenomeni significativi.

La selezione del diodo, dei valori della induttanza e della resistenza e della frequenza di lavoro del generatore è stato oggetto di una tesina precedentemente svolta da altri studenti nell'ambito dell'esame di teoria dei circuiti. La linea guida che si è seguita principalmente è stata di trovare una frequenza di lavoro accettabile (non troppo elevata) per la strumentazione a disposizione.

Per far ciò si è selezionato un diodo, tra le tantissime scelte possibili, che presentasse una capacità non piccola, per questo motivo la scelta è caduta su un diodo di potenza invece che su un diodo di segnale.

La resistenza serie si è scelta relativamente piccola per non abbassare il fattore di merito del circuito risonante, si ricordi infatti che per piccoli valori della tensione il circuito può, in linea di principio, essere linearizzato ottenendo proprio un circuito R-L-C serie. La presenza di questa resistenza ha anche l'utile effetto collaterale di permettere la misura della corrente che passa nel circuito. La frequenza di funzionamento ottimale è stata cercata per tentativi, fino a trovarne una che presentasse un ampio spettro di dinamiche (soluzione fondamentale, sub-armoniche, caos, finestre periodiche nel caos). E' comunque interessante studiare il circuito RLD facendo variare sia la frequenza che l'ampiezza del segnale di pilotaggio, si nota così che al variare della frequenza non solo le ampiezze a cui si sviluppano i vari fenomeni cambiano, ma anche che non sempre i fenomeni caotici sono presenti [4].

2.4.2 Circuito caotico di Chua

Il circuito di Chua deriva dagli studi sul caos del prof. Leon O. Chua, docente dell'università della California, Berkeley, ed è l'unico circuito in cui la presenza del

caos è stata provata in maniera analitica. Il vantaggio fondamentale del circuito di Chua è quello di essere un circuito autonomo, cioè di non aver bisogno di un segnale in ingresso.

Questo circuito, che fa parte della famiglia degli oscillatori caotici, presenta i tre requisiti minimi necessari per poter avere comportamenti caotici in circuiti autonomi:

- Dinamica almeno del terzo ordine, quindi almeno tre componenti dinamici indipendenti;
- Almeno un componente non lineare;
- Almeno un componente attivo.

Queste sono condizioni necessarie ma non sufficienti perché un sistema possa generare un comportamento caotico inteso come comportamento aperiodico, duraturo nel tempo, delle traiettorie di un sistema deterministico.

In tal caso, a causa della dipendenza sensibile dalle condizioni iniziali, si possono avere traiettorie che non raggiungono punti di equilibrio e non si chiudono su cicli limite, ma continuano a muoversi nello spazio di stato presentando oscillazioni non periodiche non determinabili a priori.

In un circuito autonomo, come quello in esame, questo comportamento non è dovuto a fattori forzanti esterni ma è una proprietà intrinseca del sistema caotico.

Il circuito di Chua, in figura 2.26, contiene tre elementi di accumulazione di energia che sono due condensatori passivi lineari e un induttore passivo lineare, un resistore passivo lineare e un resistore non lineare a due terminali detto anche “Diodo di Chua”.

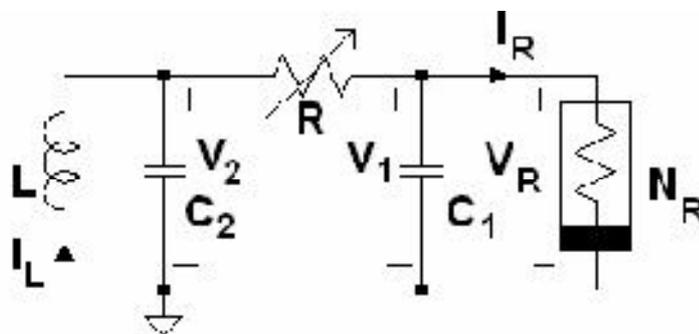


Figura 2.26 - Circuito di Chua

Il sistema presenta diversi gradi di libertà, vale a dire che esiste la possibilità di agire sul valore delle due capacità, sul valore dell'induttore e sul valore del resistore variabile per osservarne le dinamiche di funzionamento. Si potrebbe pensare di variare il valore di una delle due capacità ma per esigenze costruttive queste variazioni non sono di grande praticità. Per quanto riguarda l'induttore è possibile utilizzare un induttore variabile ma, essendo vincolati nell'intorno di un determinato valore, per ottenere le migliori condizioni di funzionamento del circuito, si utilizza come grado di libertà il valore del resistore variabile.

Dal momento che la resistenza R , l'induttanza L e le capacità C_1 e C_2 sono valori positivi, è chiaro che questo circuito per oscillare, e tanto più diventare caotico, dovrà presentare un resistore non lineare attivo, nel senso che la sua caratteristica tensione-corrente deve esibire regioni (secondo e quarto quadrante) in cui il prodotto $V \cdot I$ è negativo, quindi fornire energia agli elementi passivi.

Nel laboratorio di circuiti è stato realizzato in passato un circuito di Chua cui schema elettrico è rappresentato in figura.

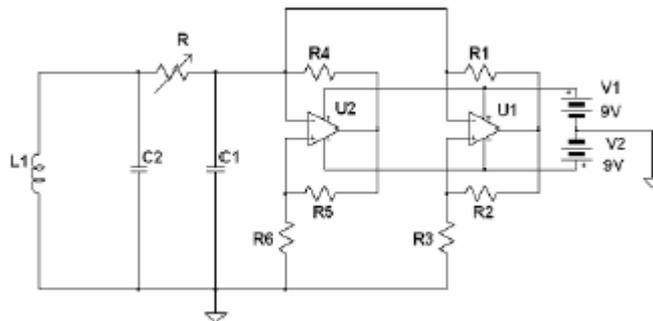


Figura 2.27 - Schema elettrico del circuito di Chua

La resistenza non lineare è un componente che presenta resistenza negativa e una non linearità. Dato che si opera con valori di frequenza non superiori ai 24kHz, è stato inserito come elemento attivo un generico amplificatore operazionale (un TL08235), che connesso ad una opportuna rete di resistenze consente di giungere alla realizzazione del resistore non lineare (o diodo di Chua), mostrato in figura 2.28.

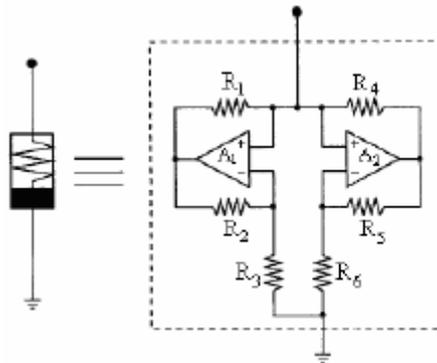


Figura 2.28 - Resistore non lineare realizzato con TL082

Questo resistore occupa un ruolo fondamentale nel circuito perché grazie ad esso è possibile osservare la varietà di biforcazioni e di andamenti caotici che il circuito di Chua genera.

Come induttore è stato poi realizzato un “induttore virtuale” con un amplificatore operazionale (TL082), una capacità e da un insieme di resistenze, opportunamente connesse, come si può osservare in figura.

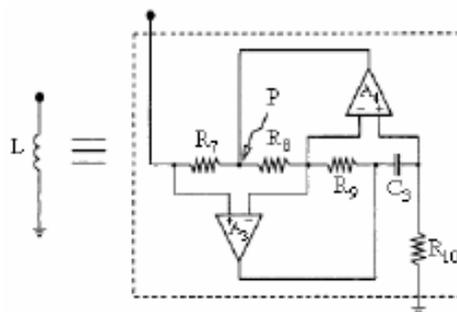


Figura 2.29 - Layout dell'induttore virtuale

Questa implementazione dell'induttore migliora le prestazioni generali del circuito di Chua, in termini di visualizzazione a schermo delle dinamiche e aumento del range di variabilità della resistenza lineare.

Per visualizzare i modi di funzionamento del circuito di Chua è stato realizzato un resistore a controllo digitale inserito poi nel circuito stesso al posto del resistore lineare variabile; esso è mostrato in figura 2.30.



Figura 2.30 - Resistore a controllo digitale

Il resistore a controllo digitale presenta da un lato due terminali, che rappresentano le due estremità della rete resistiva, mentre dall'altro si trova un cavo che consente la connessione al circuito di switch.

L'interfaccia grafica, realizzata con labview, permette di variare il valore resistivo in uscita dal resistore digitale ed il relativo stato di funzionamento del circuito di Chua, collegando due sonde ai capi dei condensatori C1 e C2 si è in grado di visualizzare con l'oscilloscopio i modi di funzionamento, dal ciclo stabile alle biforcazioni, dall'attrattore strano fino al caos [2].

Capitolo 3

REALIZZAZIONE DEL SOFTWARE DI CONTROLLO, PER IL LABORATORIO DI CIRCUITI

In questo capitolo verrà descritto nel dettaglio il programma di controllo necessario al funzionamento dell'hardware a disposizione e tutti gli aspetti software, che consentono la fruizione dei servizi messi a disposizione dal "Laboratorio remoto di circuiti".

Verranno inizialmente descritte le caratteristiche fondamentali dell'ambiente di sviluppo LABVIEW ed in seguito il software di controllo realizzato.

3.1 Introduzione al Labview

3.1.1 Cos'è il Labview?

Il LabView (Laboratory Virtual Instrument Engineering Workbench) è un ambiente di sviluppo per applicazioni principalmente orientate:

- all' acquisizione di dati ed alla gestione di strumentazione elettronica;
- all'analisi ed elaborazione dei segnali.

Il suo contributo specifico allo sviluppo lo si deve al vasto numero di schede di acquisizione e sistemi di misura, dalle quali è usato come principale software attraverso cui interagiscono e vengono programmate[5].

LabView fornisce un ambiente di programmazione di tipo grafico ad oggetti denominato “G-Language”, il quale consente di realizzare programmi in forma di diagrammi a blocchi, in quanto utilizza icone invece di linee di testo per creare applicazioni.

LabView conserva comunque molte similitudini con gli ambienti di programmazione tradizionali: presenta tutti i tipi di dati e gli operatori predefiniti di uso comune, permette di generare nuovi tipi di dati combinando tra loro i tipi di dati elementari e di controllare l'esecuzione dei programmi ricorrendo a strutture di controllo di flusso come ad esempio cicli e costrutti per l'esecuzione condizionale di codice. Questo ambiente di sviluppo software presenta anche delle peculiarità che lo differenziano notevolmente dai linguaggi procedurali più comunemente noti. La prima differenza sostanziale è l'ambiente di sviluppo grafico, sia dell'interfaccia utente sia dell'algoritmo di elaborazione; altra differenza importante è che un programma LabView non esegue il flusso delle istruzioni, ma bensì, segue il flusso di dati. Infatti i programmi scritti in LabView tendono ad essere di tipo data-driven, nel senso che si tende ad enfatizzare come i dati si muovono tra i diversi blocchi operativi più che la sequenza delle istruzioni da eseguire. Mette inoltre a disposizione del programmatore una serie di librerie di funzioni che possono essere richiamate ed utilizzate all'interno dei programmi: le librerie comprendono funzioni di uso comune (funzioni aritmetiche e statistiche, funzioni per la manipolazione di stringhe, ecc.) ed inoltre funzioni specializzate per l'acquisizione e l'elaborazione dei segnali, il controllo di strumentazione numerica via interfaccia IEEE-488 o VXI, la trasmissione di dati mediante l'uso di porte speciali oppure mediante il protocollo di comunicazione TCP/IP. E' possibile inoltre definire nuove funzioni ed arricchire le librerie in dotazione con LabView.

Infine il programma consente di effettuare il debug delle applicazioni create in linguaggio G attraverso opportune modalità di esecuzione dei programmi, come ad esempio il modo “highlight execution” o “single step” e per mezzo di oggetti che consentono in run-time la modifica di variabili di programma.

LabView presenta alcuni vantaggi rispetto ad un linguaggio di programmazione

tradizionale:

- E' di facile apprendimento, in quanto presenta una modalità di programmazione a blocchi, di tipo visuale ed intuitivo;
- Permette di dare al codice una struttura modulare che consente di suddividere programmi complessi in sottoprogrammi più semplici che possono essere riutilizzati;
- Consente di raccogliere i VI in librerie, ovvero in un insieme di sub-vi utilizzabili da altri VI e velocemente inseribili nel codice sorgente dal programmatore;
- Fornisce un considerevole insieme di librerie per lo sviluppo di applicativi, tra le quali si trovano funzioni di tipo matematico e statistico, controllo di dispositivi per mezzo di alcuni tipi di interfaccia, comunicazione tra calcolatori, ecc.

3.1.2 Utilizzo principale di LabView

LabView è stato pensato principalmente per il controllo di schede di espansione connesse direttamente al bus di un calcolatore o di strumentazione connessa al calcolatore stesso attraverso opportune interfacce come il bus IEEE 488, RS-232, strumenti VXI o ancora attraverso Internet mediante il protocollo TCP/IP.

L' ambiente di sviluppo consente di costruire programmi i quali prendono il nome di **strumenti virtuali** (Virtual Instrument, **VI**).

Un Virtual Instrument permette l' interazione tra calcolatore e strumentazione fornendo contemporaneamente all' utente un opportuno pannello frontale grafico per il dialogo con il VI stesso. In questo modo l'utente interagisce con un nuovo dispositivo (Instrument), costituito da calcolatore, interfacce, strumenti e programma il quale presenta una realtà (Virtual) diversa dai singoli oggetti fisici che compongono il sistema stesso. Tale fatto spiega il nome di Virtual Instrument dato ad un programma LabView[5].

3.1.3 Virtual Instrument (VI)

Si ricorda che i programmi che si possono realizzare utilizzando il linguaggio grafico LabView sono chiamati **Virtual Instrument (VI)**, dove il termine “strumenti” è dovuto al fatto che durante l’esecuzione i programmi sviluppati presentano agli utilizzatori una interfaccia analoga a quella di uno strumento di misura, mentre il termine “virtuali” è dovuto al fatto che l’interazione avviene con un programma in esecuzione e non con un dispositivo fisico dedicato. L’utente può modificare il valore di alcune grandezze agendo su opportune manopole o interruttori visualizzati dal programma e può osservare il risultato delle elaborazioni condotte internamente al VI su display grafici molto simili a quelli che si trovano sulla strumentazione numerica.

Un VI è composto da tre parti fondamentali:

- Pannello frontale (Front Panel);
- Diagramma a blocchi funzionale (Block diagram);
- Icona/connettore (Icon/connector).

Il **Front Panel** (pannello frontale) è la finestra che rappresenta l’interfaccia tra il programma e l’utente: il nome deriva dal fatto che può essere strutturato in modo tale da ricordare il pannello frontale dotato di display, indicatori, manopole, tasti, ecc.. Nel pannello frontale vengono allocati i *controllori* gli *indicatori* dello strumento virtuale. Il *controllore* è una variabile di ingresso, che può essere modificata agendo sul pannello frontale, tramite la *tools palette*: è un pannello che mette a disposizione una serie di strumenti che consentono di modificare il valore di un controllore o di selezionare il testo interno; consente di selezionare, posizionare, ridimensionare oggetti; consente di editare il testo e di creare etichette; consente di collegare i terminali dei nodi (ossia controllori con VI, indicatori con VI, ecc.) presenti nel block diagram; apre il menù di pop-up di un oggetto; ecc. Per *indicatore* si intende una variabile di uscita il cui valore può essere modificato dal programma e non dall’utente.

E’ possibile interagire con un front panel mediante tastiera o mouse, introducendo valori numerici o stringhe di caratteri, modificare lo stato di elementi grafici, come ad esempio

manopole, bottoni e così via.

Il **Block Diagram** (diagramma a blocchi funzionale) contiene il codice nella forma di diagramma a blocchi ed è costituito da: *Nodi*, sono degli elementi di elaborazione; *Collegamenti*, uniscono i nodi e permettono lo scambio di informazioni. Le informazioni passano da un nodo all'altro del pannello frontale per mezzo dei connettori, che li uniscono. Pur presentandosi in forma grafica diversa, il diagramma a blocchi presenta possibilità di programmazione analoghe a quelle offerte da un comune linguaggio di programmazione del tipo text-based. La coppia **Icon/connector** (icona/connettore) è il terzo elemento fondamentale di un programma LabView e consente di trasformare un programma in un oggetto. L' icona è un simbolo grafico di piccole dimensioni , che rappresenta sinteticamente il VI stesso: essa è normalmente visibile in alto a destra della finestra front panel e della finestra block diagram. Il connettore consente di associare, nell' ambito del pannello frontale alle aree dell' icona, i controllori e indicatori che caratterizzano il funzionamento del VI [5].

3.1.4 Sub-VI

Un **Sub-VI** è un VI, utilizzato all' interno di un altro VI come sottoprogramma. Qualunque VI, una volta creato, può essere utilizzato come sub-VI all' interno di un altro VI di più alto livello. Di conseguenza un Sub-VI è analogo ad una subroutine in un linguaggio di programmazione di tipo text-based. Come avviene in altri linguaggi con le subroutine, non c'è alcun limite al numero di Sub-VI inseribili all' interno di un programma in linguaggio G. E' inoltre possibile chiamare un Sub-VI dall' interno di un altro Sub-VI.

Quando un Sub-VI, viene inserito all' interno di un VI, presenta l' icona ad esso associata nel block diagram che lo contiene.

Quando impiegato come SubVI, un VI normalmente non mostra a video il proprio pannello frontale, ma si limita a ricevere i dati di ingresso per mezzo dei collegamenti tra il sub-VI stesso ed il resto del programma che lo contiene ad elaborarli e a fornire le eventuali uscite[5].

3.2 Descrizione del software di controllo per il prototipo

Di seguito verrà descritto il software, realizzato in ambiente Labview 7.1, per il controllo del prototipo in oggetto e necessario a rendere operativi tutti gli apparati del laboratorio didattico sui circuiti; tale software è il risultato di uno sviluppo portato avanti col tempo, grazie anche al contributo di laureandi degli anni passati.

La volontà di mettere assieme l'hardware a disposizione, per creare un laboratorio remoto ed in aula di circuiti, necessitava di un software in grado di controllare attraverso un'unica interfaccia, l'intero complesso di apparati di cui è costituito; obiettivo del lavoro svolto è stato dunque quello di mettere a punto il software preesistente e crearne di nuovo per riuscire ad ottenere lo scopo prefissato.

Con questo software, unito al circuito d'acquisizione integrato nella scheda prototipo, è possibile selezionare l'esperimento che si desidera eseguire tra quelli disponibili (Chua o RLD), utilizzare il generatore di segnali anch'esso implementato nel prototipo, o applicare una tensione in uscita al DAC del circuito di acquisizione. Il software realizzato verrà dapprima descritto in linea generale, mentre più avanti, verranno illustrati nel dettaglio i singoli sub-VI che lo costituiscono.

3.2.1 Software di controllo: le diverse versioni

Il software di controllo del circuito è stato realizzato in due versioni a seconda che lo si voglia utilizzare da remoto oppure in laboratorio/aula, questo sia per aspetti propri del software, che verranno approfonditi più avanti, sia per una questione di funzioni da mettere a disposizione dei "diversi utenti". Alcune funzioni infatti, come ad esempio il controllo del generatore di segnali, possono risultare superflui per un utilizzo da remoto e quindi non ha senso implementarle in una versione "remota" del software. Al contrario per un utilizzo in laboratorio, o in aula, l'utente può avere la necessità di utilizzare tutte le funzionalità implementate nel circuito e quindi verrà fornito un software più completo, in grado altresì di realizzarle.

3.2.2 Versione del software per l'utilizzo in aula o in laboratorio

Questa versione è stata pensata per l'utilizzo specifico in aula o in laboratorio, essa infatti mette a disposizione dell'utente tutte le funzionalità implementate dall'hardware a disposizione per la didattica sperimentale, ovvero: controllo del circuito di Chua, del circuito RLD, del generatore di segnali e scrittura nel DAC del circuito di acquisizione.

Il programma principale che dà accesso ai vari esperimenti e funzioni si chiama `mainprogram.vi`, esso presenta varie finestre di dialogo strutturate "ad albero", cioè su più livelli, che guidano l'utente alla scelta dell'attività desiderata; tale struttura è mostrata in figura.

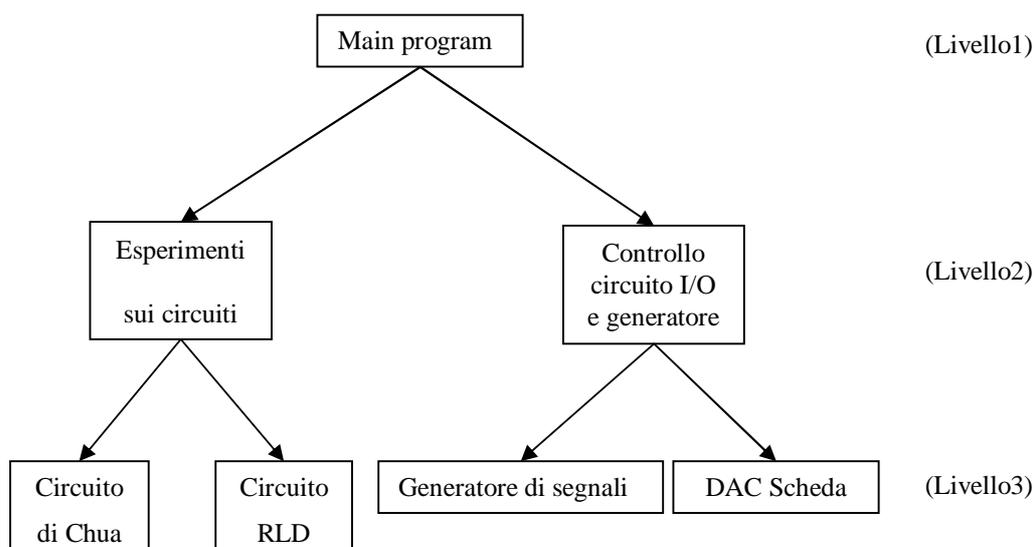


Figura 3.1 - Struttura ad albero di "mainprogram.vi"

L'interfaccia del programma "mainprogram.vi", realizzata in ambiente labview, è mostrata in figura 3.2



Figura 3.2 - Frontal panel di mainprogram.vi

- Il pulsante “*esperimenti sui circuiti*”, indirizza l’utente verso una maschera, sotto forma di finestra pop-up³, che permette di scegliere uno tra gli esperimenti a disposizione per la didattica sperimentale.
- Il pulsante “*Controllo circuito di I/O e generatore di segnali*”, indirizza invece l’utente verso un’altra maschera, anch’essa sotto forma di finestra pop-up, che consente di scegliere se controllare il generatore di segnali o il circuito d’acquisizione integrati nella scheda prototipo.
- Il pulsante “*stop*” termina l’esecuzione del programma.

Le maschere che appaiono premendo o il primo o il secondo pulsante, sono mostrate rispettivamente in figura 3.3 e 3.4:

³ Per ottenere questa possibilità è stata selezionata l’opzione “show frontal panel when called” dalle proprietà “customize windows appearance” del sub-VI di destinazione.



Figura 3.3 - Frontal panel di "Esperimenti sui circuiti.vi"



Figura 3.4 - Frontal panel di "scheda e generatore.vi"

Tornando al programma principale, “mainprogram.vi”, c’è da dire che esso oltre a gestire l’avvio degli esperimenti, esegue anche le inizializzazioni necessarie al corretto funzionamento degli apparati a nostra disposizione; in figura ne viene mostrato il block diagram:

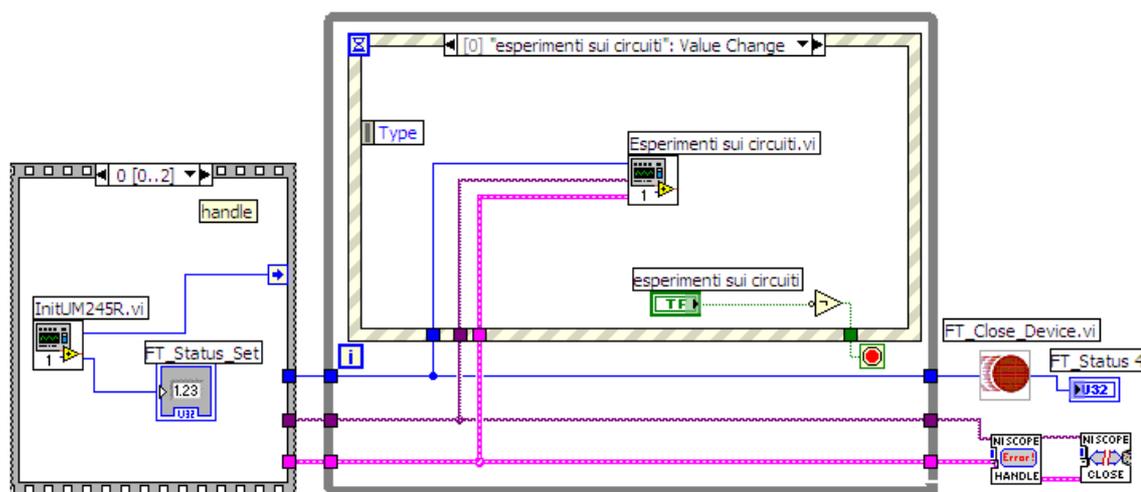


Figura 3.5 - Block diagram di “mainprogram.vi”

Procedendo da sinistra verso destra, abbiamo un primo blocco di codice costituito da una stacked sequence structure⁴ all’interno della quale sono stati inseriti nell’ordine i sub-VI per l’inizializzazione del modulo UM245R, dell’integrato MCP23S17 e della scheda d’acquisizione veloce PCI-5102. L’ordine di inizializzazione, deve necessariamente prevedere in testa quella del modulo UM245R poiché questo integrato, fungendo da master, deve poter pilotare tutti gli altri dispositivi slave.

L’inizializzazione del modulo UM245R è realizzata attraverso il sub-VI “InitUM245R.vi” che apre la comunicazione con il modulo USB e ritorna il valore del puntatore Handle, un indicatore che mostra una sorta di “indirizzo” identificativo del dispositivo collegato e che sarà usato per accessi successivi; ritorna inoltre il parametro FT_STATUS, un indicatore che fornisce lo stato del dispositivo[2].

⁴ La stacked sequence structure è una struttura che permette la sequenzializzazione dei sub-diagrammi contenuti in essa, o per meglio dire delle funzioni presenti al suo interno.

Concatenato a questo sub-VI, nella seconda frame, c'è il sottoprogramma "InitMCP23S17", esso riceve in ingresso l'handle e consente sia di settare i pin dell'integrato (come input oppure come output), sia di configurarne i registri interni; nel nostro caso per pilotare il circuito di Chua e controllare il multiplexer che seleziona gli esperimenti, è necessario impostare i pin della porta A e quelli della porta B del *port expander*, come pin di uscita.

Infine la stacked sequence structure presenta una terza frame, in cui è stato inserito il sub-VI di inizializzazione della scheda PCI-5102, fornito con la scheda stessa.

Di seguito sono mostrate le frames che eseguono le suddette operazioni:

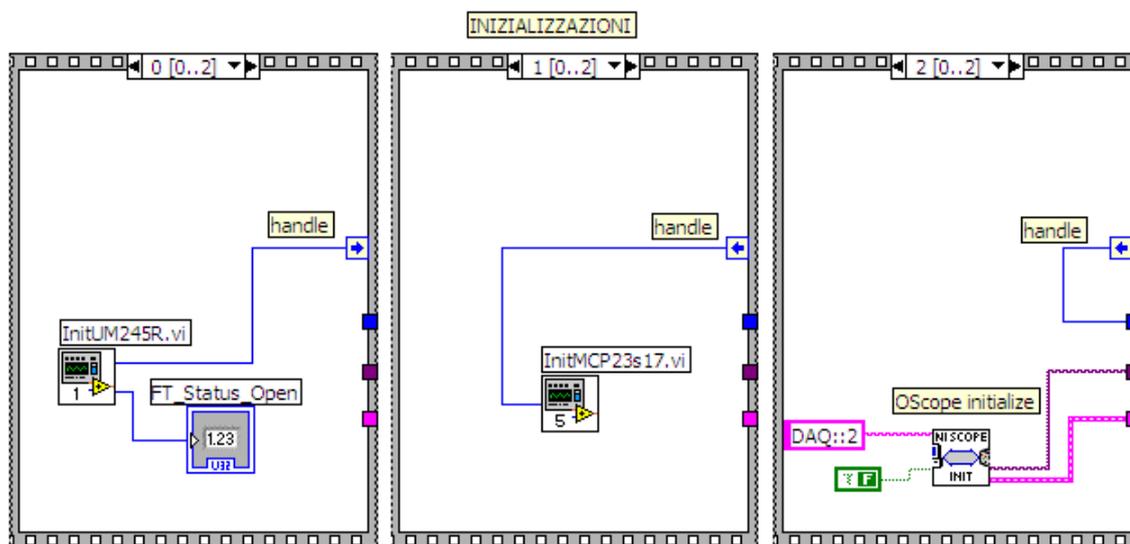


Figura 3.6 - Sequenza di inizializzazioni

Passiamo ora a descrivere il codice che realizza l'interfaccia del menu principale, mostrato in figura 3.5.

L'interfaccia utente può essere realizzata mediante un ciclo while che ad ogni iterazione, esegue una lettura su tutti i controlli del frontal panel per verificare se qualcuno risulta modificato rispetto all'ultima iterazione; questa verifica viene eseguita indipendentemente dal fatto che un controllo sia cambiato o meno e ciò fa intuire che questa tecnica non sfrutta in maniera efficiente le risorse del PC.

Per questo motivo, tale soluzione è stata scartata e la scelta è ricaduta su di una event structure.

Con la versione 6.1 di Labview è stata introdotta una nuova caratteristica che rende la programmazione di interfacce utente più semplice ed efficiente: la struttura ad eventi. Con questa struttura, l'applicazione attende che un evento si verifichi sul pannello frontale, piuttosto che dover ripetutamente leggere i valori delle variabili; essa si avvale della funzionalità del sistema operativo per essere informato quando l'utente ad esempio modifica un valore, sposta il mouse o chiude l'applicazione, in modo da riservare la CPU per altri programmi in esecuzione sul computer.

Quando LabVIEW esegue la event structure, si pone il VI in attesa fino a quando si verifica uno degli eventi preconfigurati; quando un evento di interesse si verifica, la struttura ad eventi automaticamente *si sveglia* ed esegue i sub-VI associati a tale evento. Nel nostro caso, è stata realizzata una event structure con tre frames, di cui due sono associate ai pulsanti “Esperimenti sui circuiti” e “Controllo generatore e scheda”, la terza, al pulsante “stop”. I dati risultanti dalle inizializzazioni, vengono *passati* all'interno della event structure e attraversano anche il ciclo while per garantire la sequenzialità delle operazioni.

La **prima frame** viene eseguita quando l'utente preme il pulsante “Esperimenti sui circuiti”; in questo caso, viene mandato in esecuzione il sub-VI “Esperimenti sui circuiti” e mostrata all'utente l'interfaccia di figura 3.3, con la quale è possibile selezionare uno tra gli esperimenti proposti, ed accedervi.

Del tutto analoga è la **seconda frame**, che viene eseguita invece quando l'utente preme il pulsante “Controllo circuito I/O e generatore di segnali”; in questo caso, viene mandato in esecuzione il sub-VI “scheda e generatore.vi” e mostrata all'utente l'interfaccia di figura 3.4.

La **terza frame** viene eseguita quando si preme il pulsante “stop”, il quale è stato connesso direttamente al nodo di terminazione del ciclo while; in questo caso il programma “esce” dal ciclo while, e chiude la comunicazione con la scheda di acquisizione veloce PCI-5102 e con il modulo USB UM245R.

E' da osservare che il software consente di eseguire un'attività, terminarla, e subito dopo eseguirne un'altra, senza dover necessariamente terminare il programma principale: ciò significa che le inizializzazioni vengono eseguite una sola volta (all'avvio del mainprogram); allo stesso modo il rilascio delle risorse, avviene solo al termine del programma principale. Questo accorgimento oltre ad essere valido da un punto di vista teorico, risulta anche efficace da un punto di vista della prestazioni, perché riduce i tempi di esecuzione del programma. In questa direzione è stata rivolta un'attività di update dei programmi già esistenti, per adattarli a questa soluzione.

Passiamo ora a descrivere il block diagram del sotto-programma “Esperimenti sui circuiti.vi”, posto all'interno di “mainprogram.vi” e il cui pannello frontale è già stato mostrato in figura 3.3.

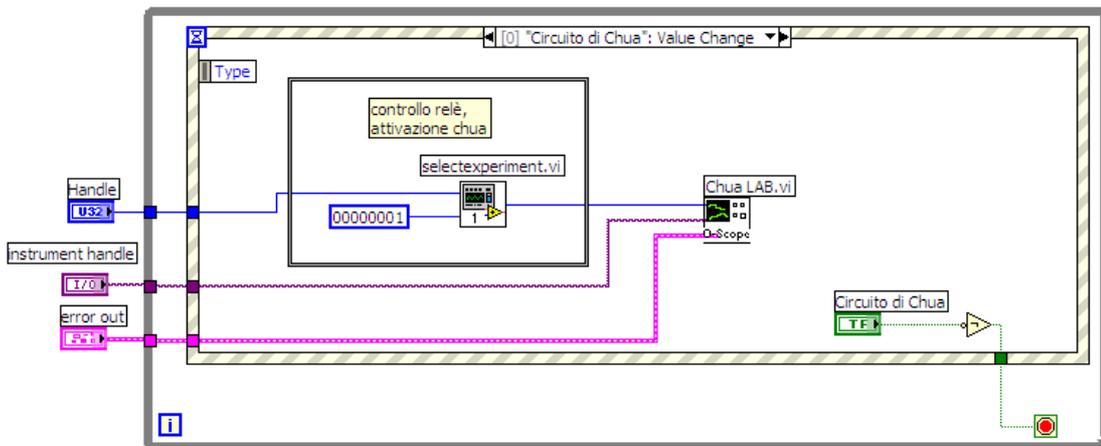


Figura 3.7 - Block diagram di “Esperimenti circuiti.vi”

Il sub-VI riceve dal programma principale, i parametri ottenuti con le inizializzazioni, in modo da poterli passare ai VI che ne necessitano.

Anche per questo VI è stata adottata una event structure composta da cinque frame di cui quattro per la scelta degli esperimenti ed una per tornare al menu principale (mainprogram.vi).

La *prima frame* viene eseguita quando l'utente sceglie l'esperimento di Chua, cioè quando viene premuto il pulsante “Circuito di Chua”. Il sub-VI selectexperiment riceve in ingresso l'handle ed il byte “00000001”; in questo modo, viene pilotato il doppio

multiplexer a relè per mettere in connessione il circuito di Chua con i canali d'ingresso analogici della scheda d'acquisizione PCI-5102. Terminata questa prima fase di "selezione dell'esperimento", viene richiamato il sub-VI "Chua LAB.vi⁵" e viene visualizzato il relativo frontal panel mostrato in figura 3.8. Questo programma fu realizzato in passato per una tesi precedente ed è stato ripreso per adattarlo alle nostre esigenze, riordinando il pannello frontale e collegando il VI al mainprogram. Il programma riceve in ingresso i controllori "error in" e "instrumental handle" ottenuti con l'inizializzazione della scheda d'acquisizione e l'handle del modulo UM245R. Esso consente di interagire col circuito di Chua, di modificare il valore della resistenza variabile attraverso gli appositi comandi, e di osservare i risultati su un grafico X-Y che mostra i valori delle tensioni sui due condensatori del circuito.

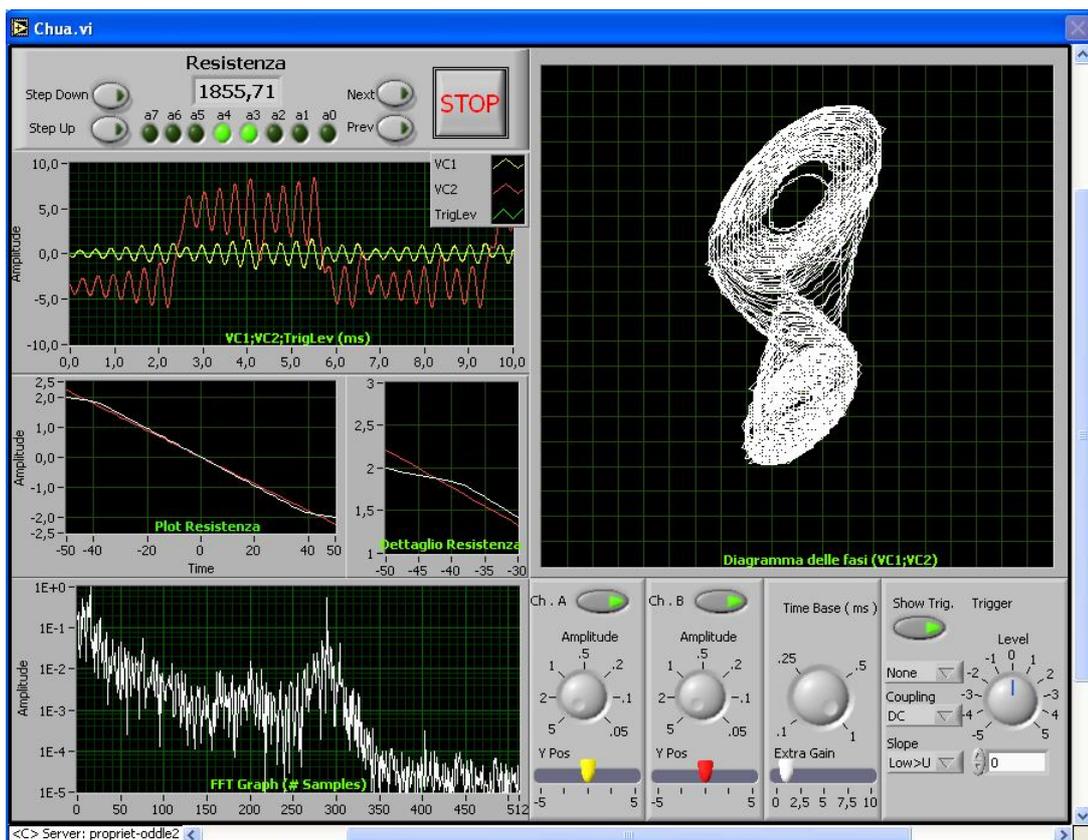


Figura 3.8 - Frontal panel di "Chua LAB.vi"

⁵ Per indicare che si tratta della versione "da laboratorio" e distinguerla da quella "remota".

La *seconda frame* della event structure, è mostrata in figura 3.9, ed è del tutto analoga alla prima: viene eseguita quando l'utente preme il pulsante "circuito RLD" e attraverso il sub-VI "selectexperiment.vi" consente di collegare fisicamente il circuito RLD alla scheda di acquisizione PCI-5102, per poterne acquisire le tensioni.

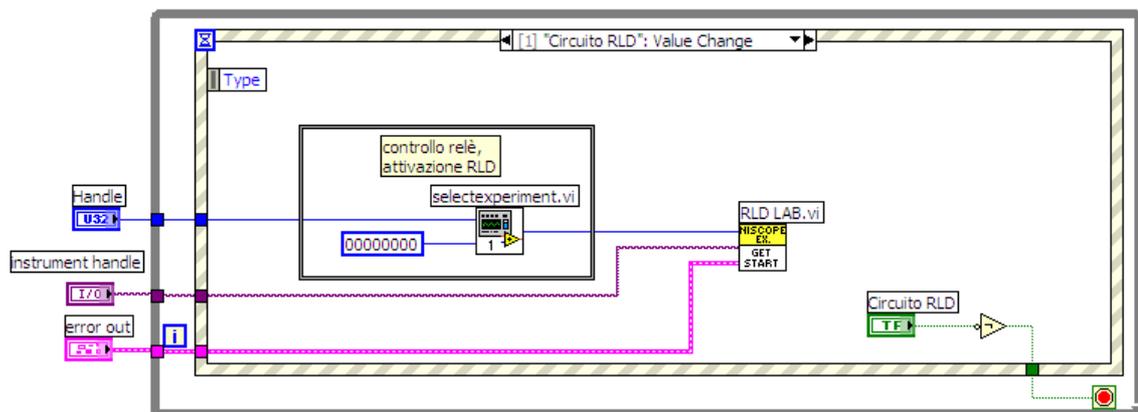


Figura 3.9 - Seconda frame di "Esperimenti circuiti.vi" per la selezione del circuito RLD

Il sub-VI "RLD LAB" riceve come ingressi l'handle e le inizializzazioni della scheda d'acquisizione; quando viene richiamato, si apre il relativo frontal panel con i comandi per modificare l'ampiezza del segnale sinusoidale in ingresso al circuito RLD e gli indicatori per visualizzarne i risultati sul "display virtuale". Il frontal panel di questo programma, mostrato in figura 3.10, realizza l'interfaccia di un "oscilloscopio virtuale" che ricorda molto quella di un oscilloscopio tradizionale. Questo VI fu realizzato in passato per un tesi precedente, tuttavia esso è stato completamente rivisitato da un punto di vista grafico e modificato per quanto riguarda la parte relativa al generatore di segnali: esso infatti pilotava inizialmente il generatore di segnali HP-33120A, tramite interfaccia GPIB; adesso invece, volendo utilizzare il generatore di segnali portatile, integrato nel prototipo, è stato necessario sostituire i sub-VI che controllavano il generatore HP, con quelli che ora pilotano il nuovo circuito. Su questo VI è stato inoltre svolto un lavoro di debug per eliminarne alcuni malfunzionamenti manifestatisi durante l'utilizzo del programma. Uno di questi riguardava la manopola "Sign. Ampl." in alto a sinistra nella figura 3.10. Il problema consisteva nel fatto che, a fronte di una variazione significativa dell'ampiezza del segnale d'ingresso, il diagramma delle fasi non veniva

ridimensionato, e questo provocava difficoltà nell'analisi delle dinamiche caotiche. Per risolvere questo problema è stata introdotta nel programma, una *case structure* alla quale è stata posta come condizione d'accesso, la variazione del valore della manopola in oggetto: quando si verifica la variazione dell'ampiezza attraverso questa manopola, viene eseguito il sub-VI "scalegraph.vi" che ritorna dei valori tali da ottimizzare la visualizzazione del diagramma. Questi valori vengono passati, mediante dei property node, alla sezione del programma che regola le scale del grafico.

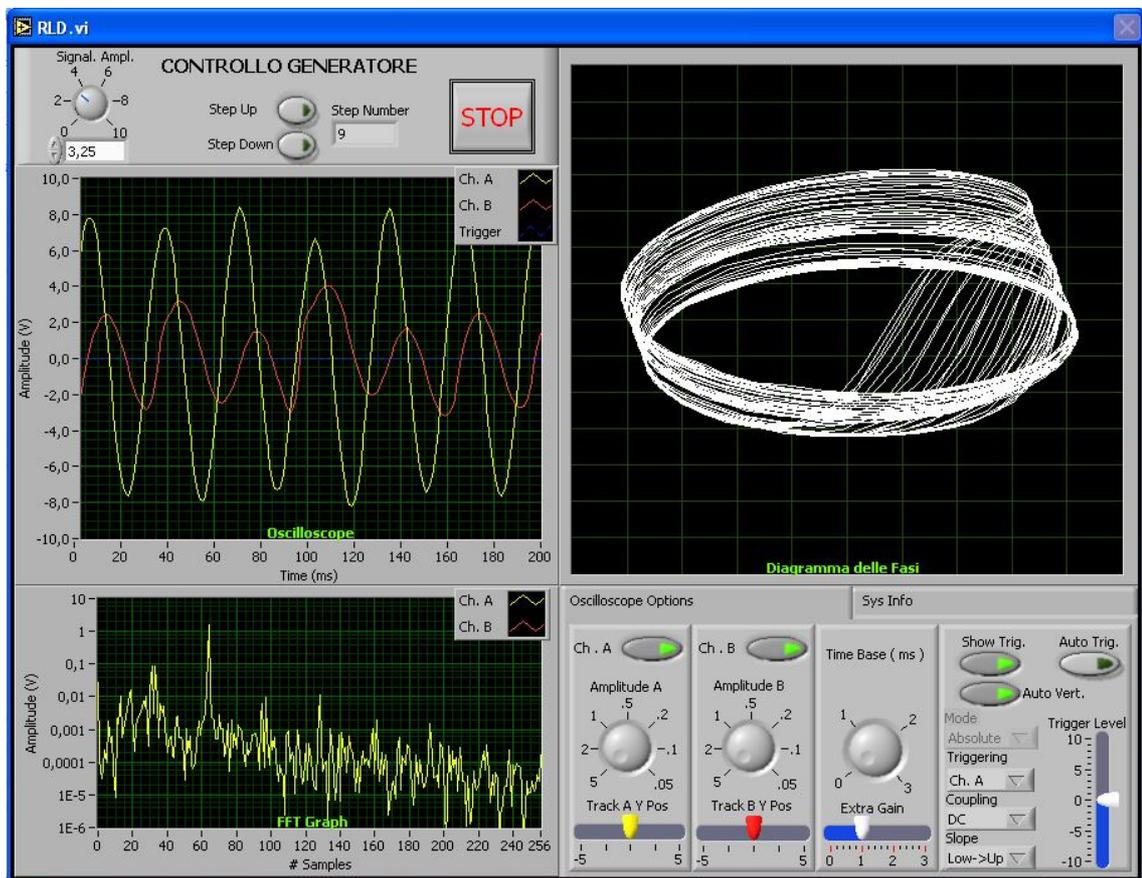


Figura 3.10 - Frontal panel di "RLD LAB.vi"

In figura 3.11 è mostrata una sezione del block diagram "RLD LAB.vi" in cui è possibile notare il blocco di codice che effettua la regolazione del diagramma delle fasi, il sub-VI "setupGeneratore" e quello "SigGenSetAmplitude.vi".

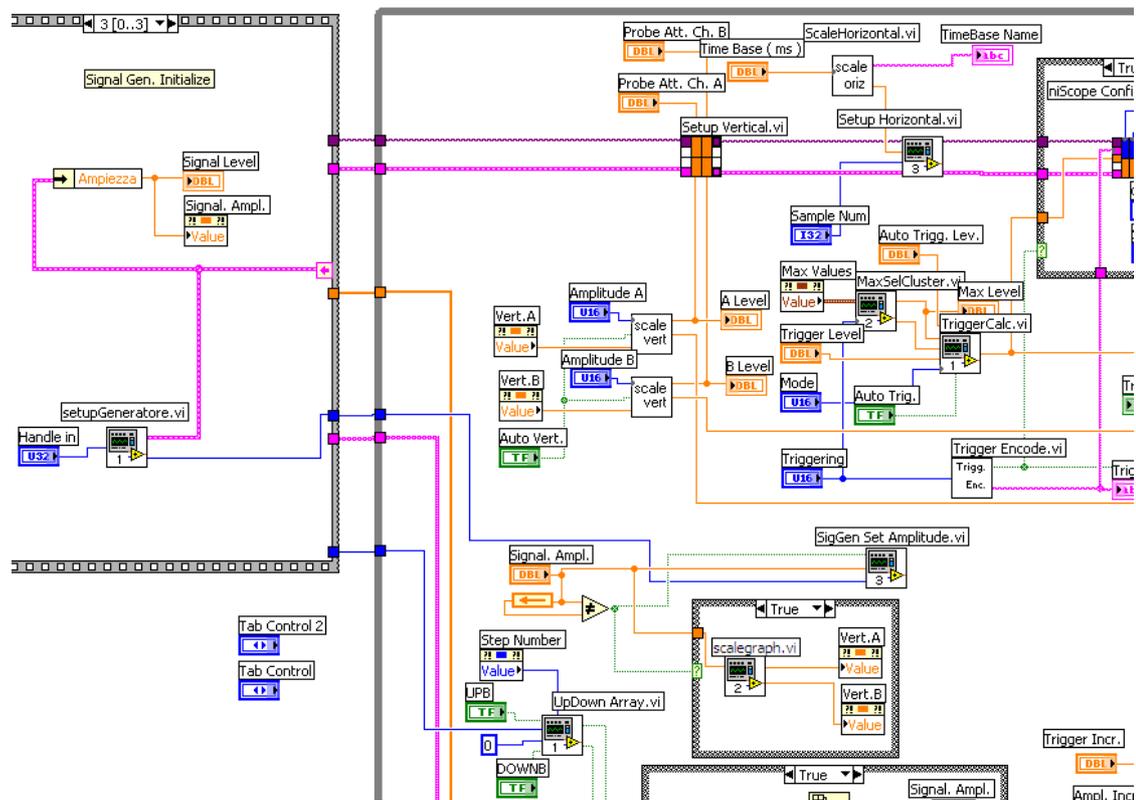


Figura 3.11 - Block diagram di “RLD LAB.vi”

Il sub-VI “setupGeneratore.vi” riceve in ingresso l’handle dell’integrato UM245R ed un cluster con i valori precaricati⁶ di ampiezza, frequenza ed offset da impostare nel generatore di segnali; in uscita, ritorna l’handle del modulo USB per consentire le operazioni successive di scrittura nel prototipo. terminate le inizializzazioni del generatore e della scheda di acquisizione 5102, il programma attende che l’utente modifichi l’ampiezza del segnale sinusoidale in ingresso al circuito RLD, mediante uno dei controllori disposti a tal scopo sul pannello frontale; quando si verifica la variazione di uno di essi, viene eseguito il sub-VI “SigGenSetAmplitude.vi”, realizzato per impostare l’ampiezza del segnale in uscita al generatore. Sul display del pannello

⁶ Tali valori sono stati scritti nel file di configurazione “setupGEN.ini”, al quale il software accede in fase di inizializzazione.

frontale, è possibile così osservare le dinamiche del circuito RLD, al variare dell'ampiezza del segnale di input.

La terza e la quarta frame, sono state predisposte per includere eventuali esperimenti futuri; per adesso, esse mostrano semplicemente un messaggio informativo:

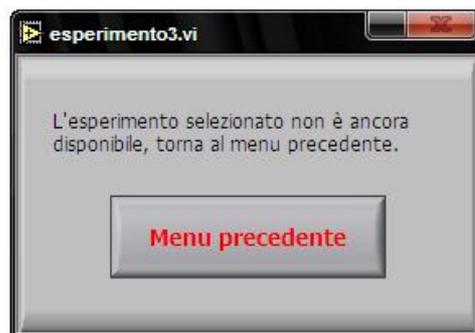


Figura 3.12 – Frontal panel di “esperimento3.vi”

Infine la quinta frame viene eseguita quando si preme il pulsante “Torna al menu principale”, il quale è stato connesso direttamente al nodo di terminazione del ciclo while; ciò consente di chiudere la finestra e di ritornare al menu principale.

Passiamo ora a descrivere il block diagram del sotto-programma “Scheda e generatore.vi”, posto all’interno di “mainprogram.vi” e il cui pannello frontale è mostrato in figura 3.4.

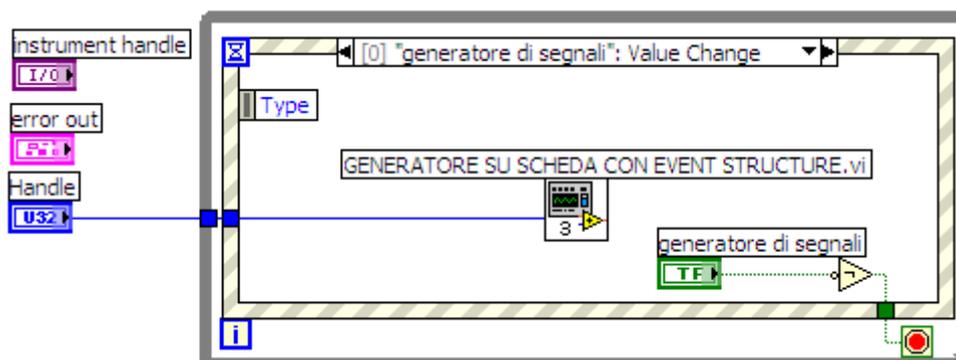


Figura 3.13 - Block diagram di “scheda e generatore.vi”

Il sub-VI riceve dal programma principale, i parametri ottenuti con le inizializzazioni, in modo da poterli passare ai VI che ne necessitano.

Anche per questo VI è stata adottata una event structure composta da cinque frame di cui quattro per la scelta delle attività ed una per tornare al menu principale (mainprogram.vi).

La prima frame della struttura ad eventi, mostrata in figura 3.13, contiene al suo interno soltanto il sub-VI per il controllo del generatore di segnali ed il pulsante “generatore di segnali”: quando l’utente clicca sul pulsante, si genera un evento e labview esegue la frame aprendo il frontal panel del sub-VI “controllo generatore su scheda con event structure.vi”.

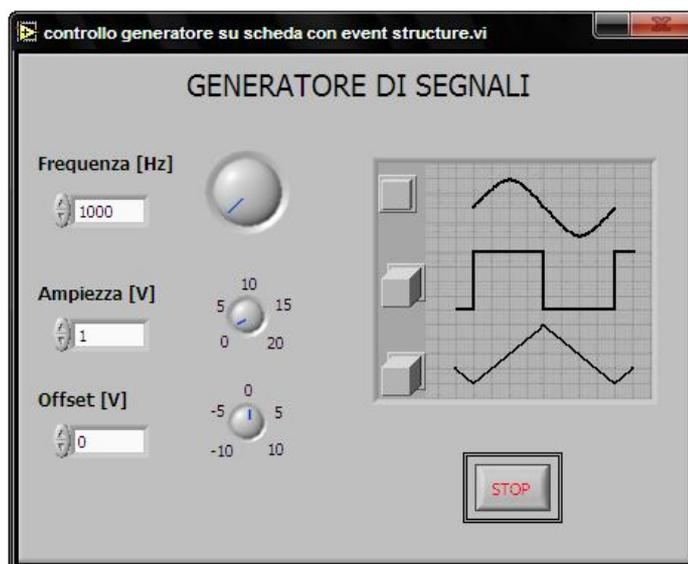


Figura 3.14- Frontal panel di “generatore su scheda con event structure.vi”

Il software di controllo del generatore di segnali è stato realizzato per una tesi precedente, tuttavia, con l’integrazione hardware del generatore, del circuito d’I/O e del circuito di switch su di un’unica scheda, si è reso necessario modificare alcuni sub-VI per evitare conflitti tra gli integrati. Tutti gli integrati del circuito infatti, sono pilotati dal master (UM245R) e condividono il bus SPI: in fase di scrittura in un particolare integrato per evitare scritture involontarie negli altri, è stato necessario disabilitarli, ponendone il chip select a livello logico alto. In questo modo si realizza una

comunicazione esclusiva, tra il master e lo slave senza produrre interferenze negli altri circuiti integrati. Rispetto alla versione di partenza, è stato inoltre sostituito il ciclo while, con una event structure che, come detto in precedenza, risulta essere più performante in termini di risorse impegnate. In figura sono riportate le frame che realizzano le funzioni del generatore di segnali:

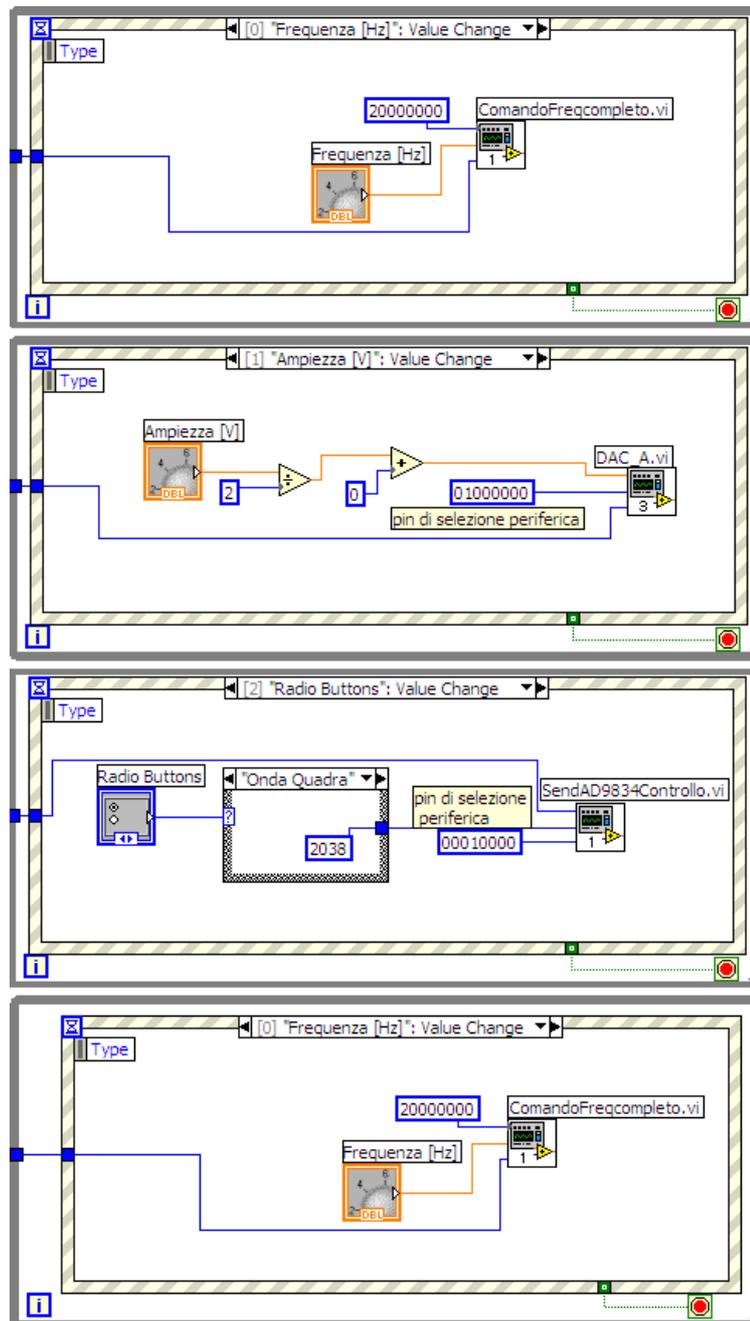


Figura 3.15 - Frame di "generatore su scheda con event structure.vi"

La prima frame setta la frequenza del segnale generato, la seconda frame ne setta l'ampiezza, la terza il tipo di forma d'onda, la quarta regola l'offset.

Davanti alla event structure appena mostrata, è stata anche aggiunta una stacked sequence structure che, all'avvio del programma, resetta i valori di frequenza, ampiezza ed offset, a dei valori di default rispettivamente di 1MHz, 1V, 0V. Come tipo di forma d'onda generata è stata impostata di default, un'onda sinusoidale.

La seconda frame del sotto-programma “Scheda e generatore.vi”, permette l'invio di una tensione compresa tra 0 e 4,095V, sul piedino d'uscita del DAC del circuito d'acquisizione di input-output.

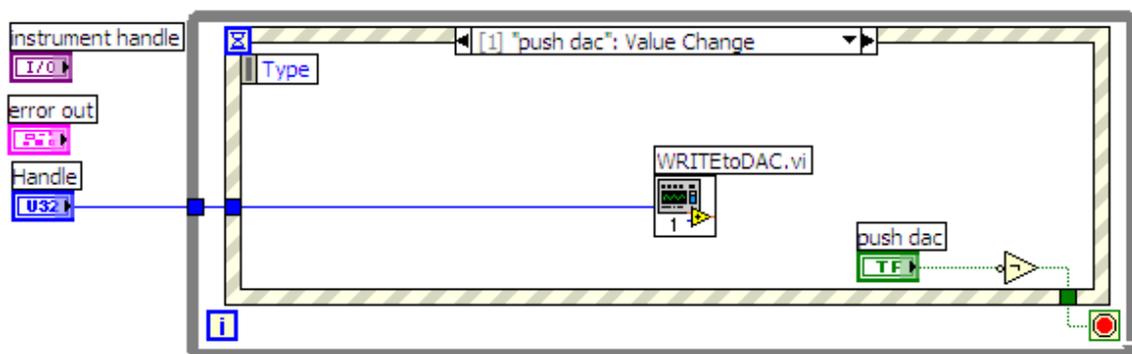


Figura 3.16 - Seconda frame di “scheda e generatore” per l'utilizzo del DAC

Quando viene premuto il tasto “push DAC”, viene eseguito il sub-VI “WRITEtoDAC.vi” che permette di inviare un valore di tensione compreso tra 0 e 4,095V, al piedino d'uscita del DAC del circuito di I-O analogico/digitale. Il frontal panel è mostrato in figura.



Figura 3.17 - Frontal panel di “WRITEtoDAC.vi”

La terza e la quarta frame, sono state predisposte per includere eventuali funzionalità future; per adesso, esse mostrano semplicemente un messaggio informativo:



Figura 3.18 - “Frontal panel di “funzionalità 3.vi”

Infine la quinta frame viene eseguita quando si preme il pulsante “Torna al menu principale”, il quale è stato connesso direttamente al nodo di terminazione del ciclo while; ciò consente di chiudere la finestra e di ritornare al menu principale.

3.2.3 Versione del software per l’utilizzo da remoto

Per l’utilizzo da remoto è stato realizzato un software dedicato, molto simile a quello appena descritto, denominato “mainprogramREMOTO.vi”: a differenza del precedente, è stata realizzata un’interfaccia su 2 livelli con la possibilità di selezionare solo gli esperimenti d’interesse didattico, ovvero il circuito di Chua e quello RLD; sono poi stati predisposti due link per gli esperimenti futuri. Il frontal panel del programma è mostrato in figura 3.20; l’interfaccia risulta estremamente semplice, allo scopo di consentire a qualsiasi utente di accedere agli esperimenti senza dover necessariamente conoscere il software di programmazione labview.

Esso infatti non richiede né l’avvio, né l’arresto da parte dell’utente, giacché queste operazioni vengono preventivamente eseguite dal laboratorio di circuiti, ove il software è installato. Per ottenere questa semplificazione, è necessario però che il tasto stop non sia visibile da remoto, poiché l’interruzione del programma, non ne consentirebbe più il riavvio, se non da parte dell’amministratore. Per ottenere questo risultato è dunque necessario “ritagliare” l’area che si desidera mostrare in remoto, escludendo il tasto

stop; la procedura verrà descritta in dettaglio nel capitolo dedicato alla remotizzazione web.



Figura 3.20 - Frontal panel di “mainprogramREMOTO.vi” visualizzato sul PC server

A questo punto l'utente che tramite internet accede al “laboratorio remoto di circuiti”, deve semplicemente preoccuparsi di selezionare l'esperimento desiderato; fatto ciò, a seconda della scelta eseguita, vedrà apparire il frontal panel del circuito di Chua o quello del circuito RLD, rispettivamente mostrati in figura 3.8 e 3.10. I pulsanti “Circuito 3” e “Circuito 4” richiamano invece un messaggio informativo, tipo quello di pagina precedente.

Il block diagram che esegue queste operazioni è mostrato in figura 3.21:

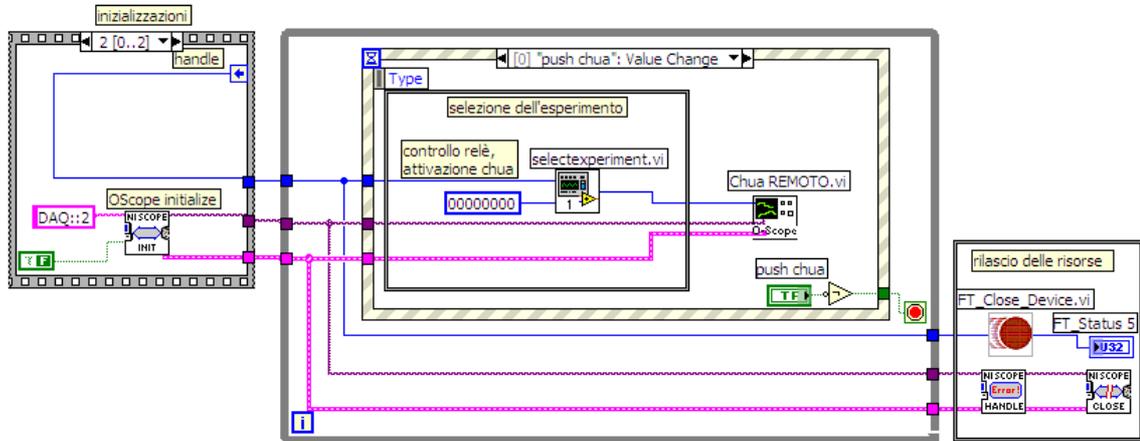


Figura 3.21 - Block diagram di “mainprogramREMOTO.vi”

Esso presenta la stessa struttura della versione “da laboratorio”: abbiamo una event structure con cinque frame di cui quattro per gli esperimenti ed una per l’arresto del programma. All’esterno della event structure, vi sono le inizializzazioni del modulo UM245R e della scheda d’acquisizione 5102, i dati così ottenuti attraversano il ciclo while per garantire la sequenzialità delle operazioni e vengono *passati* all’interno della event structure ai sub-VI che ne necessitano. Quando l’utente seleziona un esperimento, viene attivata la relativa frame, selezionato il circuito scelto attraverso il pilotaggio dei multiplexer, ed eseguito l’esperimento. Al termine di un esperimento, il programma torna alla event structure ed attende un nuovo evento. Quando si verifica l’evento “stop”, ovvero quando l’amministratore preme il relativo pulsante, la struttura ad eventi termina e vengono eseguiti i sub-VI posti all’esterno, per il rilascio delle risorse inizializzate.

Tuttavia la grossa differenza rispetto alla versione “da laboratorio”, si trova all’interno dei sub-VI “Chua Remoto.vi” ed “RLD Remoto.vi”; la necessità di introdurvi delle varianti, nasce da un malfunzionamento del software che può verificarsi proprio con l’utilizzo da remoto: quando un utente accede al laboratorio remoto di circuiti, visualizza in una finestra del browser web il programma principale “mainprogramREMOTO.vi” in versione “da remoto”, così come mostrato in figura 3.20; l’utente selezionando uno tra gli esperimenti proposti vedrà apparire una finestra pop-up con i controlli e gli indicatori dell’esperimento scelto; a questo punto, se l’utente

dovesse chiudere incautamente il browser web che sta eseguendo “mainprogramREMOTO.vi”, anche la finestra pop-up verrebbe chiusa; tuttavia sul computer server del laboratorio ove è installato il software, il programma rimarrebbe in esecuzione, provocando un grave disagio agli utenti che in seguito volessero usufruire degli esperimenti. Si possono infatti verificare due casi:

- L’utente che si connette al laboratorio remoto seleziona lo stesso esperimento che l’ultimo utente ha eseguito e terminato in modo non previsto: in questo caso, chi si connette successivamente, visualizza il pannello frontale dell’esperimento nello stato in cui è stato lasciato dal precedente utente. E’ possibile tuttavia controllare l’esperimento, terminarne l’esecuzione e selezionarne un altro tra quelli proposti.
- L’utente che si connette al laboratorio seleziona un esperimento diverso da quello che l’ultimo utente ha eseguito e terminato in modo non previsto: in questo caso il programma si blocca, non permettendo più l’esecuzione di alcun esperimento.

Per risolvere questo malfunzionamento del programma, è stato quindi inserito all’interno dei programmi “Chua remoto.vi” e “RLD remoto.vi”, un blocco di codice che effettua ricorsivamente un controllo per verificare se l’utente è connesso effettivamente al laboratorio remoto oppure no; nel momento in cui l’utente dovesse risultare disconnesso, l’esperimento in esecuzione termina automaticamente, consentendo a chiunque altro desideri utilizzare gli esperimenti, di farlo senza problemi. Questa cosa fornisce autonomia e robustezza al laboratorio remoto di circuiti, riducendo al minimo l’intervento umano dal lato server.

Il codice che effettua questo controllo, come già accennato, è stato inserito all’interno dei VI per il controllo del circuito di Chua e del circuito RLD. Esso è rappresentato in figura 3.22 in cui però, la restante parte del codice dell’esperimento, presente nel ciclo while, è stata schematizzata per semplicità con un rettangolo.

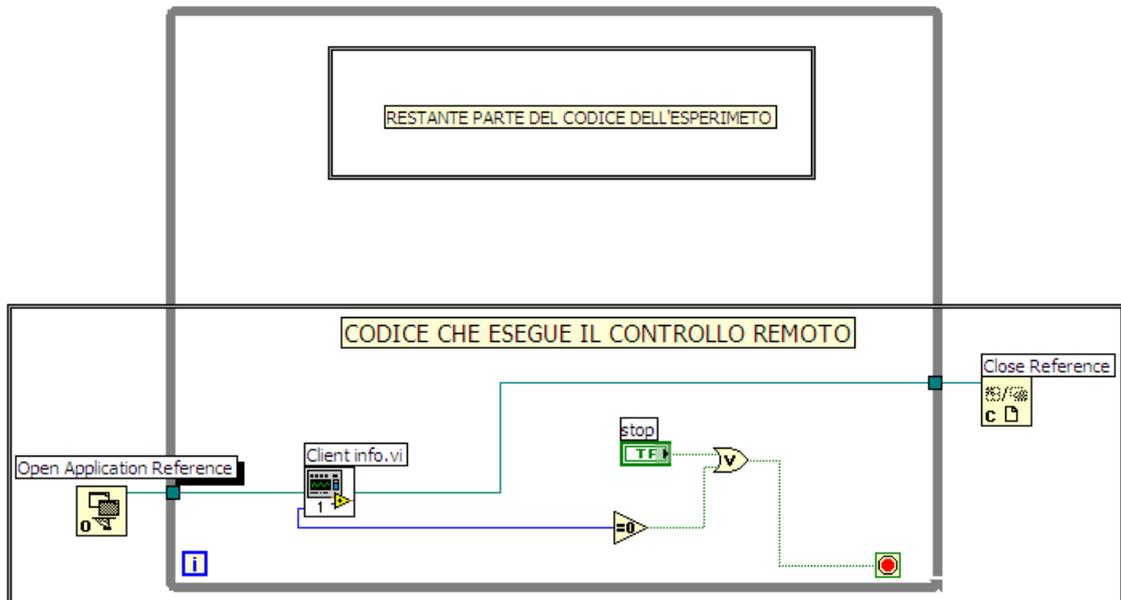


Figura 3.22 - Codice all'interno di "Chua remoto.vi" e RLD "remoto.vi" che esegue il controllo remoto

Il sub-VI "Client info.vi" ritorna un array di clusters contenenti informazioni sui clients⁷, che da remoto sono connessi agli esperimenti proposti. Tra le informazioni che si possono ottenere, vi sono:

- il nome del PC remoto;
- il numero di porta attualmente utilizzato dal computer client.
- Il tempo trascorso dall'inizio della connessione;
- Lo stato di connessione al server (se è 0, l'utente è disconnesso).

Sfruttando proprio il valore della variabile "status", è possibile far terminare l'esecuzione dell'esperimento quando l'utente chiude inaspettatamente la finestra del browser web.

⁷ Nel nostro caso, trattandosi di un laboratorio remoto "monoutente", l'utente connesso sarà soltanto uno.

3.3 Virtual Instrument per il testing del prototipo

Per effettuare il testing dei vari componenti del circuito, man mano che avveniva il loro montaggio sulla scheda, sono stati realizzati alcuni VI “ad hoc”, illustrati di seguito.

3.3.1 Sub-VI: ”TestUM245R.vi”

Questo VI esegue la scrittura di un byte, sui piedini DB0÷DB7 dell’integrato UM245R, attraverso un’opportuna concatenazione di sub-VI mostrata in figura:

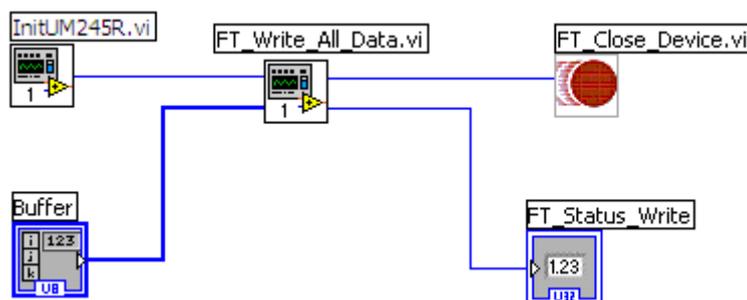


Figura 3.23 - Block diagram di “TestUM245R.vi”

- “InitUM245R.vi” inizializza il modulo USB;
- “FT_Write_All_Data.vi” effettua la scrittura del byte “Buffer”, sui piedini dell’”UM245R.vi”;
- FT_Close_Device.vi termina la comunicazione col il dispositivo.

3.3.2 Sub-VI: ”Test MCP23S17.vi”

Questo programma esegue l’invio di un byte sulla porta A e/o B dell’integrato MCP23S17, attraverso un opportuno collegamento dei sub-VI mostrati in figura 3.24, e inseriti all’interno di una flat sequence structure. Tramite questo VI è possibile testare contemporaneamente sia il port expander (MCP23S217) che il multiplexer a relè, essendo questi due componenti connessi tra loro. Per il testing basta scrivere un certo byte sulle porte del port expander e verificare le uscite dei due dispositivi.

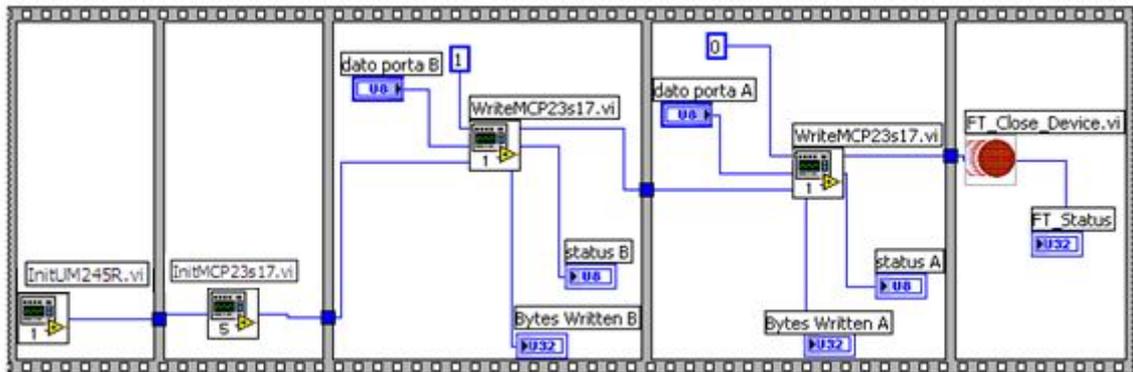


Figura 3.24 - “TestMCP23S17.vi”

- “InitUM245R.vi” inizializza il modulo USB;
- “InitMCP23S17.vi” inizializza l’integrato MCP23S17 per eseguire la scrittura sulla porta A e/o B;
- “WriteMCP23S17.vi” scrive sulla porta A se riceve in ingresso il bit “0”, sulla porta B e riceve il bit “1”;
- FT_Close_Device.vi termina la comunicazione col il dispositivo.

Infine per il testing del generatore di segnali e del DAC del circuito d’acquisizione, sono stati utilizzati i rispettivi programmi di controllo: “generatore su scheda con event structure.vi” visto in precedenza e mostrato in figura 3.15 e “WRITEtoDAC.vi” (figura 3.17) che verrà descritto in seguito nei files di libreria.

3.4 Sub-VIs di libreria per il software di controllo

In questo paragrafo verranno descritti singolarmente, i sub-VI utilizzati per il software di gestione del laboratorio di circuiti, suddivisi in librerie; particolare attenzione verrà rivolta ai sottoprogrammi aggiunti o modificati rispetto a quelli già esistenti, per i quali verranno descritti solo gli aspetti più importanti e si rimanda comunque agli elaborati precedenti.

3.4.1 Files di libreria del modulo UM245R

-InitUM245R.vi

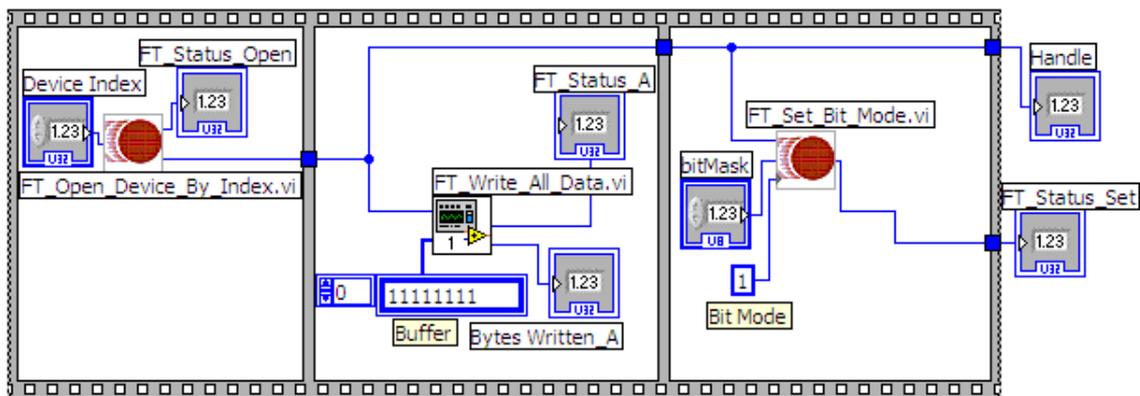


Figura 3.25 – Block diagram di “InitUM245R.vi”

Questo SubVI è stato realizzato tramite una Flat Sequence Structure, una struttura che permette la sequenzializzazione dei sub-diagrammi contenuti in essa, o per meglio dire delle funzioni presenti al suo interno. La prima frame contiene la funzione FT_OPEN_DEVICE_BY_INDEX.vi, la quale apre la comunicazione con il dispositivo UM245R. Essa riceve in ingresso il parametro Device Index, un controllore che indica il numero di dispositivi da aprire, connessi al PC; nel caso in cui ce ne sia uno solo occorre porlo uguale a zero, se invece sono due bisogna settarlo a uno e così via. Dato che nella nostra realizzazione usiamo un solo dispositivo questo parametro avrà come valore zero. La funzione ritorna il valore del puntatore Handle, un indicatore che mostra una sorta di “indirizzo” identificativo del dispositivo collegato, che sarà usato per accessi successivi ed il parametro FT_STATUS, un indicatore che fornisce lo stato della funzione. La seconda frame setta ad 1 tutti i pin d’uscita del modulo UM245R; tale operazione è necessaria poiché quando viene attivato l’integrato, possono verificarsi delle transazioni spurie sui CS, con conseguente attivazione degli integrati. Per la sua implementazione è stato utilizzato il SubVI FT_Write_All_Data.vi, che consente di scrivere dati all’ interno del dispositivo UM245R; essa riceve in ingresso un buffer di

valori costanti, in particolare 11111111, i quali corrispondono rispettivamente ai pin DB7-DB0; la funzione ritorna la FT_Status_A un indicatore che fornisce lo stato della funzione e i Bytes Written_A che indica il numero di Byte effettivamente scritti. L'ultima frame configura l'integrato UM245R: è stato utilizzato per la sua implementazione la funzione FT_SET_BIT_MODE.vi, la quale consente di configurare i pin d'uscita DB0÷DB7, singolarmente come input o come output. In particolare, tale funzione riceve in ingresso oltre all'Handle della FT_OPEN_DEVICE_BY_INDEX.vi, il valore della maschera "Bit Mask", un controllore nel quale i bit sono configurati singolarmente per l'input o per l'output: se il bit è 0, il corrispondente pin è settato come input, se il bit è 1, il corrispondente pin è settato come output. Nel nostro caso, è stato settato come "01111111", cui bit vanno letti dal più significativo a quello meno significativo, ovvero da DB7 a DB0. DB0 è configurato per l'invio dei dati (MOSI), il pin DB1 è configurato per la trasmissione del clock (SCLK), i pin DB2-DB6 sono configurati come \overline{CS} , per l'attivazione dei dispositivi Slave ed infine il pin DB7 per la ricezione dei dati (MISO).

Un altro input della funzione è il "bit mode", un controllore che serve a settare il tipo di configurazione del dispositivo: nel nostro caso è stato posto ad 1 in quanto indica la configurazione di tipo "bit bang mode" asincrona da noi utilizzata.

In uscita alla funzione abbiamo l'indicatore FT_STATUS_SET, che fornisce lo stato della stessa.

- FT_Write_All_Data.vi

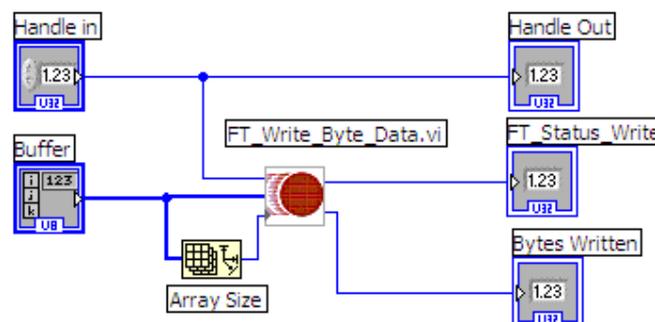


Figura 3.27 – FT_Write_All_Data.vi

Questo sub-VI esegue la scrittura di un array di byte sui piedini del modulo USB. La funzione FT_Write_Byte_Data.vi riceve in ingresso l'handle del dispositivo, l'array con i byte da scrivere, ed il numero stesso di byte da inviare al modulo.

- FT_Close_Device.vi

Questo sub-VI riceve in ingresso l'handle del dispositivo e ne termina la comunicazione.

3.4.2 Files di libreria del Port Expander MCP23S17

-InitMCP23S17.vi

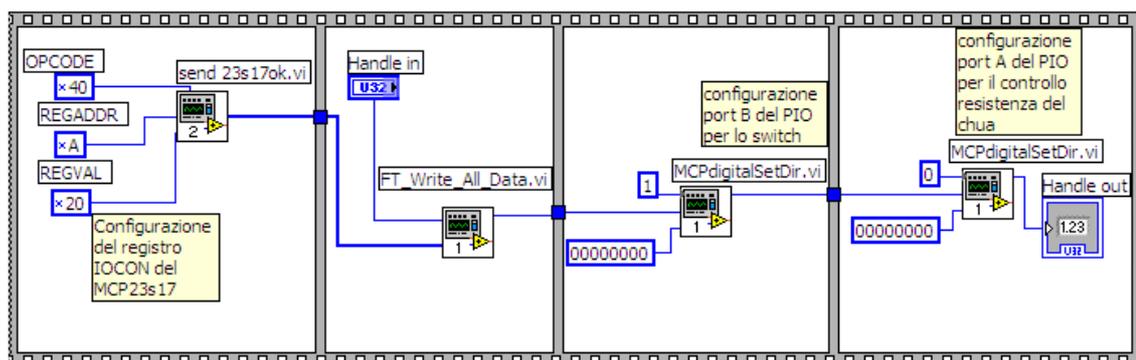


Figura 3.26 - InitMCP23S17.vi

In questo sub-VI viene effettuata la configurazione dell'integrato MCP23S17.

Nella prima frame, il sub-VI send23s17ok.vi riceve i dati di configurazione da scrivere nel registro IOCON dell'integrato; restituisce un vettore di byte che viene inviato al port expander attraverso la funzione FT_Write_All_Data.vi che va a scrivere nel modulo UM245R. Per ottenere una comunicazione esclusiva tra il master ed il port expander il sub-VI send23s17ok.vi è stato modificato, in modo da porre tutti i CS relativi agli altri integrati del prototipo, a livello logico alto.

Con la terza e la quarta frame si va a scrivere rispettivamente nel registro IODIRA e IODIRB in modo da settare i pin della porta A e B dell'integrato come input oppure come output; per i nostri scopi, tutti i pin di I/O, devono essere settati come output, per

cui la funzione MCPdigitalSetDir.vi riceve in ingresso un byte di tutti “0”. Tale funzione è illustrata in figura 3.27.

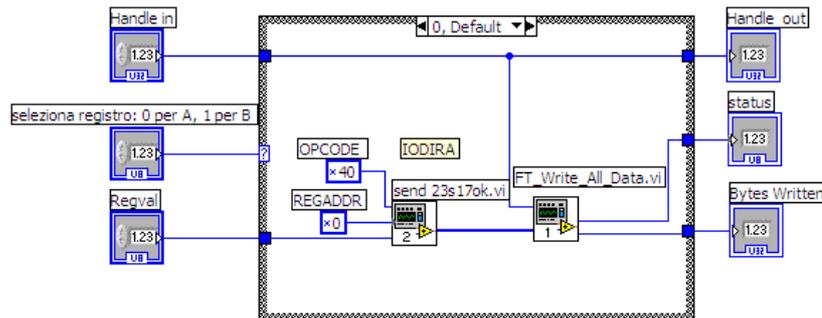


Figura 3.27 - Block diagram di MCPdigitalSETdir.vi

-WriteMCP23S17.vi

Questo SubVI, realizzato per una tesi precedente, effettua la scrittura nei registri OLATA e OLATB dei dati in uscita dai rispettivi lati del dispositivo. La sua implementazione è stata realizzata utilizzando una struttura CASE, alla quale è stata posta come condizione di accesso la selezione di uno dei due registri, in particolare è stato associato 0 al registro A e 1 al registro B. All'interno della struttura, troviamo inizialmente un collegamento diretto tra l'indicatore Handle in e il controllore Handle out; inoltre è stata posta la funzione FT_WRITE_All_DATA.vi che riceve come ingressi insieme all'Handle_in, l'array in uscita dal Sub-VI send23s17ok.vi; Per quanto riguarda il sub-VI send23s17ok.vi, esso riceve in ingresso insieme all'indicatore “Dato da trasmettere”, che contiene il byte da inviare, altri due ingressi: “opcode” (il cui valore, quando si effettua l'operazione di scrittura, è in rappresentazione esadecimale pari a x40 per entrambi i registri OLATA/B) ed infine “regaddr” (che contiene l'indirizzo in rappresentazione esadecimale, del registro OLATA (x15) e del registro OLATB (x14))[2].

Il block diagram di questo sub-VI è mostrato in figura 3.28.

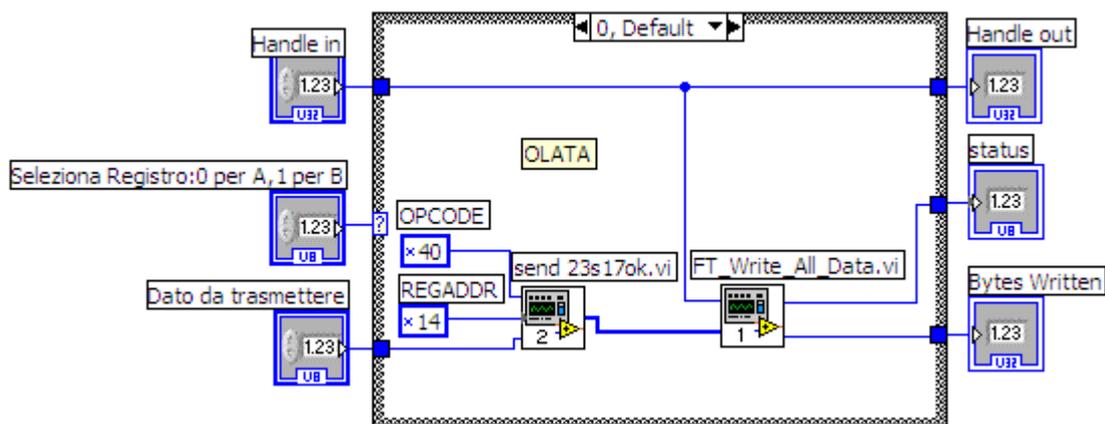


Figura 3.28 - WriteMCP23S17.vi

3.4.3 File di libreria del DAC TLV5616

-WRITEtoDAC.vi

Invia sul piedino d'uscita del DAC del circuito d'acquisizione, una tensione desiderata, compresa tra 0 e 4,095V.

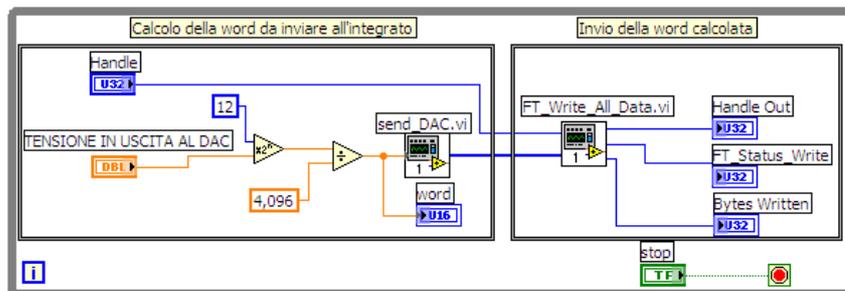


Figura 3.29 – Block diagram di “WRITEtoDAC.vi”

Esso è composto da due sottosezioni, di cui una calcola il valore della word da inviare all'integrato a partire dalla tensione desiderata, l'altra ne realizza l'invio.

Ricordando che la tensione in uscita al DAC è:

$$V_{\text{out}} = \frac{\text{word} \cdot 2 \cdot V_{\text{ref}}}{2^{12}}$$

(dove word è un valore in binario di 12 bit compreso tra 0 e 4095), la word da inviare all'integrato per ottenere in uscita V_{out} è data da:

$$\text{word} = \frac{V_{\text{out}} \cdot 2^{12}}{2 \cdot V_{\text{ref}}}$$

Questa word va in ingresso al sub-VI send_DAC che realizza, a partire da questo valore, il segnale da inviare effettivamente all'integrato.

-Send_DAC.vi

Questo SubVI è stato realizzato per una tesi precedente e consente di realizzare il segnale effettivamente inviato al DAC.

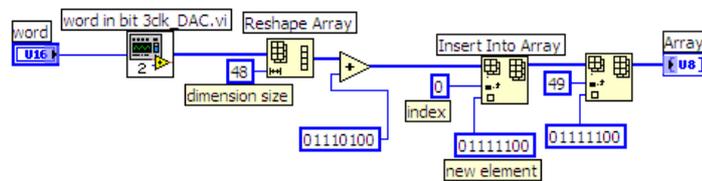


Figura 3.30- Block diagram di “send_DAC.vi”

Rispetto alla versione esistente, è stata apportata una modifica in modo da ottenere una comunicazione esclusiva tra master (UM245R) e slave (DAC) senza interferire con gli altri integrati: è stato necessario aggiungere come da figura, il byte “01111100” con il quale vengono *alzati* tutti i CS relativi agli integrati, con esclusione di quello (evidenziato con una sottolineatura) associato al DAC.

3.4.4 Files di libreria del Generatore di segnali

-SetupGeneratore.vi

Questo sub-VI è stato inserito all'interno del programma “RLDlab.vi” e “RLDremoto.vi” per l'inizializzazione del generatore. Il sub-VI riceve in ingresso l'handle dell'integrato UM245R ed un cluster con i valori preconfigurati di ampiezza, frequenza ed offset da impostare nel generatore di segnali, prelevandoli da un file di configurazione salvato sul PC;

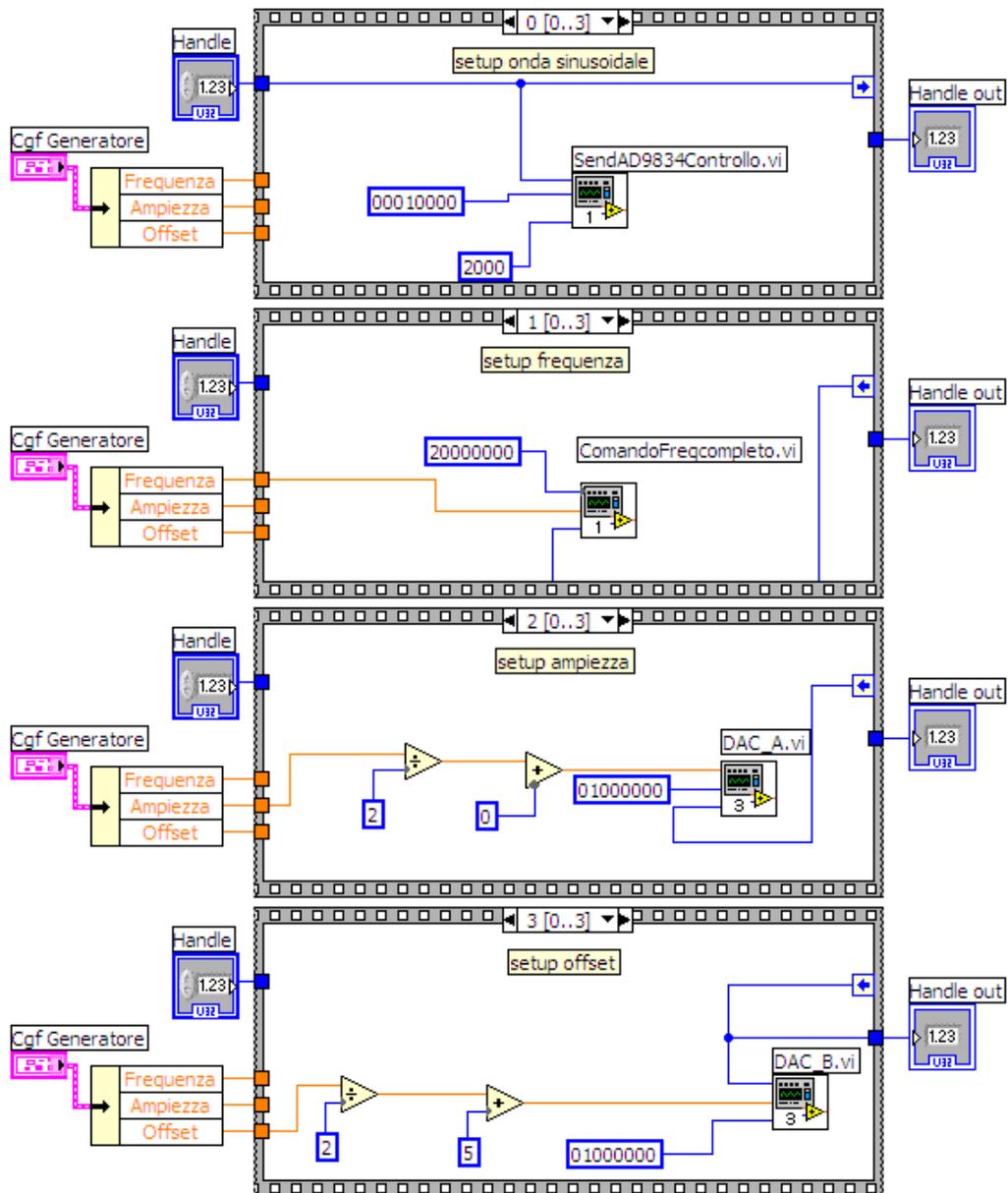


Figura 3.31 - Frame di “ SetupGenerator.vi”

in uscita, ritorna l’handle per consentire le operazioni successive di scrittura nel modulo USB. Per la realizzazione è stata utilizzata una stacked sequence structure, cui frame sono mostrate in figura 3.31.

-SigGenSetAmplitude.vi

Questo sub-VI è stato inserito all'interno del programma "RLDlab.vi" e "RLDremoto.vi", sostituendo di fatto il sub-VI che regolava l'ampiezza del segnale generato dall'HP-33120. Il sub-VI riceve in ingresso un valore booleano, il valore dell'ampiezza da settare e l'Handle del dispositivo USB ed è realizzato attraverso una *case structure*: quando viene modificata l'ampiezza del segnale, "Update Ampl." diventa "vero", la structure viene eseguita ed avviene il settaggio dell'ampiezza in uscita al generatore di segnali.

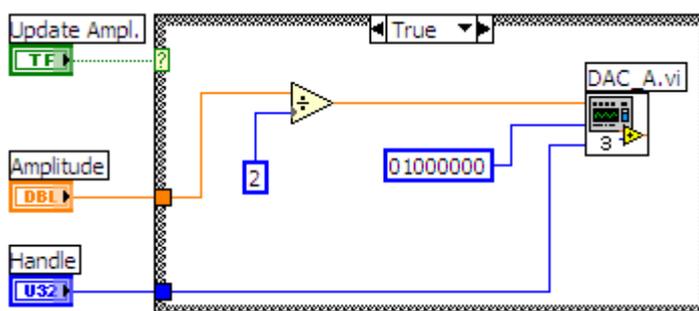


Figura 3.32 - Block diagram di "SigGenSetAmplitude.vi"

-Word in bit.vi

Questo sub-VI realizzato per una tesi precedente, è di fondamentale importanza nella realizzazione del generatore di segnali poiché consente la realizzazione del segnale da inviare effettivamente agli integrati AD9834 e DAC a partire dai dati che si vogliono trasmettere. In ambo i chip i dati sono caricati come una parola di 16 bit sotto il controllo del Serial Clock Input, generato dal Master (UM245R). Vediamo, quindi, il diagramma di temporizzazione per questa operazione in entrambi i circuiti integrati. Partiamo dall'analizzare quello dell'AD9834 mostrato in figura:

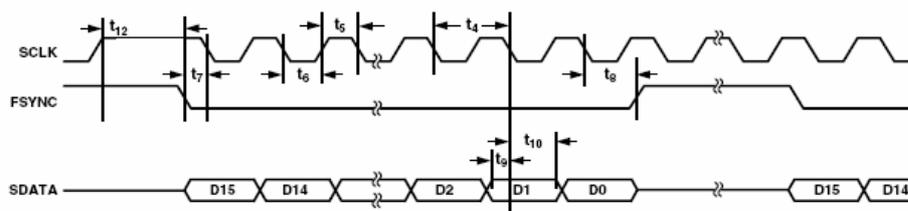


Figura 3.33 - Diagramma di temporizzazione del protocollo seriale dell'AD9834

Dopo che FSYNC diventa basso, i dati vengono spostati nel registro a scorrimento di input del dispositivo in corrispondenza dei bordi di discesa del SCLK per 16 impulsi di clock. Il passaggio del singolo bit lo si ottiene in due scritture poiché in una l'impulso di SCLK passa da 1 a 0, permettendo il trasferimento del bit nel dispositivo, nell'altra l'impulso di SCLK passa da 0 a 1 per consentire la trasmissione successiva. Esaminiamo ora il diagramma di temporizzazione del DAC mostrato in figura:

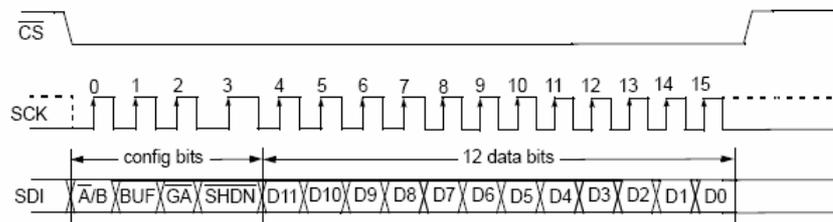


Figura 3.34 - Diagramma di temporizzazione del protocollo seriale del DAC

La scrittura seriale dei dati nel dispositivo inizia portando il \overline{CS} a livello logico basso; successivamente i dati saranno trasferiti nell'integrato in corrispondenza dei bordi di salita del SCK per 16 impulsi di clock. Come si può notare la scrittura dei dati nei due dispositivi è molto simile, fatta eccezione per il Serial Clock Input richiesto, infatti uno è l'inverso dell'altro. Si sono, dunque, realizzati due diversi VI per l'implementazione di quest'operazione: "Word in bit.vi", è quello usato per il DAC, l'altro, detto "Parola in bit per AD9834.vi", utilizza "Word in bit.vi" come Sub-VI e verrà esaminato successivamente.

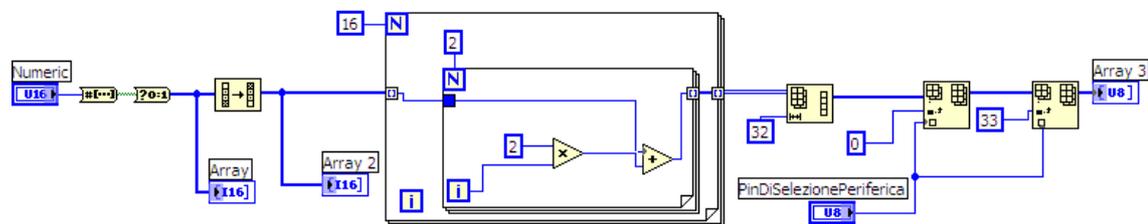


Figura 3.35 - Block Diagram di "Word in bit.vi"

Il controllore Numeric contiene i dati da inviare. L'array ottenuto dopo i cicli for, è ulteriormente modificato tramite la funzione "Insert Into Array", che permette

d’inserire un nuovo elemento nell’array nella posizione indicata. Vengono inseriti due nuovi elementi, uno in testa e l’altro in coda, in modo da abbassare ed alzare il CS ad inizio e fine trasmissione. Questa cosa si realizza settando opportunamente il controllore “PinDiSelezionePeriferica”, utilizzato per indicare quale integrato è attivo[7].

-Parola in bit per AD9834.vi

Realizzato per un tesi precedente, consente la creazione del segnale effettivamente inviato all’AD9834 a partire dai dati che si vogliono trasmettere; esso utilizza il “Word in bit.vi” poiché la modalità di scrittura dei dati nell’AD9834 e nel DAC sono simili, cambia solo il Serial Clock Input richiesto[7].

Rispetto alla versione esistente, è stata apportata una modifica all’array in uscita in modo da ottenere una comunicazione esclusiva tra master (UM245R) e slave (AD9834) senza interferire con gli altri integrati: è stato necessario aggiungervi il byte “01101100” con il quale vengono alzati tutti i CS relativi agli integrati, con esclusione di quello associato al DDS.

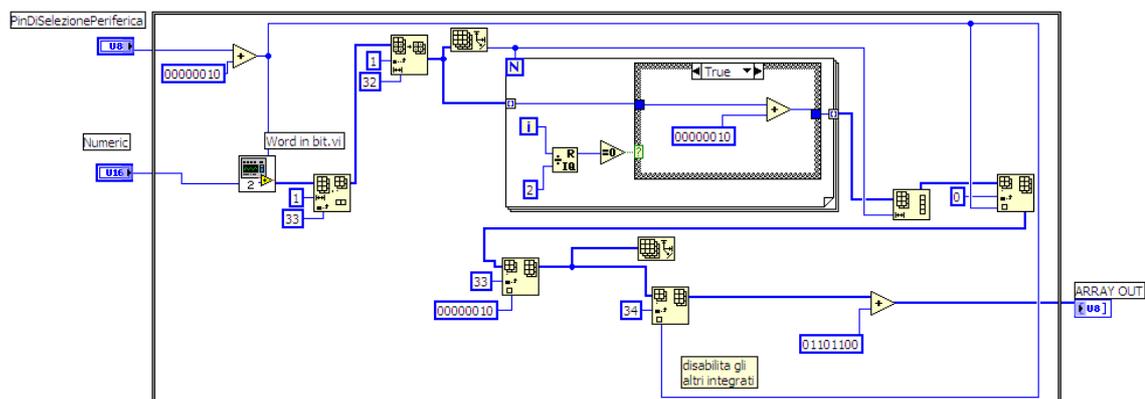


Figura 3.36 - Block diagram di “Parola in bit per AD9834.vi”

-ImpostazioneFreqReg.vi

Il sub-VI è stato realizzato per una tesi precedente e calcola il valore da caricare nel registro di frequenza selezionato dell’AD9834: nota la frequenza da generare, indicata con f_{OUT} e ricordando che l’output analogico dall’AD9834 è:

$$f_{out} = \frac{f_{MCLK}}{2^{28}} \cdot FREQREG$$

dove FREQREG è il valore caricato nel registro di frequenza selezionato e f_{MCLK} è la frequenza di MCLK, invertendo la formula otteniamo il valore da collocare nel registro:

$$FREQREG = \frac{f_{out} \cdot 2^{28}}{f_{MCLK}}$$

La sua implementazione non è altro che la realizzazione in LabVIEW della formula inversa[7].

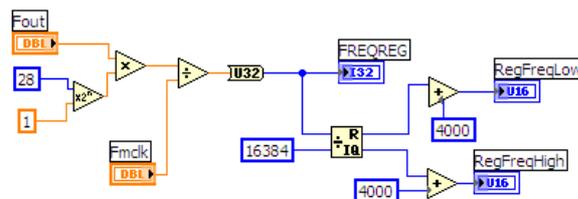


Figura 3.37 - Block diagram di “ImpostazioneFreqReg.vi”

-SendAD9834Frequenza.vi

Realizzato per una tesi precedente, effettua la trasmissione della f_{OUT} desiderata nell'integrato AD9834. La sua implementazione è data dalla concatenazione di due VI: “Parola in bit per AD9834.vi”, che realizza il segnale effettivamente inviato a partire dal dato che si vuole spedire, e “FT_Write_All_Data.vi”, che trasmette effettivamente il segnale così ottenuto. Poiché per trasmettere f_{OUT} bisogna effettuare due scritture consecutive, una per i bit meno significativi ed una per quelli più significativi, è stata utilizzata una Stacked Sequence Structure[7].

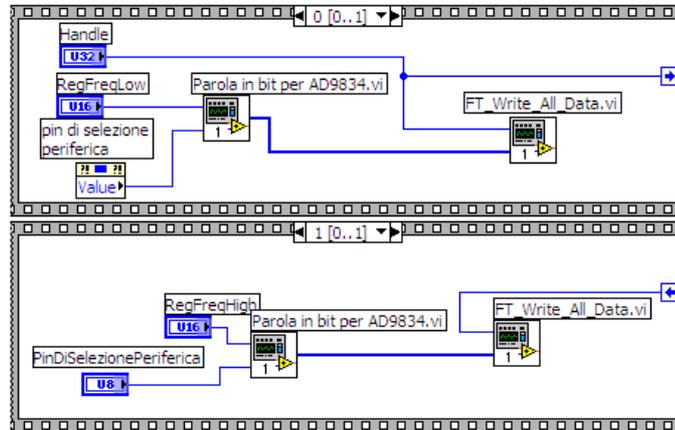


Figura 3.38 – Frame di “SendAD9834Frequenza.vi”

- ComandoFreqCompleto.vi

E' stato realizzato per riuscire ad avere un solo VI che piloti la trasmissione della f_{OUT} . Difatti la sua implementazione è data dalla concatenazione di due VI: “ImpostazioneFreqReg”, che calcola il valore da caricare nel registro di frequenza selezionato, nota f_{OUT} , e “SendAD9834Frequenza”, che ne effettua la trasmissione effettiva. Quindi avrà due soli controllori, uno in cui inserire la f_{OUT} voluta e l'altro in cui immettere f_{MCLK} [7].

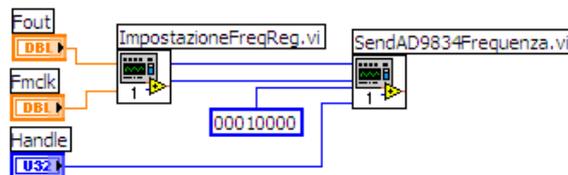


Figura 3.39 - Block diagram di “ComandoFreqCompleto.vi”

-SendAD9834Controllo.vi

Effettua la trasmissione della forma d'onda che si vuole generare all'AD9834. La sua implementazione è la stessa del “SendAD9834Frequenza”, l'unica cosa che cambia è il controllore che conterrà un valore atto a selezionare il tipo d'onda desiderata[7].

-DAC_A.vi e DAC_B.vi

Questi sub-VI realizzati per una tesi precedente, consentono di inviare al DAC del

generatore di segnali, il parametro di ampiezza e quello di offset della forma d'onda. La loro implementazione è data dalla concatenazione di due VI: “Word in bit” e “FT_Write_All_Data”, che effettua nella pratica l’invio del segnale generato. Ovviamente ci sono degli opportuni accorgimenti dovuti al principio di funzionamento del DAC: il valore posto in input viene moltiplicato per una costante pari a 400, ciò permette di convertire il range desiderato di ampiezza del segnale, nel range di valori che effettivamente il DAC può ricevere in input. Al valore ottenuto dalla moltiplicazione viene poi sommato una word costante rappresentante la configurazione di utilizzo del DAC così settata: 0101000000000000 per il DACA e 1101000000000000 per il DACB, per semplicità d’uso sono stati creati due vi separati: DAC_A.vi e DAC_B.vi contenenti le due differenti word[7].

Rispetto alla versione esistente, è stata apportata una modifica all’array in uscita in modo da ottenere una comunicazione esclusiva tra master (UM245R) e slave (DAC) senza interferire con gli altri integrati: è stato necessario aggiungergli il byte “00111100” con il quale vengono alzati tutti i CS relativi agli integrati, con esclusione di quello associato al DAC.

Il Block Diagram è mostrato in figura:

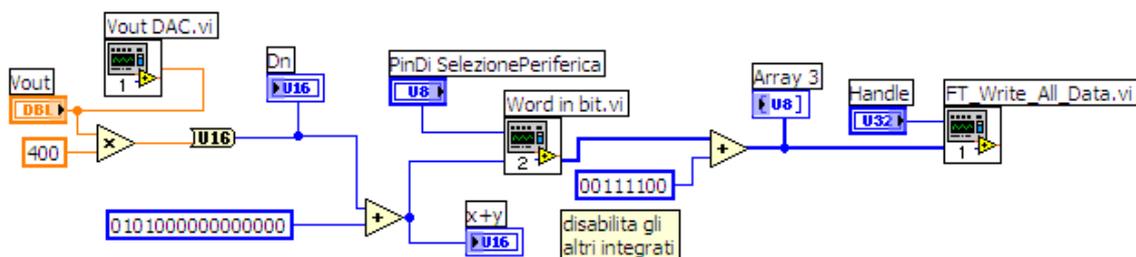


Figura 3.40 – Block diagram di “DAC_A.vi”

3.4.5 Altri sub-VI per il funzionamento del software

-Selectexperiment.vi

Questo sub-VI esegue nella pratica, la selezione del circuito di Chua o del circuito RLD⁸, andando a pilotare i multiplexer a relè; esso riceve in ingresso l'handle del dispositivo USB ed un byte in grado di far commutare opportunamente i relè. Il dato viene effettivamente scritto sulla porta B del port expander (quella connessa ai multiplexer), attraverso la funzione "WriteMCP23S". Il block diagram di questo sub-VI è mostrato in figura.

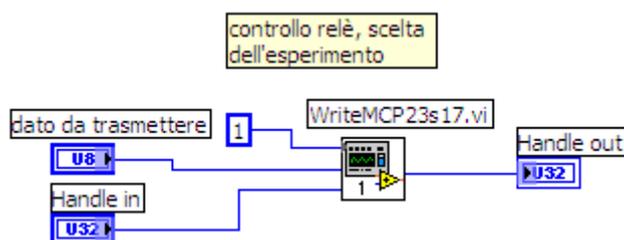


Figura 3.41 - Block diagram di "Selectexperiment.vi"

-Client info.vi

Questo sub-VI è di estrema importanza per il funzionamento da remoto del software di controllo.

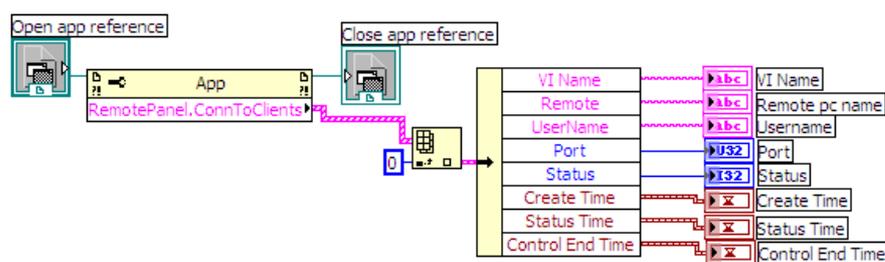


Figura 3.42 - Block diagram di "Client info.vi"

Attraverso un property node associato ad un refnum, si ottiene un array di cluster il quale contiene un insieme di informazioni sui clients, che da remoto, utilizzano gli

⁸ In futuro anche degli altri esperimenti.

esperimenti online. In realtà il laboratorio remoto consente l'accesso ad un solo utente per volta, per cui attraverso la funzione "Index Array" è stato sufficiente prelevare soltanto il primo elemento dell'array.

Tramite questo sotto-programma è possibile sfruttarne le informazioni fornite, per monitorare lo stato della connessione remota e, tra le altre cose, terminare l'esecuzione degli esperimenti qualora l'utente vi risulti disconnesso.

-Scalegraph.vi

Questo sub-VI è stato inserito all'interno del programma relativo al circuito RLD, sia per la versione da laboratorio sia per quella da remoto, in modo da ottenere un ridimensionamento automatico del diagramma delle fasi, a fronte di una variazione significativa dell'ampiezza del segnale d'ingresso. Quando si verifica tale variazione, il sub-VI viene mandato in esecuzione, restituendo dei valori tali da ottimizzare la visualizzazione del diagramma. In particolare se l'ampiezza è compresa tra 0,4 ed un valore $<1,5$ vengono restituiti i valori 2 e 1, se invece è compresa tra 1,5 ed un valore $<2,76$, vengono restituiti i valori 3 ed 1, e così via.

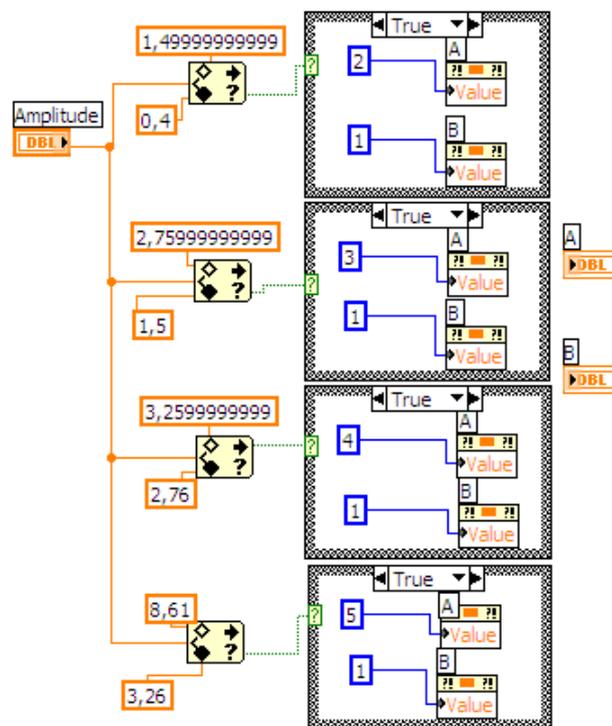


Figura 3.42 - Block diagram di "Scalegraph.vi"

Capitolo 4

REMOTIZZAZIONE WEB

DEGLI ESPERIMENTI

L'idea stessa di poter usufruire di risorse in "remoto" nasce dal concetto chiave di non poterle utilizzare in "locale", ovvero di non essere presenti nel luogo ove queste hanno collocazione fisica.

Oggigiorno l'avvento delle nuove tecnologie su reti telematiche, dell'utilizzo di Internet su scala globale, e la massiccia presenza dei Personal Computer nelle nostre case, ha permesso tutto ciò anche in maniera piuttosto semplice.

Se si pensa alla posta elettronica, all'e-commerce o alla miriade di siti web presenti in rete, si può facilmente comprendere come sia facile interagire da "remoto" in molte situazioni "digitali" che si presentino ai nostri occhi, ed è testimonianza del fatto che un vastissimo numero di applicazioni (tra le quali fa capolino LabVIEW) basate su piattaforme Windows o Macintosh, stiano sviluppando in maniera via via crescente questa funzionalità.

La possibilità di eseguire esperimenti da remoto, accresce in maniera considerevole le opportunità di fare didattica sperimentale sui circuiti, poiché in tal modo si garantisce la disponibilità degli esperimenti durante tutto l'arco della giornata, senza i vincoli propri delle dimostrazioni in aula o in laboratorio.

In questo capitolo verranno descritte le procedure per pubblicare online il software di controllo degli esperimenti ed il sito web attraverso il quale potervi accedere, con dei cenni sul linguaggio di programmazione utilizzato per realizzarlo, l'HTML.

4.1 Cenni sul linguaggio HTML

L'HTML è il linguaggio con cui è possibile indicare come i vari elementi vanno disposti in una pagina Web. Un documento html non è nient'altro infatti che un file di testo con delle indicazioni sul colore delle scritte, sulla posizione delle immagini all'interno della pagina, su come far scorrere il testo, e altre cose di questo genere.

HTML è l'acronimo di Hypertext Markup Language ("Linguaggio di contrassegno per gli Iper testi") e non è un linguaggio di programmazione (sono linguaggi di programmazione il C, il C++, il Pascal, il Java, e sono linguaggi di scripting il PHP, l'ASP, il PERL, il JavaScript).

Si tratta invece di un linguaggio di contrassegno (o 'di marcatura'), che permette di indicare come disporre gli elementi all'interno di una pagina: le indicazioni vengono date attraverso degli appositi marcatori, detti "tag".

Ciò significa che l'HTML non ha meccanismi che consentono di prendere delle decisioni ("in questa situazione fai questo, in quest'altra fai quest'altro"), e non è in grado di compiere delle iterazioni ("ripeti questa cosa, finché non succede questo"), né ha altri costrutti propri della programmazione.

Il linguaggio HTML, pur essendo dotato di una sua sintassi, non presuppone la logica ferrea e inappuntabile dei linguaggi di programmazione: se si dimentica di chiudere un tag, non verranno prodotti messaggi di errore; se non si rispetta la sintassi probabilmente non si otterrà la visualizzazione della pagina desiderata, ma nient'altro.

Il Browser è il programma necessario per navigare nel Web e svolge principalmente due compiti:

- scarica i vari files che si trovano su un computer remoto (il server) e che fanno riferimento a un certo indirizzo;
- legge i documenti scritti in html, e a seconda delle indicazioni ivi contenute, visualizza la pagina in un modo, piuttosto che in un altro; inoltre i vari files

associati a quel documento (ad esempio le immagini, o i filmati in flash) vengono disposti secondo le indicazioni del codice html.

La visualizzazione di un file html da parte del browser prende il nome di rendering della pagina. Motore di rendering è dunque quella sezione del browser che si occupa di mostrare sul video la pagina.[10]

4.2 Il sito web per il laboratorio remoto di circuiti

La versione precedente del laboratorio remoto, prevedeva un unico esperimento, e disponeva di un sito web in cui ne venivano trattati aspetti teorici e realizzativi; considerato l'attuale scenario, in cui si hanno a disposizione più esperimenti, si è pensato di realizzare un nuovo sito web raggiungibile da chiunque fosse interessato, aggiornato con le nuove tematiche, e che conducesse al software descritto nel precedente capitolo.

4.2.1 Struttura del sito

Esso prevede in alto un'intestazione, cui aspetto grafico è in linea con quello del sito del dipartimento d'ingegneria elettrica, sulla sinistra un menù "a tendina" con alcuni collegamenti ipertestuali che indirizzano l'utente verso le diverse pagine, ed un corpo centrale in cui sono inseriti i contenuti.

Il menù laterale espanso in tutte le sue sezioni, presenta una struttura di questo tipo:

HOME PAGE

Il laboratorio remoto

+Generalità

+Dispositivi hardware

-Schede d'acquisizione

-Generatore di segnali

Esperimenti disponibili

+Circuito di Chua

+Circuito RLD

Vai agli esperimenti

+Requisiti minimi

+Simulazione

In figura è mostrato il menù a tendina “aperto”:



Figura 4.1 - Menù a tendina del sito

Ciascun link del menù è abbinato ad immagini diverse per evidenziarne lo stato di attivazione.

4.2.2 Realizzazione delle pagine in HTML

Per la realizzazione della pagine html è stato utilizzato l’editor di testo “blocco note” presente in tutti i sistemi operativi Microsoft[®]. Per prima cosa è stata importata la grafica del sito <http://diel.unina.it>, prelevandone il codice html; successivamente per ciascuna pagina creata, sono stati aggiunti i link del menù laterale ed i contenuti nel corpo centrale. In alto nel corpo centrale è stato inoltre replicato il menù laterale, attraverso dei link distribuiti in linea orizzontale.

4.2.3 I tag HTML utilizzati per la realizzazione del sito

Per la realizzazione del sito web sono stati utilizzati alcuni tra i più comuni tag HTML:

- Il tag `
` inserisce una interruzione di riga; essendo un tag vuoto, è obbligatorio scriverlo come `
`;
- Il tag `<p>` definisce un paragrafo di testo e lascia uno spazio prima e dopo la propria chiusura;
- Il tag `<a>` definisce un ancora. Può essere usata sia per creare dei link esterni che dei collegamenti all'interno della pagina;
- Il tag `` definisce un'immagine; essendo un tag senza contenuto va chiuso con il carattere di /;
- Il tag `` permette di definire la formattazione dei caratteri;
- Il tag `` indica del testo in grassetto;
- Il tag `<i>` indica del testo in corsivo;
- Il tag `<u>` definisce del testo sottolineato;
- Il tag `<center>` centra orizzontalmente il testo incluso;
- Il tag `` definisce una lista puntata;
- Il tag `` definisce un elemento di una lista[11].

I sito nella sua versione finale è mostrato in figura 4.2:

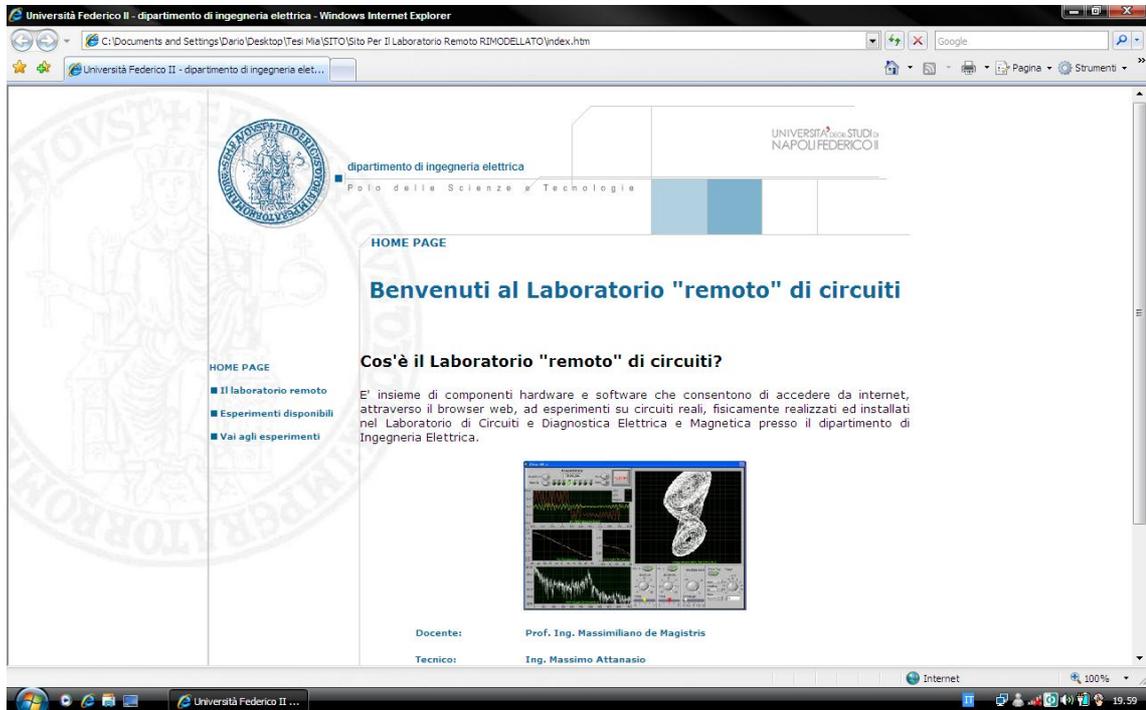


Figura 4.2 - Home page del sito

Esso è stato successivamente installato sul server del dipartimento, ed è attualmente disponibile all'indirizzo <http://www.diel.unina.it/esperimenti/index.htm>

4.3 Remotizzazione del software di controllo

4.3.1 Procedura per la remotizzazione

Il passaggio finale dell'esperimento remoto, presente sul web, corrisponde con un interfaccia grafica di una pagina con al suo interno il programma principale descritto nel capitolo precedente, che da accesso ai diversi esperimenti. Questa garantisce il controllo da remoto, e rappresenta la schermata LabVIEW vera e propria con la quale è possibile il controllo di pulsanti e indicatori.

È molto interessante capire come questa pagina sia stata realizzata: tra le peculiarità del linguaggio LabVIEW esiste anche un "Tool di pubblicazione grafica via web", che attraverso pochi passaggi permette di realizzare il tutto[9].

Questa funzionalità, presente a partire dalle ultime versioni, è stata integrata insieme ad altre di maggiore utilizzo più tipiche del software stesso. Dopo l'apertura del file da remotizzare, è possibile accedere a questa funzione selezionando la voce *Web Publishing Tool* dal menù *Tool*, come mostrato in figura.

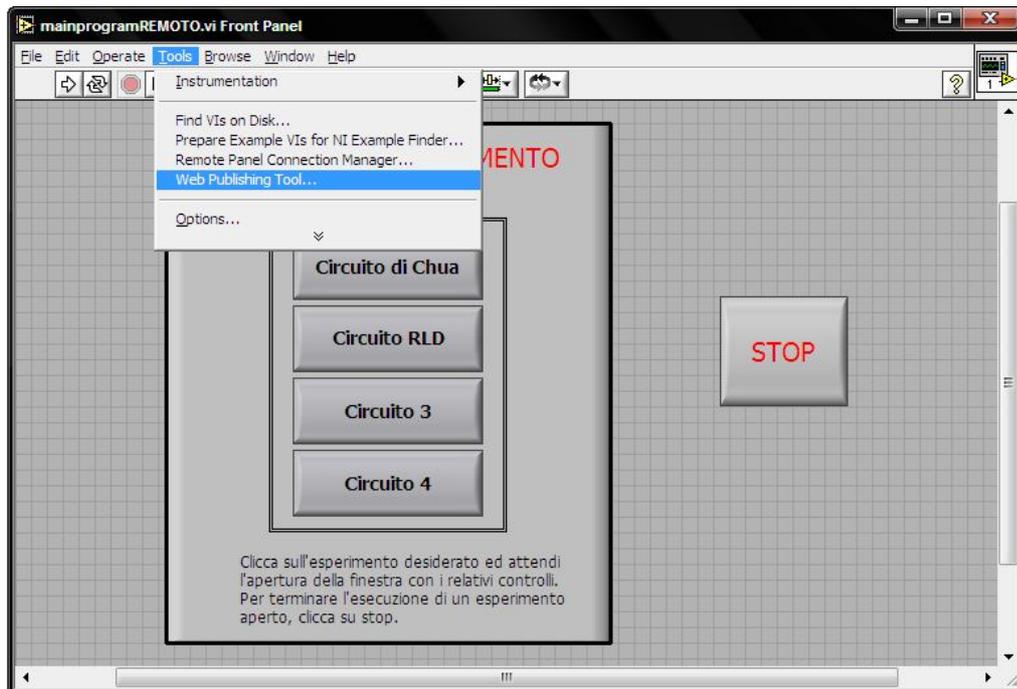


Figura4.3 - Web publishing tool dal menù "Tool" di Labview

All'apertura della finestra (in figura 4.4) si ha accesso a tutte le opzioni che si vogliono impostare per la remotizzazione.

Tra queste si possono settare :

- Il nome della pagina (il documento HTML che verrà salvato sull'hard disk della macchina server) ;
- Un "header" testuale con possibili spiegazioni;
- Il nome del file con estensione *.vi;
- Un "footer" testuale con possibili conclusioni a piè di pagina;
- Le opzioni di visualizzazione e del controllo remoto da passare alla macchina

client (ovvero al possibile fruitore del software remoto).

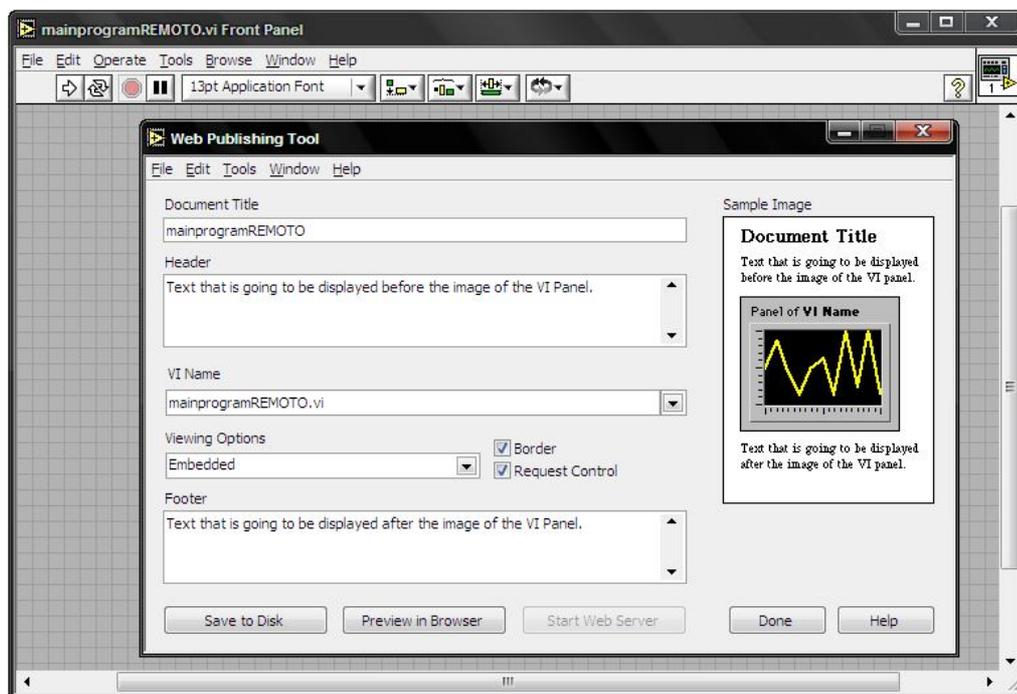


Figura 4.4 - Web publishing tool di Labview

A questo punto in fondo alla finestra compaiono tre pulsanti attraverso i quali è possibile salvare la pagina in un file in formato *.html sul disco fisso in una cartella condivisa predefinita da LabVIEW, visionare una sua anteprima attraverso il browser di navigazione installato (Internet Explorer, Opera, Mozilla, ecc.), e far partire il tutto come se fosse un Web Server.

A valle di queste operazioni è necessario inoltre adottare delle misure di sicurezza, per mostrare da remoto soltanto il programma principale (e di conseguenza gli esperimenti in esso contenuti) “nascondendo” qualsiasi altro VI, presente sul PC server, che non si desidera fare vedere. A tal proposito è sufficiente selezionare la voce *Web Server: Visible VIs* dal menu *Options* di Labview, ed aggiungere alla lista il nome del file da voler mostrare in remoto, così come illustrato in figura 4.5:

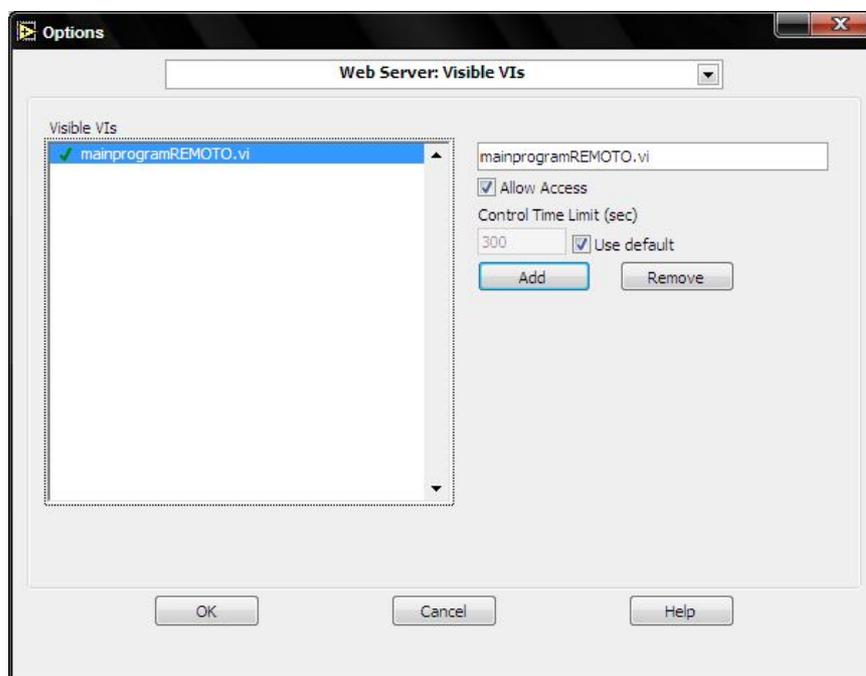


Figura 4.5 - Impostazioni di visibilità di “mainprogramREMOTO.vi”

Come ultimo step della procedura di remotizzazione, è stato inserito il programma principale per l’accesso agli esperimenti, all’interno del sito web, sfruttando un frammento di codice presente nel precedente sito del laboratorio remoto. Questo codice è stato importato all’interno della pagina web di accesso agli esperimenti (goexperiment.htm) del nuovo sito, e consente la visualizzazione del pannello frontale di labview come se fosse parte integrante della pagina stessa.

Lo script che realizza tale funzione richiede il nome del VI da aprire (mainprogramREMOTO.vi), e l’indirizzo IP del server su cui il VI è stato salvato (server del laboratorio); inoltre bisogna specificare le dimensioni dell’area della pagina web, all’interno della quale si desidera visualizzare il pannello frontale; Proprio questa operazione ci consente di mostrare all’utente soltanto un’area del pannello frontale, escludendone altre: in particolare è stata dimensionata quest’area in modo tale da mostrare solo i pulsanti per la selezione degli esperimenti, nascondendo il tasto stop⁹.

⁹ Che continuerà ad essere visibile ed utilizzabile dal PC server del laboratorio.

Come si è visto la procedura è abbastanza semplice ed automatizzata. Naturalmente tutti i passaggi sopra elencati sono solo a scopo descrittivo. Infatti chiunque voglia accedere al sito e alla pagina in questione non avrà bisogno di conoscerne tale realizzazione, ma avendo già installato sul proprio PC il software LabVIEW oppure il pacchetto “Labview Runtime Engine”, una volta cliccato sul link di riferimento, scaricherà dal sito della National Instrument, in automatico, le librerie necessarie al controllo remoto e potrà accedere agli esperimenti senza altre preoccupazioni. Unico limite all’utilizzo degli esperimenti online, è rappresentato dal fatto che si tratta di un laboratorio remoto “monoutente”, vale a dire che consente l’accesso ad un solo utente per volta: ciò è dovuto sia ad una mancanza di ulteriori dispositivi hardware, sia alla licenza Labview, la quale è predisposta per un solo utente, e pertanto necessiterebbe di un upgrade. Il risultato finale è mostrato in figura 4.6.

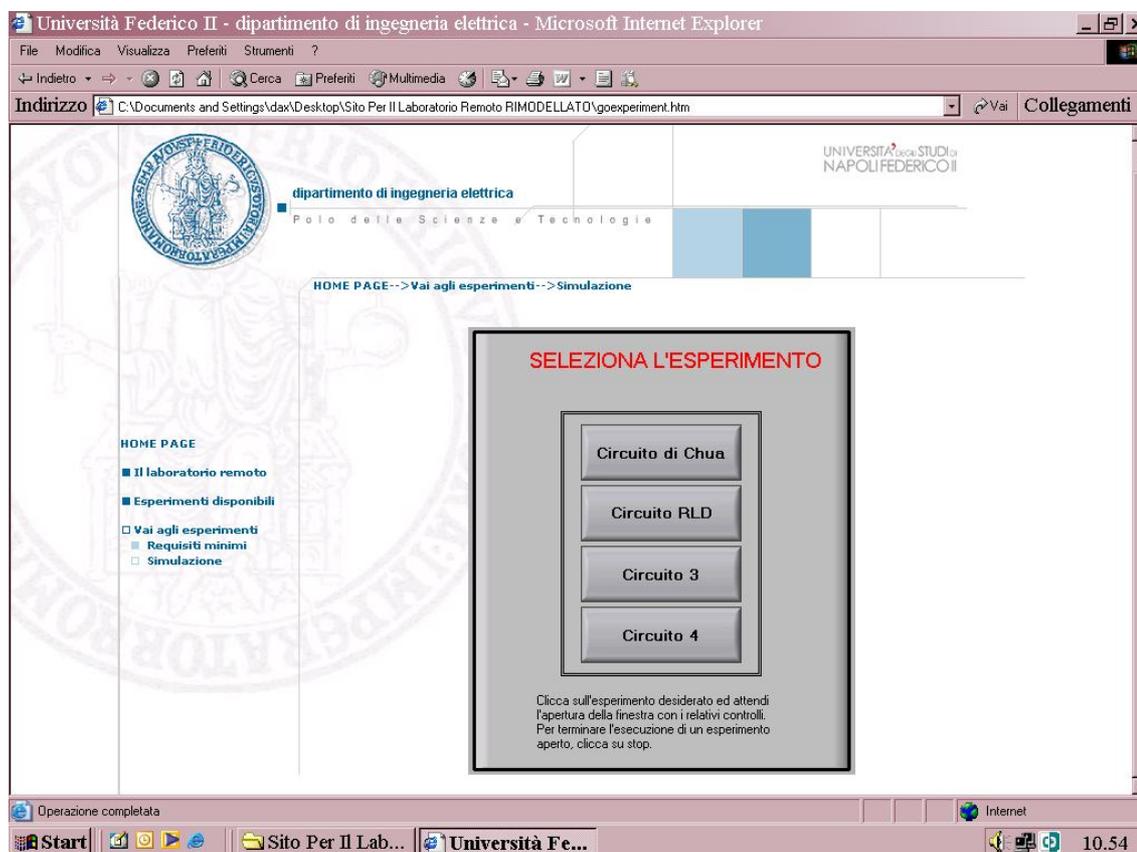


Figura 4.6 - Pagina del sito web per l’accesso agli esperimenti.

CONCLUSIONI

Con il lavoro descritto in questo elaborato, si può davvero sostenere che il progetto del laboratorio per la didattica sperimentale sui circuiti, sia in aula che da remoto, sia ormai entrato nella sua fase operativa; sono stati infatti raggiunti in maniera soddisfacente gli obiettivi previsti, ovvero di realizzare un software per la selezione degli esperimenti, e in grado di farli funzionare garantendone un certo grado di robustezza e autonomia. Il sito web realizzato risponde alle esigenze di funzionalità e chiarezza richieste, e fornisce adeguate informazioni sulle tematiche affrontate.

Il laboratorio di circuiti è però un progetto in costante sviluppo, sono infatti in corso di realizzazione ulteriori esperimenti didattici, che andranno ad aggiungersi a quelli già a disposizione.

Altro aspetto da poter sviluppare in futuro, riguarda quello relativo alla *multiutenza*, la quale però prevede la realizzazione di altro hardware identico a quello attualmente disponibile. Questo progetto certamente non impiegherà tempi brevissimi, soprattutto se s'intende aumentare di n volte il numero di utenti contemporaneamente collegabili. In alternativa si potrebbe quindi pensare all'ingegnerizzazione degli apparati realizzati, magari con la realizzazione di pacchetti hardware/software da fornire a scuole, università o a chiunque fosse interessato, con l'opportunità di trasformare un progetto, in una realtà affermata nel settore della didattica sui circuiti.

BIBLIOGRAFIA

- [1] **G. Amato.** *Realizzazione di un circuito caotico di Chua*, (Tesi di laurea);
- [2] **G. Mirra.** *Realizzazione di una scheda di input/output analogica/digitale per il laboratorio "remoto" di circuiti*, (Tesi di laurea);
- [3] www.wikipedia.it;
- [4] **M. Colandrea.** *Realizzazione di un dimostratore didattico per circuiti caotici*, (Tesi di Laurea);
- [5] **Bertocco.** *"Introduzione al LabView" Università di Padova Facoltà di Ingegneria*;
- [6] <http://www.zone.ni.com>;
- [7] **G. Cimminiello.** *"Realizzazione e caratterizzazione di un generatore di segnale amplificato USB"*, (Tesi di laurea);
- [8] **V. Di Fiore.** *"Realizzazione e testing di hardware per la didattica sperimentale sui circuiti"*, (Tesi di laurea);
- [9] **D. Acanfora.** *"Remotizzazione web di esperimento sul circuito caotico di Chua"*, (Tesi di laurea);
- [10] <http://xhtml.html.it>;
- [11] <http://www.morpheusweb.it>;
- [12] **FTDI Chi.** *UM245R Datasheet*;
- [13] **Microchip Technology.** *MCP23S17 Datasheet*;
- [14] **M. de Magistris, G. Miano.** *Fondamenti di circuiti per l'ingegneria*;
- [15] **L.O. Chua, R.N. Madan.** *Sights and Sounds of Chaos*.