

**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 DI UN GENERATORE
DI SEGNALE PORTATILE
CONTROLLATO VIA USB**

RELATORE

CH.MO PROF. ING.

MASSIMILIANO DE MAGISTRIS

CORRELATORE

ING. MASSIMO ATTANASIO

CANDIDATO

ANTONIO ROSA

MATR. 540/467

ANNO ACCADEMICO 2005/2006

*Ai miei genitori Ugo ed Annamaria
che mi hanno dato la possibilità di studiare e
che in questo lungo percorso universitario
mi hanno sostenuto con il loro affetto
Alla mia dolce metà Francesca
che col suo amore mi ha aiutato a superare
i momenti difficili che si sono presentati
in questi anni di Università e
che mi ha sempre rassicurato prima di ogni esame*

Ringraziamenti

Il primo, più sentito e sincero ringraziamento va al Prof. Massimiliano de Magistris, relatore della presente tesi, per la professionalità, la disponibilità e la pazienza avuta nei miei confronti. Ringrazio l'Ing. Massimo Attanasio per la simpatia e per il sostegno offertomi nel laboratorio di elettrotecnica e l'Ing. Marco Colandrea per l'aiuto, i consigli e la competenza dimostratomi.

Grazie ai miei genitori che con tanti sacrifici ed amore mi hanno appoggiato in questi anni universitari; insegnandomi che nella vita non bisogna mai arrendersi alle prime difficoltà, ma bisogna sempre lottare per realizzare i propri sogni.

Grazie a mio fratello Stefano che, nonostante tutto, col suo modo di fare, mi è sempre stato accanto, sostenendomi e sopportandomi, specie quando incasinavo il computer.

Grazie al mio amore. Tu che, forse più di chiunque altro, mi hai sopportato in questa lunga carriera universitaria, aiutandomi a superare i momenti difficili, le mie insicurezze e tutti i miei dubbi che ho incontrato in questi anni. Tu che hai gioito con me dei miei successi e che hai sdrammatizzato le mie disfatte. Tu che hai sempre trovato le parole giuste per infondermi coraggio prima di ogni esame. Spero che oggi Tu sia fiera di me e se ho raggiunto questo obiettivo è anche merito tuo.

Grazie a tutti i miei parenti: nonne, zii, zie, cugini e cugine che hanno sempre creduto in me, infondendomi una gran fiducia nelle mie possibilità.

Grazie ai miei nonni che da la su, insieme a tanti angeli, vegliano su di me.

Grazie alla Signora Renata che con il suo affetto ha saputo confortarmi nei momento difficili e rallegrarsi dei miei successi.

Grazie ad Alessia, la mia cara cognata, che, tra un battibecco e l'altro, mi è corsa sempre in aiuto quando ne avevo bisogno, incoraggiandomi prima di ogni esame e prendendomi in giro ogni qualvolta prendevo un buon voto.

Grazie ai miei suoceri, Nunzio ed Alessandra, per aver creduto in me e per l'affetto mostratomi in questi anni.

Grazie alla Signora Elisa per l'amorevolezza con cui ogni giorno mi chiedeva quando sarebbe arrivato questo giorno e con cui mi prepara il vero caffè napoletano.

Grazie a Giovanni e Roberto, i miei due cognatini, per il sostegno datomi.

Grazie ai miei amici di vecchia data Umberto e Luca e alle loro famiglie per essermi stato accanto in questi anni universitari con tanta amorevolezza.

Grazie a Chiara, la mia cara amica, che con la sua simpatia ed il suo affetto mi ha aiutato a superare i cosiddetti momenti no, facendomi tornare subito il sorriso sulle labbra.

Grazie ad Enza, Teresa, Francesco, Olimpia, Raffaella, Lisa, Rosario, Francesco ed Enza per i bei momenti trascorsi insieme che hanno reso più piacevoli le lunghe ore di studio e non, per l'affetto sincero mostrato nei miei confronti, per il sostegno e l'incoraggiamento prima di ogni esame. Insomma per tutto vi dico ancora grazie.

Grazie a Francesco, Giuseppe, Valentina e Andrea per l'essermi stati accanto nei momenti felici e no di questi anni.

Grazie a tutti coloro che mi vogliono bene.



INDICE

<i>Introduzione</i>	pag. 8
---------------------------	--------

Capitolo 1

<i>DIDATTICA SPERIMENTALE PER I CIRCUITI</i>	pag. 11
---	---------

<i>1.1 Apparatì sperimentali per la didattica</i>	pag. 11
--	---------

<i>1.2 Strutture per la didattica sperimentale</i>	pag. 12
---	---------

<i>1.3 Dispositivi per la didattica in aula</i>	pag. 13
--	---------

<i>1.3.1 Schede di acquisizione</i>	pag. 14
--	---------

<i>1.3.2 Interfaccia USB</i>	pag. 15
---	---------

<i>1.3.3 USB/GPIB</i>	pag. 16
------------------------------------	---------

<i>1.3.4 Generatori dedicati</i>	pag. 17
---	---------

<i>1.4 Generatore di segnale didattico</i>	
---	--

<i>sperimentale</i>	pag. 17
----------------------------------	---------

<i>1.4.1 Cos'è un generatore di segnale</i>	pag. 18
--	---------

<i>1.4.2 Motivazioni della realizzazione di un generatore di segnale didattico sperimentale</i>	pag. 18
--	---------

<i>1.4.3 Funzioni del generatore realizzato</i>	pag. 19
--	---------



1.5 Alcuni circuiti per la didattica

sperimentale.....pag. 20

1.5.1 Carica e scarica del

condensatore/induttore.....pag. 20

1.5.2 Circuito del primo ordine con

forzamento sinusoidale.....pag. 21

1.5.3 Circuito lineare con forzamento periodico...pag. 21

1.5.4 Circuiti risonanti.....pag. 22

1.5.5 Circuito caotico RLD.....pag. 24

1.5.6 Circuito caotico di Chua.....pag. 25

Capitolo 2

PROGETTAZIONE E REALIZZAZIONE DI UN GENERATORE DI SEGNALE

USB PORTATILE.....pag. 27

2.1 Analisi dello stato dell'arte.....pag. 27

2.2 Analisi degli integrati disponibili.....pag. 31

2.3 Il circuito integrato AD9834.....pag. 34

2.3.1 Descrizione generale.....pag. 34

2.3.2 Descrizione dei pin.....pag. 34

2.3.3 Descrizione dei circuiti interni all'AD9834...pag. 36

2.3.4 Descrizione funzionale.....pag. 39

2.4 Altri dispositivi per la realizzazione.....pag. 42

2.4.1 Il modulo UM245R.....pag. 42



2.4.2 Il dispositivo DAC.....	pag. 47
2.4.3 Il moltiplicatore.....	pag. 49
2.5 Collegamento dei moduli.....	pag. 52
2.6 Risultati.....	pag. 57

Capitolo 3

SOFTWARE PER IL PILOTAGGIO E CONTROLLO DEL GENERATORE VIA USB.....

pag. 62

3.1 Breve premessa.....	pag. 62
3.2 Introduzione al LabVIEW.....	pag. 62
3.2.1 Cos'è il LabVIEW?.....	pag. 63
3.2.2 Utilizzo principale di LabVIEW.....	pag. 65
3.2.3 Virtual Instrument (VI).....	pag. 65
3.3 Funzioni di pilotaggio dei vari moduli.....	pag. 67
3.3.1 VI : PilotUSB.....	pag. 68
3.3.2 VI : Word in bit.....	pag. 71
3.3.3 VI : ImpostazioneFreqReg.....	pag. 77
3.3.4 VI : Parola in bit per AD9834.....	pag. 79
3.3.5 VI : SendAD9834Frequenza.....	pag. 81
3.3.6 VI : ComandoFreqcompleto.....	pag. 83
3.3.7 VI : SendAD9834Controllo.....	pag. 85
3.3.8 VI : DAC.....	pag. 86
3.4 Pannello di controllo del generatore.....	pag. 89



3.5 Conclusioni.....	pag. 91
3.6 Sviluppi futuri.....	pag. 92
<i>Bibliografia</i>.....	pag. 93

INTRODUZIONE

Le attività di tipo sperimentale hanno un ruolo basilare nella comprensione delle nozioni tecnico-scientifiche; infatti affiancare ai concetti teorici anche un riscontro pratico ne consente uno intendimento maggiore. Per tale motivo alcuni corsi di laurea completano il curriculum degli studenti con esami prettamente di laboratorio, consentendo allo studente di mettere in pratica i concetti teorici studiati tramite la realizzazione di esperimenti più o meno semplici, ma anche di prendere confidenza con la strumentazione di laboratorio.

Le ore di studio teorico possono essere completate con interessanti esperimenti di laboratorio che, sintetizzando efficacemente ciò che si è sviluppato in teoria, ne permettono in molti casi una migliore comprensione; ciò è ancor più vero in quelle situazioni in cui l’abitudine a certe metodologie risolutive o a certi comportamenti dei sistemi porta, per estrapolazione, a conclusioni intuitive contrastanti con l’effettiva realtà fisica.

In tal contesto molto importante è stato l’avvento dei calcolatori elettronici che ha portato allo sviluppo di programmi di calcolo, di simulazione e di modellazione dei sistemi. Tramite questi programmi è stato possibile realizzare esperienze di laboratorio virtuali da affiancare allo studio teorico.

Le parole chiave di questo genere di simulazioni sono: capacità di soluzione delle equazioni differenziali nel caso dei programmi di calcolo, capacità di determinare l’evoluzione di un sistema nel caso dei programmi di simulazione. Ed inoltre, comune a tutte queste

categorie di programmi, capacità grafiche per visualizzare in maniera sintetica, ma espressiva, i risultati nelle elaborazioni, realizzando, per esempio, semplici animazioni del sistema fisico simulato. In questa maniera si riesce ad offrire una simulazione che non abbia come risultati semplici tabelle o grafici, ma che mostri il sistema fisico come si presenta nella realtà.

Sebbene gli esperimenti virtuali possono essere di grande aiuto nella comprensione delle materie scientifiche, mancano comunque di realtà, lasciando sempre una sensazione di non totale veridicità dei risultati ottenuti. Per ovviare a ciò l'unica possibilità è realizzare esperimenti reali con oggetti fisicamente esistenti. Purtroppo alcuni esperimenti, per non dire molti, necessitano di attrezzature non trasportabili e quindi disponibili esclusivamente nei laboratori a scapito della didattica sperimentale.

Proprio in questo ambito si inserisce il lavoro esposto in questa tesi, il cui scopo è quello di realizzare un generatore di segnale integrato portatile controllato via USB , da utilizzare per scopi didattici in esperimenti in aula. Difatti in commercio non sono disponibili generatori dal costo non elevato, dalle dimensioni e peso trascurabili ed interfacciabili col calcolatore.

L'intero lavoro proposto è organizzato in tre parti distinte.

La **prima parte**, indicando i dispositivi che ad oggi possono esser utilizzati per le didattica sperimentale, spiega le motivazioni che ci hanno spinto alla realizzazione di un generatore di segnali ad hoc.

La **seconda parte** analizza i vari circuiti integrati necessari alla realizzazione del generatore di segnali e l'interconnessione degli stessi.

La **terza parte** ed ultima parte, di carattere prevalentemente implementativo, descrive il funzionamento dei programmi realizzati per pilotare i vari integrati adoperati mediante software LabVIEW, presentando, infine, l’interfaccia d’utilizzo del generatore di segnale.

Capitolo 1

DIDATTICA SPERIMENTALE PER I 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 permetterebbe 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 completamento allo studio teorico, e di consentire l'acquisizione da parte degli studenti di una maggiore dimestichezza, anche pratica, con gli strumenti elettronici.

1.1 Apparati sperimentali per la didattica

Un *apparato sperimentale* è costituito da un insieme di dispositivi e/o elementi funzionali connessi tra loro che servono ad un unico scopo. Può essere realizzato utilizzando le risorse di un personal computer

portatile in unione a componentistica aggiuntiva di piccola dimensione.

Gli apparati sperimentali possono esser utilizzati per la realizzazione di sistemi prototipali che mostrano l’evolversi di meri esperimenti. Infatti possono essere realizzati per sostituirsi ad elementi del sistema non disponibili oppure per concretizzare un progetto.

Un *apparato sperimentale* applicabile alla didattica sperimentale dei circuiti elettrici, da un punto di vista logico, può esser diviso in tre parti:

- Intelligenza di controllo, ovvero il dispositivo che controlla gli elementi costituenti l’apparato, come un personal computer portatile.
- Circuito d’interfaccia, vale a dire l’elemento che permette di collegare l’intelligenza di controllo con il circuito sotto test, come il generatore realizzato unito alla scheda di acquisizione del PC portatile.
- Circuito sotto test, cioè il circuito di cui si vuole analizzare la dinamica, come un circuito RLD.

1.2 Strutture per la didattica sperimentale

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 classi, soprattutto dei primi anni. Ciò determina che i laboratori non sono disponibili per un uso generale delle classi di studenti, quindi gli universitari non possono usufruire a pieno ed in condizioni ottimali degli strumenti convenzionali. Tale problematica

determina, inoltre, la necessità di realizzare in altro modo le attività sperimentali.

L'impossibilità di realizzare queste attività in laboratorio ha portato a voler spostare l'esperimento in aula. Questo comporta la perdita della disponibilità della strumentazione più convenzionale, in quanto le dimensioni e il peso sono tali da non consentirne il trasporto al di fuori del laboratorio. Da ciò 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 direttamente interfacciabili col calcolatore perché in tal modo è possibile anche realizzare una proiezione su di uno schermo dell'evoluzione dell'attività sperimentale e controllarli direttamente dal computer senza doversi concentrare su più oggetti contemporaneamente.

Non potendo usufruire della strumentazione convenzionale al di fuori di un laboratorio per la realizzazione delle attività di tipo sperimentali, si è pensato di poterla utilizzare in “remoto”. Pertanto creare un esperimento in “remoto” accessibile via web, del quale studenti e persone interessate possano servirsi, per poter accedere ai risultati di testing su apparati sperimentali presenti in laboratorio, che altrimenti potrebbero essere visionati solamente in “locale” (rif. bibl. 3).

1.3 Dispositivi per la didattica in aula

Oggigiorno ci sono diversi dispositivi disponibili per la realizzazione della didattica in aula, come:

- Schede di acquisizione.
- Interfaccia USB.

- USB/GPIB.

Vediamone gli aspetti salienti.

1.3.1 Schede di acquisizione

La *scheda di acquisizione* è un dispositivo hardware che consente la raccolta automatizzata delle informazioni da sorgenti di misura analogica e digitale. Ovvero, essa comporta la raccolta di segnali da sorgenti di misura e la digitalizzazione del segnale per immagazzinamento, analisi e presentazione su PC. Il suo compito è quello di trasferire nel modo più adatto e performante i segnali esterni sul PC, in modo tale da poterli riprodurre ed elaborare.

In particolare essa si adopera a “catturare” il segnale analogico/digitale uscente 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.

Una scheda di acquisizione può presentare anche delle uscite controllate tramite degli appositi algoritmi (rif. bibl. 2).

Il suo impiego, nella didattica, è relativo alla possibilità di fornire al PC portatile i dati da esaminare, poiché si collega al personal computer tramite l'interfaccia PCMCIA, normalmente presente su tutti i personal computer portatili.

Esempi di schede di acquisizione sono mostrati nella figura 1.1:

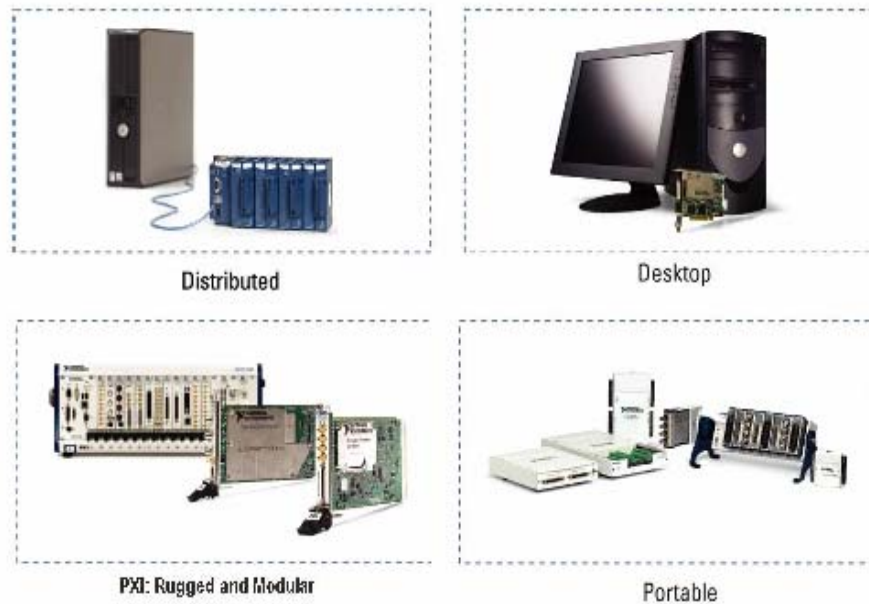


Figura 1.1: Schede di acquisizione

1.3.2 Interfaccia USB

L'USB è uno standard di comunicazione seriale che consente di collegare diverse periferiche ad un computer (rif. bibl. 21).

Sta sostituendo le uscite presenti “storicamente” sul PC, come porte parallele o seriali, e in parte anche il BUS interno con un BUS esterno per periferiche di tipo digitale, mouse, tastiera, stampanti ed altre.

Consente la trasmissione dei dati a velocità elevata con prestazioni superiori rispetto alle porte seriali o parallele.

Il design dell'USB è stato pensato per consentire un semplice inserimento e rimozione delle periferiche. Lo standard prevede che le periferiche vengono rilevate e configurate al momento del loro collegamento (Plug and Play) e possono essere aggiunte o rimosse senza che sia necessario spegnere e riavviare il computer.

La sua introduzione nella didattica in aula deriva dall'esser l'unico standard presente sui PC portatili per collegare le periferiche, come il convertitore USB/GPIB.

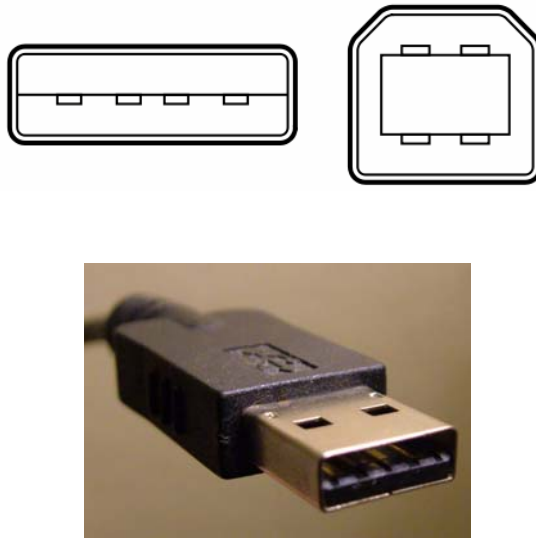


Figura 1.2 : Interfaccia USB

1.3.3 USB/GPIB

Il *GPIB* è uno standard di comunicazione con strumentazione, nelle nostre applicazioni, elettrica. Utilizzato dagli strumenti per interfacciarsi con i calcolatori, in quanto uno strumento di misura è dotato di un interfaccia GPIB che, però, manca dal lato del calcolatore.

Una prima soluzione a tale incompatibilità è stata quella di dotare fisicamente il personal computer 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, ottenuta tramite le schede di espansione, portando verso un'espansione esterna tramite USB. Da qui l'utilizzo dell'USB/GPIB nella didattica, ovvero di un'interfaccia che permetta di collegare PC portatili a strumentazione come Oscilloscopi o Generatori di segnale.



Figura 1.3 : USB/GPIB

1.3.4 Generatori dedicati

I *Generatori dedicati* sono sistemi di generazione di segnali realizzati appositamente per soddisfare delle particolari esigenze richieste dall'utente. In questo contesto è inserito tale elaborato di tesi che prevede la realizzazione di un generatore di segnale da utilizzare nella didattica in aula.

1.4 Generatore di segnale didattico sperimentale

Dopo questa panoramica sulle attività didattiche sperimentali in generale, consideriamo più da vicino l'ambito specifico di questo elaborato di tesi: la realizzazione di un generatore di segnale portatile controllato via USB.

1.4.1 Cos'è un generatore di segnale

Un generatore di segnale è un dispositivo che genera forme d'onda di frequenza, forma ed ampiezza di molteplici tipi, note e regolabili. Ovvero, 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, inserendo armoniche, sweepando il segnale oppure tosando la cima del segnale (vale per l'onda sinusoidale, triangolare, a dente di sega). 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.

1.4.2 Motivazioni della realizzazione di un generatore di segnale didattico sperimentale

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 alquanto più complessa qualora si volesse effettuare gli esperimenti in altre situazioni, esterne ai laboratori, come una dimostrazione didattica in aula, proprio per quest'ultimo motivo.

Inoltre, nella maggior parte dei casi, i generatori di segnali commerciali 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.

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 alle più comuni necessità sperimentali per cui potrà esser usato per vari compiti.

1.4.3 Funzioni del generatore realizzato

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

- Generazione di onde sinusoidali e triangolari.
- Regolazione della frequenza.
- Regolazione dell'ampiezza.
- Regolazione dell'offset continuo.

1.5 Alcuni circuiti per la didattica sperimentale

1.5.1 Carica e scarica del condensatore/induttore

La carica e la scarica di un condensatore/induttore consentono di mostrare come evolve un semplice circuito dinamico RC o RL, mostrati in figura 1.4, con induttore e condensatore lineari e tempo invarianti.

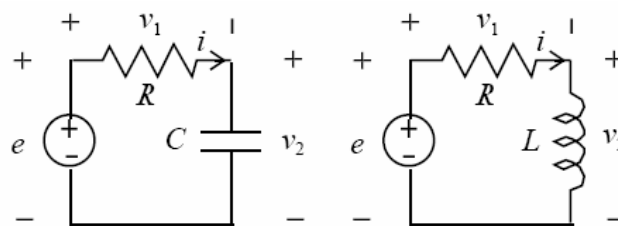


Figura 1.4 : Circuito RC e circuito RL, rispettivamente

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 (rif. bibl. 4).

Nel momento in cui si voglia realizzare un tal circuito con l'ausilio del generatore realizzato in tale elaborato di laurea bisogna effettuare un dimensionamento opportuno dei componenti. Ovvero un resistore ed un condensatore, nel caso del circuito RC, o un resistore ed un

induttore, nel caso del circuito RL, tali che “ τ ” vari nel range che va da 0,05ms a 0,05s. In tal modo è possibile vedere la carica e la scarica del condensatore/induttore all’interno di un singolo periodo del segnale generato dal generatore di segnale realizzato.

1.5.2 Circuito del primo ordine con forzamento sinusoidale

Un circuito del primo ordine con forzamento sinusoidale permette d’introdurre la soluzione in regime sinusoidale. Difatti dopo che il transitorio si è estinto, il regime di funzionamento che si instaura nel circuito è sinusoidale nel tempo con la stessa frequenza del generatore ed indipendente dalla condizione iniziale (rif. bibl. 4).

Per voler realizzare praticamente un circuito di tal tipo si deve tener presente che il reciproco della pulsazione deve essere confrontabile con τ . Ciò porta a dover scegliere un forzamento sinusoidale con una pulsazione che vari nel range che va da 62,5rad/s a $625 \cdot 10^4$ rad/s.

1.5.3 Circuito lineare con forzamento periodico

Un circuito lineare con forzamento periodico permette d’introdurre il regime periodico (rif. bibl. 4).

In particolare se consideriamo un circuito RL forzato da un’onda quadra, può essere analizzato tramite il principio di sovrapposizione degli effetti. Difatti un qualunque segnale periodico può essere rappresentato tramite il suo sviluppo in serie di Fourier, cioè come una somma infinita di funzioni sinusoidali. Tali funzioni avranno una pulsazione ω proporzionale a $2\pi/T$, ove T è il periodo dell’onda quadra, ed anche un’ampiezza proporzionale a quella dell’onda quadra. Il numero di armoniche considerate dipende dall’approssimazione con cui si desidera conoscere i risultati.

Relativamente al generatore di segnale realizzato l'onda quadra avrà un periodo massimo pari ad 1s ed un'ampiezza massima di 10V.

1.5.4 Circuiti risonanti

Per poter introdurre tale tipo di circuiti ricordiamo la definizione di *risonanza*. Consideriamo, quindi, un rete elettrica lineare, alimentata da un generatore sinusoidale. In generale, la corrente erogata dal generatore e la tensione ai capi del generatore stesso non sono in fase tra loro.

Può tuttavia capitare, che per particolari valori di frequenza del generatore, o per particolari valori degli elementi reattivi che costituiscono la rete elettrica stessa, si verifichi che la corrente erogata dal generatore e la tensione ai suoi capi risultino in fase.

Questa particolare condizione è detta *risonanza*; quando essa non si verifica diremo che la rete elettrica si trova fuori risonanza.

Se indichiamo con $Z = R + jX$ l'impedenza della rete vista ai capi del generatore, e se ricordiamo che lo sfasamento tra tensione e corrente, φ , risulta essere dato da:

$$\varphi = \arctan X/R$$

Possiamo dire che una rete si trova in condizione di risonanza qualora la parte immaginaria X dell'impedenza è nulla.

Deduciamo subito che un rete elettrica passiva, per poter essere in condizione di risonanza, deve essere composta sia da induttori che da condensatori, in modo tale che gli sfasamenti di segno opposto si compensino per particolari valori della frequenza del generatore, dando luogo a sfasamento nullo. Quindi un circuito in regime sinusoidale, comunque complesso, nel quale siano presenti resistenze, induttanze e capacità ed un solo elemento attivo si dice in risonanza

quando rispetto al generatore che lo alimenta si comporta come un circuito puramente ohmico.

I classici esempi di circuiti di tale tipo sono il circuito risonante RLC serie e il circuito risonante RLC parallelo, così chiamati in base al modo con cui sono collegati tra di loro gli elementi capacitivi ed induttivi. Tali circuiti sono costituiti da un resistore di resistenza R , un condensatore di capacità C e un induttore di induttanza L , tutti passivi, collegati in serie o in parallelo ed alimentati da un generatore sinusoidale(rif. bibl. 4).

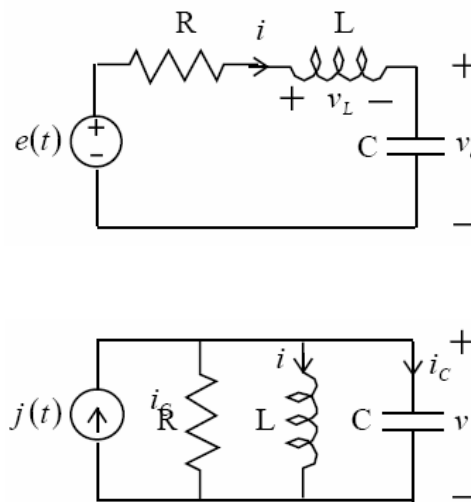


Figura 1.5 : Circuito risonante RLC serie e parallelo, rispettivamente

Nel momento in cui si voglia realizzare un tal circuito con l’ausilio del generatore di segnale realizzato bisogna considerare una pulsazione di risonanza confrontabile con $6,28 \cdot 10^3 \text{ rad/s}$ e quindi, tenendo presente tale pulsazione, dimensionare gli elementi circuitali che lo costituiscono.

1.5.5 Circuito caotico RLD

Il circuito *RLD* 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 regime lineare. 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 questa 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 tesi 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 RLC 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.

E’ comunque interessante studiare il circuito RLD facendo variare sia la frequenza che l’ampiezza del segnale di pilotaggio, si vede così che al variare della frequenza cambiano non solo le ampiezze a cui si sviluppano i vari fenomeni, ma che non sempre i fenomeni caotici sono presenti (rif. bibl. 1).

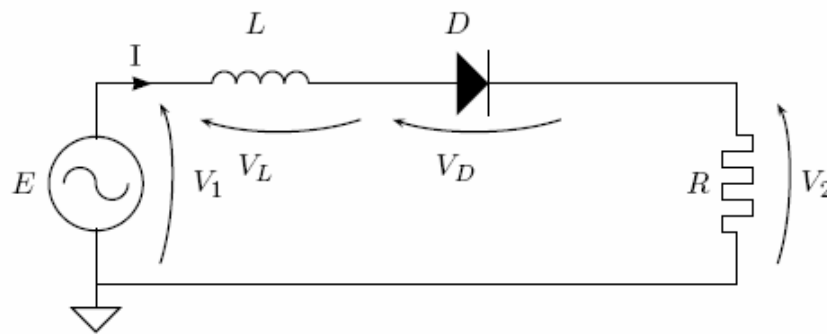


Figura 1.6 : Circuito RLD

1.5.6 Circuito caotico di Chua

Il circuito di Chua deriva dagli studi del prof. L. O. Chua sul caos ed è l’unico circuito in cui la presenza del caos è stata provata in maniera analitica. Il circuito teorico è molto semplice: due condensatori, una induttanza, una resistenza ed un elemento resistivo non lineare con un tratto a pendenza negativa.

La realizzazione pratica del circuito è più complessa perché oltre ai componenti lineari fondamentali richiede due operazionali ed alcune resistenze per implementare il comportamento non lineare.

Il pregio fondamentale del circuito di Chua è quello di essere un circuito autonomo, cioè di non aver bisogno di un segnale in ingresso.

Normalmente si fa variare il valore della resistenza per mostrare la gamma delle possibili dinamiche di questo circuito e questo è proprio il problema fondamentale di questo circuito che sostanzialmente ne preclude l’uso in un sistema sperimentale automatizzato (rif. bibl. 1):

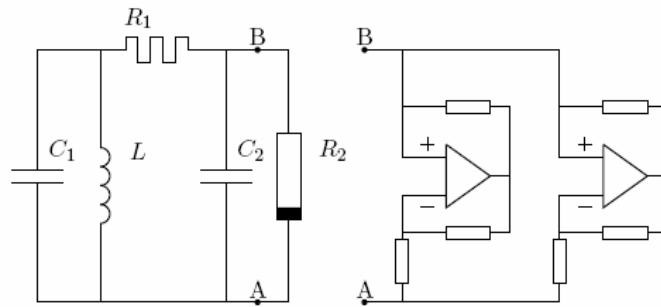


Figura 1.7 : Circuito di Chua

Capitolo 2

PROGETTAZIONE E REALIZZAZIONE DI UN GENERATORE DI SEGNALE USB PORTATILE

2.1 Analisi dello stato dell’arte

Attualmente sul mercato sono disponibili diverse tipologie di generatori di segnali aventi prestazioni, dimensioni e costi differenti, ma soprattutto caratteristiche d’applicabilità dissimili che ne influenzano l’impiego. Fra i vari generatori di segnali esistenti in commercio ne considereremo alcuni per osservarne vantaggi, svantaggi e prestazioni e quindi valutare l’opportunità di realizzarne uno ad hoc.

Tra questi il primo ad essere stato considerato è l’HP-33120A, prodotto dalla Hewlett-Packard e mostrato in figura 2.1. L’HP33120A è un generatore di funzioni arbitrarie, infatti oltre alle tre forme d’onda principali è possibile programmarlo per generare segnali periodici la cui forma, definita per punti, sostanzialmente arbitraria, è inoltre dotato di ingressi di segnali per modulare sia l’ampiezza che la frequenza del segnale generato.



Figura 2.1 : Generatore di segnale HP-33120A

Questo generatore consente di generare forme d'onda arbitrarie con una frequenza di campionamento fissa di 40MSamples/sec, prevede un range di frequenza che va da 100 μ Hz a 15MHz, è dotato di una interfaccia seriale RS-232 e di una interfaccia parallela GPIB per il collegamento a personal computer. Dal punto di vista del controllo software del generatore di segnale sono disponibili sul sito della National Instruments gli appositi driver per LabVIEW che permettono una rapida e semplice integrazione dello strumento all'interno di una applicazione.

Sebbene le prestazioni di questo generatore sono ottimali ed il costo non eccessivo, presenta il serio problema di non essere portatile in quanto ha dimensioni notevoli, infatti presenta un'altezza pari a 103,6 mm, una profondità di 374 mm ed una larghezza di 254,4mm per un peso di 4kg (rif. bibl. 5).

La stessa problematica la presenta un altro generatore di segnali, il 33250A, prodotto dall'Agilent Technologies e mostrato in figura 2.2.



Figura 2.2 : Generatore di segnale 33250A

Difatti questo generatore di segnale permette la generazione di forme d'onda arbitrarie, definite per punti, oltre, ovviamente, a quella triangolare, sinusoidale e quadrata, inoltre consente la modulazione sia in ampiezza che in frequenza del segnale generato essendo munito di appositi ingressi di segnali. Il 33250A fornisce un range di frequenza che va da 1 μ Hz a 80MHz e la possibilità di generare forme d'onda arbitrarie con una frequenza di campionamento di 200MSamples/sec. In più è provvisto sia dell'interfaccia RS-232 che di quella GPIB, può essere controllato da PC tramite l'utilizzo LabVIEW e cosa non meno importante ha un costo contenuto. Però, come già detto, il 33250A ha il problema delle dimensioni avendo un'altezza pari a 104mm, una profondità di 374mm ed una larghezza di 254mm per un peso di 4,6Kg, caratteristiche che, quindi, lo rendono non agevolmente portatile (rif. bibl. 5).

Tale questione è in parte risolta in un altro generatore al momento disponibile sul mercato, il PicoScope 5204, prodotto dalla Pico Technology e mostrato in figura 2.3.



Figura 2.3 : Generatore di segnale PicoScope 5204

Precisamente il PicoScope 5204 è contemporaneamente un oscilloscopio ed un generatore di segnale controllabili via USB anche se non viene specificata la possibilità di gestione tramite LabVIEW. Inoltre è in grado di generare sia le forme d'onde principali che quelle arbitrarie con un frequenza di campionamento di 125MSamples/sec. Ma l'aspetto più importante sono le sue dimensioni, difatti ha una altezza di 40mm, una profondità pari a 255mm e una larghezza di 170mm per un peso di 0,9Kg, quindi ridotte rispetto agli altri generatori visti, ma non ancora tali da renderlo trasportabile. A questa diminuzione delle dimensioni fa da contrappeso un prezzo elevato, che in relazione a quello che fa non è esagerato, tenendo conto che è anche uno oscilloscopio, ma comunque è una cifra sostanziosa (rif. bibl. 6).

Per ovviare a tali problematiche si è pensato di realizzare un generatore di segnale che, oltre a presentare le caratteristiche necessarie di frequenza ed ampiezza in relazione ai segnali generati, fosse portatile, totalmente controllabile da computer e non costoso.

2.2 Analisi degli integrati disponibili

Il generatore di segnale che si vuole realizzare, da un punto di vista hardware, lo si può vedere composto dalle seguenti parti, come si può anche notare dalla figura 2.4:

- Oscillatore o formatore d’onda a frequenza variabile.
- Amplificatore a guadagno programmato.
- Generatore di offset.
- Amplificatore d’uscita.
- Circuito di controllo e d’interfaccia.

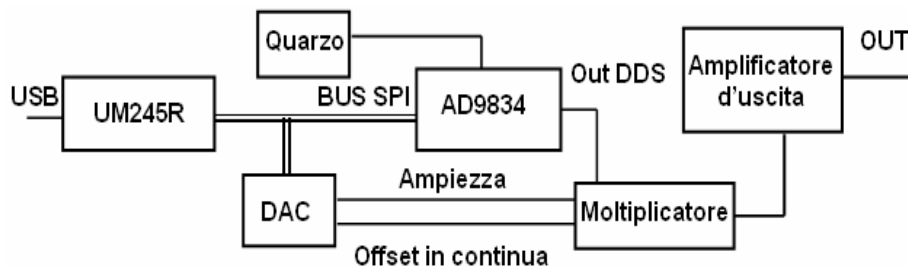


Figura 2.4 : Schema a blocchi del generatore di segnale

Bisogna ringraziare lo sviluppo della tecnologia dei circuiti integrati se è stato possibile sviluppare un valido generatore di segnale con così pochi 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 stato certamente piuttosto critico da mettere a punto.

L'oscillatore è la parte del circuito che è effettivamente dedicata alla generazione dei segnali, è realizzato con un circuito integrato dedicato allo scopo.

Un primo generatore di forma d'onda che si è pensati di utilizzare è l'ICL8038, prodotto dall'Intersil, un circuito integrato monolitico capace di produrre con alta accuratezza forme d'onde sinusoidali,

triangolari, quadre, a dente di sega ed impulsive con una frequenza che può essere selezionata esternamente nel range da 0,001Hz a 300kHz usando o resistori o capacitori (rif. bibl. 7 e 8).

Un' alternativa può essere il XR2206, prodotto dall'Exar, un circuito integrato monolitico capace di generare con alta qualità forme d'onde sinusoidali, triangolari, quadre, a rampa ed impulsive di alta stabilità e accuratezza con una frequenza di operazione che può essere selezionata esternamente nel range da 0,01Hz a 1MHz (rif. bibl. 9 e 10).

Come si può notare le loro prestazioni sono piuttosto limitate in frequenza, questo fatto in aggiunta alla loro maggiore difficoltà d'uso ne hanno sconsigliato la scelta. Inoltre questi integrati per funzionare hanno bisogno di una circuiteria esterna cospicua ed anche un po' più critica nella scelta rispetto ad altri metodi di generazione.

La soluzione più adatta alla realizzazione del generatore di segnale è il circuito integrato MAX038, prodotto dalla Maxim, essendo molto veloce e avendo prestazioni decisamente superiori in quanto è capace di produrre forme d'onde sinusoidali, triangolari, quadre, a dente di sega ed impulsive con una frequenza che può essere selezionata esternamente nel range da 0,1Hz a 20MHz (rif. bibl. 11). Però tale chip è andato fuori produzione, quindi non è più reperibile. In realtà lo si potrebbe realizzare con componenti discreti, quindi avrebbe richiesto l'uso di operazionali, transistori ,però tale operazione è lunga e difficile poiché richiede una buona conoscenza dell'elettronica ed un lungo periodo di lavoro/sviluppo.

La soluzione, quindi, è cambiare tecnologia passando ad un sintesi di tipo digitale, detta DDS (Direct Digital Synthesis), cioè un metodo elettronico per creare digitalmente forme d'onda arbitrarie e frequenze da una singola, fissata sorgente di frequenza (rif. bibl. 21). 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.

In realtà esistono anche altre tecnologie, che non essendo in forma integrata, richiedono l'uso di una quantità consistente di componenti, mentre i DDS sono sottoforma di chip.

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, contemporaneamente non è molto complesso, può essere montato in maniera prototipale perché presenta pochi pin abbastanza spaziosi. Ci sarebbero state altre soluzioni alternative, però prevedevano l'impiego di dispositivi non a nostra disposizione. Altro vantaggio è quello di avere tutto in un unico integrato e la possibilità di ottenere gratuitamente i campioni dalle aziende produttive, avendo per conseguenza costi ridotti o, per meglio dire, nulli del prototipo e la risoluzione dei problemi di reperibilità poiché le aziende adottano una politica delle vendite orientata ai grandi numeri.

2.3 Il circuito integrato AD9834

2.3.1 Descrizione generale

L'AD9834, prodotto dalla Analog Devices e mostrato in figura 2.5, è un circuito integrato DDS a bassa potenza con frequenza massima di clock di 50MHz capace di generare output sinusoidali, triangolari e, essendo dotato di un comparatore interno, anche onde quadre.



Figura 2.5 : Chip AD9834

Contiene al suo interno due registri di frequenza di 28 bit che permettono di ottenere una risoluzione di 0,2Hz, con un clock rate di 50MHz, e di 0,004Hz, con un clock rate di 1MHz.

Questo circuito integrato riceve i dati tramite una interfaccia seriale a 3-fili che opera a clock rate fino a 40MHz, funziona con un'alimentazione elettrica da 2,3V a 5,5V, presenta una sezione analogica ed una digitale che sono indipendenti e che possono essere gestite d'alimentazioni elettriche diverse. Inoltre l'AD9834 ha un pin (SLEEP) che ne permette lo spegnimento se l'integrato non è usato nel circuito in cui è inserito, consentendo un controllo esterno di tal modalità. In più anche sezioni del dispositivo che non sono state usate possono essere spente per minimizzare i consumi (rif. bibl. 17 e 19).

2.3.2 Descrizione dei pin

L'AD9834 è dotato di 20 pin la cui descrizione è indicata nella figura 2.6, dove il circuito integrato è visto dall'alto.

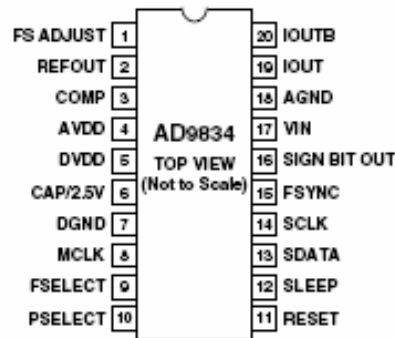


Figura 2.6 : Descrizione dei pin del chip AD9834

FS ADJUST – Full-Scale Adjust Control

REFOUT – Voltage Reference Output. L’AD9834 ha una tensione di riferimento interna di 1,20V che è resa disponibile a tale pin.

COMP – Usato per disaccoppiare la tensione di polarizzazione del DAC

AVDD – Alimentazione positiva per la sezione analogica, può avere un valore compreso tra 2.3V e 5.5V.

DVDD – Alimentazione positiva per la sezione digitale, può avere un valore compreso tra 2.3V e 5.5V.

CAP/2.5V – Utilizzato per motivi di funzionamento interno, viene connesso tramite un opportuno condensatore al pin DGND.

DGND – Digital Ground.

MCLK – Digital Clock Input. Le frequenze d’output sono espresse come una frazione binaria della frequenza di MCLK, cioè la frequenza del segnale di riferimento posto in tale pin.

FSELECT – Frequency Select Input. Tale pin controlla quale registro di frequenza, FREQ0 o FREQ1, è usato nell’accumulatore di fase.

PSELECT – Phase Select Input. Questo pin controlla quale registro di fase, PHASE0 o PHASE1, è addizionato all’output dell’accumulatore di fase.

RESET – Active High Digital Input. Questo pin resetta gli appropriati registri interni a zero.

SLEEP – Active High Digital Input. Quando questo pin è alto, il DAC è spento perché non usato.

SDATA – Serial Data Input. Una parola-dati seriale di 16 bit è applicata a tale input.

SCLK – Serial Clock Input.

FSYNC – Active Low Control Input. Questo è il segnale di sincronizzazione della frame per i dati d’input. Quando è settato basso, la logica interna è informata che una nuova parola è stata caricata nel dispositivo.

SIGN BIT OUT – Logic Output. Uscita del comparatore

VIN – Input to Comparator.

AGND – Analog Ground.

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 (rif. bibl. 17 e 19).

2.3.3 Descrizione dei circuiti interni all’AD9834

Il circuito interno dell’AD9834, mostrato in figura 2.7, consiste di diverse parti:

- un accumulatore di fase a 28 bit,
- una SIN ROM,
- un DAC,
- un Comparatore.

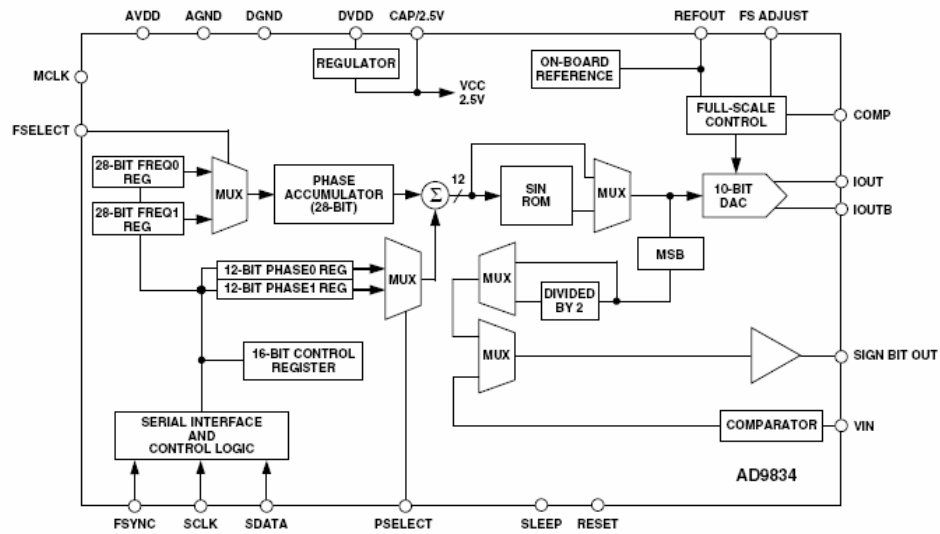


Figura 2.7 : Diagramma a blocchi funzionale dell'AD9834

Il ruolo fondamentale nella generazione di un segnale è svolto dall'accumulatore di fase. Per comprenderne il funzionamento bisogna fare una digressione teorica. Difatti dobbiamo ricordare che le onde sinusoidali sono tipicamente espresse in termini della loro ampiezza, come $a(t) = \sin(\omega t)$, non lineari e non facili da realizzare eccetto tramite una costruzione lineare a tratti. Invece l'informazione angolare, ovvero la fase, è lineare rispetto al tempo e dipende dalla frequenza del segnale che si vuole generare tramite la classica relazione $\omega = 2\pi f$.

Noto che la fase di un'onda sinusoidale è lineare e dato un intervallo di riferimento (periodo di clock), la variazione della fase per tale periodo può essere così determinato:

$$\Delta \text{Phase} = \omega \Delta t$$

Risolvendo per ω

$$\omega = \Delta \text{Phase} / \Delta t = 2\pi f$$

Risolvendo per f e sostituendo la frequenza di clock di riferimento per il periodo di riferimento ($1/f_{\text{MCLK}} = \Delta t$)

$$f = \Delta\text{Phase} * f_{\text{MCLK}}/2\pi \quad 1$$

Su questa semplice relazione è basata l'output dell'AD9834.

Vediamo, quindi, qual è il contributo dell'accumulatore di fase a 28 bit, un cui schema è mostrato nella figura 2.8, nella generazione del segnale.

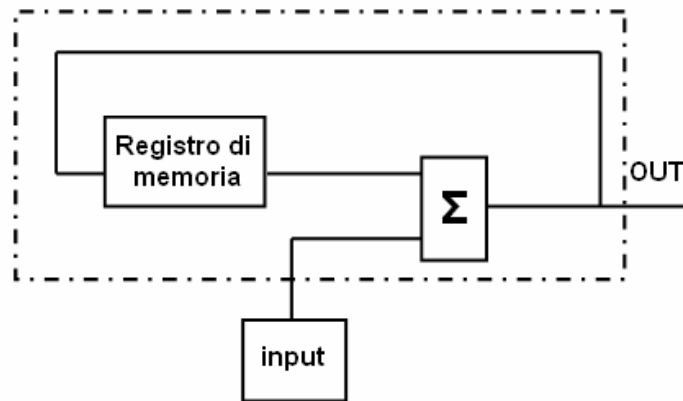


Figura 2.8 : Schema a blocchi dell'accumulatore di fase

L'input dell'accumulatore di fase può essere selezionato tra due registri di frequenza di 28 bit, interni all'AD9834, chiamati *FREQ0 Register* e *FREQ1 Register*, e può essere opportunamente selezionato e caricato dall'utente. L'accumulatore effettua la somma in modulo 28 bit, per ogni ciclo di clock, tra il valore fornitogli dal registro di frequenza selezionato ed il valore contenuto nel suo registro di memoria. Il risultato ottenuto viene sia memorizzato nel suo registro che inoltrato. In realtà del valore trasmesso vengono forniti solo i 12 bit più significativi dei 28 ottenuti, poiché la SIN ROM è da 12 bit.

L'accumulatore di fase nell'AD9834 è implementato con 28 bit, quindi, nell'AD9834, $2\pi = 2^{28}$.

Inoltre, il termine ΔPhase può variare tra 0 e $2^{28}-1$, poiché la fase varia tra 0 e 2π . Sostituendo nell'equazione 1 abbiamo:

$$f = \Delta\text{Phase} * f_{\text{MCLK}}/2^{28}$$

La SIN ROM è utilizzata per effettuare una conversione della fase istantanea della sinusoide nell'ampiezza che ad essa corrisponde. Infatti realizza la conversione dei 12 bit di fase in 10 bit d'ampiezza della sinusoide da inviare al DAC dell'integrato.

Il DAC effettua semplicemente la conversione del valore digitale in corrente.

Il Comparatore può essere utilizzato per la generazione di un'onda quadra tramite un apposito schema di funzionamento (rif. bibl. 17 e 19).

2.3.4 Descrizione funzionale

Il registro di controllo ed i registri di frequenza sono stati esaminati, da un punto di vista funzionale, per comprendere il modo in cui settare i bit che li costituiscono in funzione dei dati da inviare all'AD9834.

Registro di controllo

L'AD9834 contiene un registro di controllo di 16 bit che predispone l'integrato come l'utente desidera utilizzarlo. Descriviamo, quindi, i bit che costituiscono il registro di controllo, mostrato in figura 2.9:

DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	B28	HLB	FSEL	PSEL	PIN/SW	RESET	SLEEP1	SLEEP12	OPBITEN	SIGN/PIB	DIV2	0	MODE	0

Figura 2.9 : Funzioni dei bit di controllo

I bit D15 e D14 servono per informare l'AD9834 che i contenuti del registro di controllo saranno alterati, quando vengono messi a 0.

Il bit D13, indicato con B28, indica se il registro di frequenza opera come 2 registri di 14 bit oppure no. Per comprenderne il significato dobbiamo dire che per caricare una word completa sono richieste due operazioni di scrittura in entrambi i registri di frequenza. Allora se è

settato a 1 permette ad una word completa di essere caricata in un registro di frequenza in due scritture consecutive, la prima contenete i bit meno significativi e la seconda quelli più significativi. Inoltre la prossima scrittura per il registro di frequenza avviene dopo che ambo i 14 bit sono stati caricati. Invece se è settato a 0 il registro di frequenza di 28 bit opera come “due” registri di 14 bit, uno per i bit meno significativi ed uno per quelli più significativi. Ciò permette di alterare uno dei “due” registri indipendentemente dall’altro, mediante una singola scrittura realizzata all’indirizzo di frequenza appropriato.

Il bit D12, indicato con HLB, permette all’utente di caricare i 14 bit meno significativi o i 14 più significativi di un registro di frequenza, mentre ignora i rimanenti 14. Difatti indica quali sono i bit trasferiti al registro di frequenza, in particolare se è settato a 1 permette la scrittura dei 14 bit più significativi del registro di frequenza, se è settato a 0 permette la scrittura dei 14 bit meno significativi del registro di frequenza. Ovviamente è usato in congiunzione con D13, infatti se D13 è pari a 0 allora D12 può essere usato, altrimenti è ignorato.

Il bit D11, indicato con FSEL, definisce quale registro di frequenza, FREQ0 Register o FREQ1 Register, è usato nell’accumulatore di fase.

Il bit D10, indicato con PSEL, definisce quale registro di fase PHASE0 Register o PHASE1 Register, è usato.

Il bit D9, indicato con PIN/SW, indica la sorgente di controllo delle funzioni interne all’integrato come la selezione del registro da usare. In particolare se è settato a 1 sono gli appropriati pin a controllare le funzioni, se è settato a 0 sono i bit opportuni.

Il bit D8, indicato con RESET, se è settato a 1 resetta i registri interni a zero, se è settato a 0 disabilita la funzione Reset.

Il bit D7, indicato con SLEEP1, se è settato a 1 disabilita l’orologio interno, se è settato basso l’abilita.

Il bit D6, indicato con SLEEP12, se è settato a 1 spegne il DAC dell'AD9834, se è settato a 0 lo attiva.

Il bit D5, indicato con OPBITEN, controlla se c'è un output al pin SING BIT OUT. Se è settato alto abilita il pin, se è settato basso vuol dire che l'utente non sta usando tale pin.

Il bit D4, indicato con SIGNPIB, controlla qual è l'output al pin SING BIT OUT. Se è settato a 1 il comparatore è connesso al pin SING BIT OUT. Se è settato a 0 i bit più significativi dei dati del DAC sono passati al pin SING BIT OUT.

Il bit D3, indicato con DIV2, è usato in associazione con i bit D5 e D4. I bit D2 e D0 sono riservati e sempre settati a 0.

Il bit D1, indicato con MODE, controlla qual è l'output ai pin IOUT/IOUTB. Settato a 0 se il bit D5=1. Se è settato a 1 in uscita avremo un'onda triangolare, se è settato a 0 in uscita avremo un'onda sinusoidale.

La word che viene inserita in tale registro, detta word di controllo, è stata opportunamente settata dopo l'analisi di tali bit. In particolare abbiamo posto tutti i bit bassi, tranne il bit D13 che settiamo alto, poiché siamo interessati a vedere il registro di frequenza come un solo registro di 28 bit, ed il bit D1 che merita un discorso a parte. Difatti viene settato sia alto che basso poiché determina il tipo di forma d'onda da generare, infatti permette di selezionare tra onda sinusoidale e triangolare. Quindi, nella realizzazione del generatore di segnale, la word di controllo ha il solo compito di determinare il tipo di forma d'onda che si vuole generare (rif. bibl. 17 e 19).

Registri di frequenza

L'AD9834 contiene due registri di frequenza di 28 bit, FREQ0 Register e FREQ1 Register, che definiscono la frequenza d'output come una frazione della frequenza di MCLK. Il loro accesso è

controllato sia dal pin FSELECT che dal bit FSEL. Se il bit di controllo $PIN/SW = 1$, il pin controlla la funzione; mentre, se $PIN/SW = 0$, è il bit a controllare. Sia che controlli il pin che il bit, se sono settati a 1 si controlla il **FREQ1 Register**, se sono settati a 0 si controlla il **FREQ0 Register** (rif. bibl. 17 e 19).

Scrittura nel registro di frequenza

Quando effettuiamo la scrittura in un registro di frequenza, i bit D15 e D14 della word scritta forniscono l'indirizzo del registro di frequenza. In particolare $D15 = 0$ e $D14 = 1$ indicano **FREQ0 Register**, mentre $D15 = 1$ e $D14 = 0$ indicano **FREQ1 Register**. Invece i restanti bit, D13-D0, contengono il valore da scrivere nel registro selezionato. Prima di effettuare la scrittura nel registro di frequenza, bisogna eseguire la scrittura di una word di controllo che indichi come opera l'integrato, laddove vi sia un cambiamento rispetto alla scrittura precedente (rif. bibl. 17 e 19).

2.4 Altri dispositivi per la realizzazione

Dopo aver visto il chip fondamentale nella costruzione del generatore di segnale in quanto realizza l'oscillatore, analizziamo gli altri componenti necessari per la costruzione del generatore di segnale.

Il modulo **UM245R** che rappresenta il circuito di controllo e d'interfaccia, l'**AD835** che si comporta come un amplificatore a guadagno programmabile.

2.4.1 Il modulo UM245R

Il modulo *UM245R*, prodotto dalla FTDI e mostrato in figura 2.10, è l'evoluzione del circuito integrato **FT245R**, che rappresenta il più

recente dei dispositivi a circuito integrato FTDI, con interfaccia USB UART (rif. bibl. 14).



Figura 2.10 : Modulo UM245R

Il circuito integrato FT245R è una interfaccia parallela FIFO (First In First Out) dotata di connessione USB, con il nuovo sistema di protezione “FTDIDchip-ID”. Inoltre è disponibile un modo di interfacciamento che consente il “Bit Bang Mode”, in modo sincrono e asincrono. Il “Bit Bang Mode” consiste in una particolare modalità di invio dei bit; in pratica i bit vengono trasmessi uno dietro l’altro, come se venissero sparati, su un bus bidirezionale (rif. bibl. 13).

I moduli ad interfaccia parallela/USB usano l’FT245R con una versione per di più semplificata, che consta di una piena integrazione sul dispositivo della memoria esterna EEPROM, del circuito di clock e dei resistori USB. L’FT245R è dotato di una serie di nuove funzioni, rispetto al suo predecessore, ed è utilizzato per parecchie aree applicative. Durante la fabbricazione il dispositivo è coniato con un unico numero di identificazione, leggibile dalla USB, che può essere usato per proteggere l’applicazione software dell’utente.

L’UM245R è fornito su una PCB (Printed Circuit Board), cioè una basetta a circuito stampato, progettata per inserire il dispositivo nella

presa standard USB, con un opportuno cavo, dotato di 24 pin, in modo tale da essere incastrato in uno zocchetto.

La descrizione dei Pin è indicata in figura 2.11, dove il modulo UM245R è visto dall’alto.

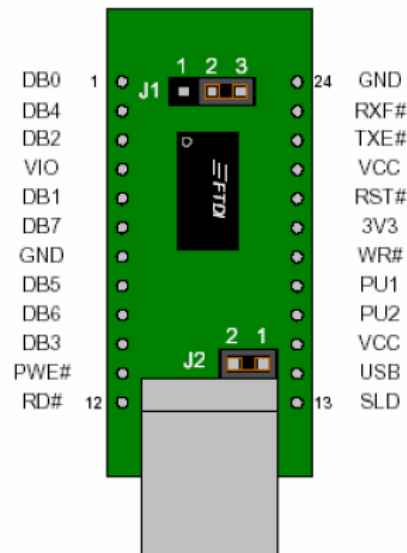


Figura 2.11 : Descrizione dei pin del modulo UM245R

Riguardo alla realizzazione del generatore di segnale portatile controllato via USB, l’attenzione è rivolta solo ad un ristretto insieme di terminali:

Vcc - Tensione di alimentazione

Gnd - Massa

DBx - Generico bit di input/output

Il vantaggio principale di questo modulo consiste nella possibilità di configurarlo in modo tale da essere alimentato tramite la porta USB; in tale configurazione non è necessaria un’alimentazione esterna e

quindi l’utente deve solo collegare il cavo USB, semplificando al massimo l’operazione di collegamento.

L’attenzione è rivolta essenzialmente agli otto bit DB0-DB7, che, nella realizzazione del generatore, devono essere configurati opportunamente per comandare i vari dispositivi.

Dobbiamo inoltre dire che i produttori dei dispositivi dotati di interfaccia USB mettono a disposizione sia le necessarie librerie software, che consentono di adoperare agevolmente questi apparecchi, sia i driver per il sistema host, che nel nostro caso rappresenta il personal computer.

Visitando il sito del produttore del modulo UM245R (rif. bibl. 12) è possibile scaricare le librerie di gestione, scritte in LabVIEW, ed il relativo driver per avere la possibilità di eseguirle.

Il driver scaricato è il file FTD2XX.dll, che va inserito nelle cartelle dove si trovano le librerie.

Nelle librerie sono presenti diversi tipi di primitive che consentono di eseguire svariate operazioni sul modulo, che vanno dalle più semplici tipo azioni di apertura, lettura, scrittura a quelle più complesse di configurazione. Per realizzare la primitiva adatta al nostro scopo, che vedremo nel terzo capitolo, bisogna opportunamente integrarle e quindi costruire il programma adeguato alla regolazione digitale del circuito.

La primitiva **FT_Open** attiva il dispositivo e ritorna il valore di un puntatore denominato Handle, che è utilizzato per un accesso successivo al dispositivo.

La primitiva **FT_SetBitMode** riceve in ingresso l’Handle generato dalla FT_Open e consente di settare i bit DB0-DB7 come bit di input o di output, attraverso il valore di una maschera “Bit Mode Mask” richiesto in ingresso. Se il valore del Bit Mask del corrispondente Pin vale 0 allora viene settato come input, se il valore del Bit Mask è 1 il

corrispondente Pin è settato come output. È richiesto inoltre il valore di un altro parametro di ingresso detto “Mode”, che determina il modo di interfacciamento del dispositivo che può risultare un interfacciamento parallelo, oppure del tipo “Bit Bang Mode”, sincrono o asincrono. Per la regolazione digitale l’UM245R è configurato con un interfacciamento parallelo.

La primitiva **FT_Write** scrive i dati sul dispositivo. Analogamente alla primitiva precedente riceve in ingresso l’Handle generato dalla **FT_Open**; attraverso il parametro di ingresso “Buffer”, che è analogo alla maschera “Bit Mode Mask”, è possibile selezionare i dati da scrivere, mentre il parametro “Bytes To Write ” indica il numero di byte che vengono scritti.

La primitiva **FT_Close** serve a chiudere il dispositivo e necessità del parametro di ingresso rappresentativo dell’Handle, generato dalla **FT_Open**.

Infine è presente, in uscita da ogni primitiva, una sorta di parametro di monitoraggio, che consente di valutare lo stato del modulo, detto **FT_Status**.

In base al valore dell’FT_Status è possibile capire se ci sono degli errori o delle anomalie di funzionamento. Di seguito vengono riportati alcuni valori di questo parametro:

FT_OK = 0

FT_INVALID_HANDLE = 1

FT_DEVICE_NOT_FOUND = 2

FT_DEVICE_NOT_OPENED = 3

FT_IO_ERROR = 4

Lavorando in LabVIEW è possibile concatenare nel modo giusto queste primitive, quindi generare un nuovo programma che consente di settare tutti gli otto bit del modulo come bit di input/output e pilotare adeguatamente i dispositivi.

2.4.2 Il dispositivo DAC

Il dispositivo *DAC* (Digital-to-Analog Converter), 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 (rif. bibl. 21).

Il DAC utilizzato nella realizzazione del generatore è il *MCP4922* appartenente alla famiglia degli *MCP492X*, ove *X* indica i diversi circuiti integrati prodotti.

I chip *MCP492X*, prodotti dalla Microchip Technology Inc., sono alimentati nel range 2,7-5,5V, assorbono poca potenza, presentano una bassa non linearità, sono DAC a 12-Bit con una interfaccia SPI (Serial Peripheral Interface). Forniscono prestazioni ad alta accuratezza ed a basso rumore per le applicazioni industriali dove calibrazione e compensazione dei segnali sono richieste.

I dispositivi *MCP492X* sono DAC a tecnologia stringa con uscita di tensione perché così progettati. Questi apparati includono amplificatori dell'input, amplificatori dell'output di tipo rail-to-rail, buffers sulla tensione di riferimento in input, circuiteria di shutdown e di gestione del reset. La comunicazione seriale è conforme al protocollo di comunicazione standard SPI.

La codifica di questi dispositivi è binaria diretta e la tensione di output ideale è data da una relazione lineare che la lega al codice scritto.

Relativamente al *MCP4922* è un chip dotato di 14 Pin, in modo tale da essere incastrato in uno zocchetto, e costituito da due DAC, DAC_A e DAC_B , in un unico integrato e per tale motivo è stato scelto. Infatti abbiamo 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 del segnale generato. Queste due

tensioni vengono passate al moltiplicatore analogico che realizza effettivamente la regolazione dell'ampiezza e dell'offset del segnale generato (rif. bibl. 15 e 16).

La descrizione dei Pin è indicata in figura 2.12, dove il circuito integrato è visto dall'alto.

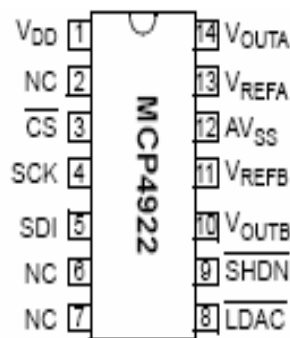


Figura 2.12 : Descrizione dei pin del DAC

V_{DD} – Tensione di alimentazione.

NC – Pin non connesso.

CS – Pin di Chip Select.

SCK – Serial Clock Input.

SDI – Serial Data Input.

LDAC – Latch DAC Input, serve per aggiornare contemporaneamente i due DAC.

SHDN – Hardware Shutdown Input, serve a spegnere il DAC quando non lo si vuole usare.

V_{OUTB} – DAC_B Output.

V_{REFB} – DAC_B Voltage Reference Input.

AV_{SS} – Analog ground.

V_{REFA} – DAC_A Voltage Reference Input.

V_{OUTA} – DAC_A Output.

2.4.3 Il moltiplicatore

Il *moltiplicatore* utilizzato per la realizzazione del generatore è l'AD835, un moltiplicatore analogico con uscita in tensione a quattro quadranti dotato di 8 Pin, prodotto dalla Analog Devices. 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.

Come nei precedenti moltiplicatori prodotti dell' Analog Devices, fornisce riferimenti di massa indipendenti per l'input e l'output ed una maggiore versatilità, questa caratteristica permette all'AD835 di operare con guadagni di tensione, cioè di comportarsi come amplificatore il cui guadagno può essere variato dalla tensione di controllo. Le sue tensioni di input, X, Y, Z, sono tutte nominalmente $\pm 1V$ del valore di fondo scala, con un overrange di almeno 20%. Gli input differenziali di moltiplicazione (X, Y) e l'input (Z) sommatore sono ad alta impedenza, consentendo di non essere influenzato dai circuiti a monte. La tensione di uscita a bassa impedenza (W) consente di non essere influenzato dai circuiti a valle, può fornire fino a $\pm 2,5V$ e può pilotare carichi fino a 25Ω .

L' AD835 presenta un rumore interno molto basso ed è semplice da usare e versatile; infatti prevede che l'input Z possa essere usato sia per sommare tensione sia per realizzare una retroazione e quindi un'amplificazione dell'output.

In termini generali, l'AD835 fornisce la seguente funzione:

$$W = \frac{(X1 - X2) * (Y1 - Y2)}{U} + Z \quad (1)$$

dove le variabili W, U, X, Y e Z sono tutte tensioni, ma U è un fattore di scala dimensionale inserito per non entrare in contrasto con l'equazione dimensionale di valore 1V.

Tale funzione è realizzata dal seguente diagramma a blocchi funzionale.

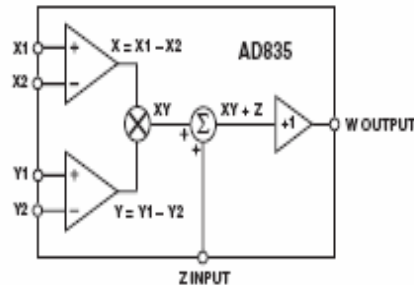


Figura 2.13 : Diagramma a blocchi funzionale

In realtà il valore di base di U può essere modificato al fine di far funzionare il dispositivo anche da amplificatore. Ciò è ottenuto inserendo un partitore di tensione tra W e Z, come mostrato nella seguente figura 2.14:

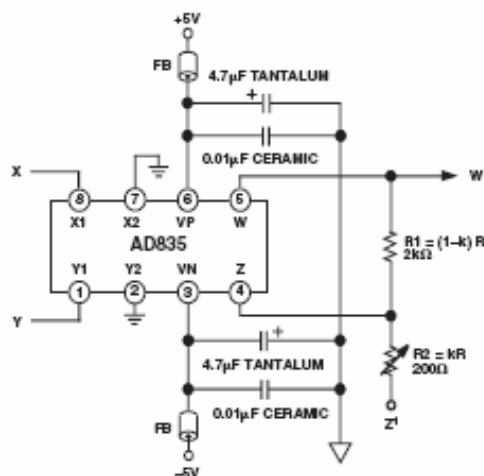


Figura 2.14 : Connessioni moltiplicatore

Quindi l'equazione 1 può essere così riscritta:

$$W = \frac{X * Y}{U} + k * W + (1 - k) * Z' \quad (2)$$

dove Z' è distinto dal segnale Z al Pin 4. Semplificando otteniamo

$$W = \frac{X * Y}{(1 - k) * U} + Z' \quad (3)$$

In questo modo, possiamo modificare il valore effettivo di U senza alterare le tensione d'input in Z' .

$$U' = (1 - k) * U \quad (4)$$

Infine vediamo la descrizione dei Pin che è indicata in figura 2.15, dove il dispositivo è visto dall'alto (rif. bibl. 17 e 18).

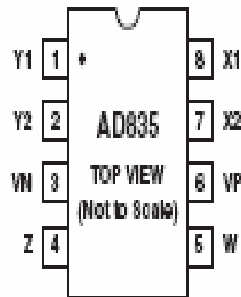


Figura 2.15 : Descrizione dei pin del AD835

Y1 – Tensione d'input.

Y2 – Tensione d'input.

VN – Tensione di alimentazione negativa collegato a -5V.

Z – Tensione d'input.

W – Tensione d'output.

VP – Tensione di alimentazione negativa collegato a +5V.

X2 – Tensione d'input.

X1 – Tensione d'input.

2.5 Collegamento dei moduli

Una volta introdotti i vari dispositivi necessari per la realizzazione del generatore di segnale, vediamo come sono stati integrati insieme, ovvero come li abbiamo collegati. Ciò è stato effettuato grazie alla possibilità di una comunicazione seriale coi dispositivi periferici via SPI (Serial Peripheral Interface). La SPI è usata soprattutto per una comunicazione seriale sincrona di un “host processore” e delle unità periferiche; infatti è una semplice interfaccia che permette la comunicazione tra controllori e circuiti integrati periferici o l’intercomunicazione fra due o più controllori. Il Bus SPI usa un protocollo sincrono, dove la trasmissione e la ricezione è guidata da un segnale di clock (SCKL) generato dal controllore. L’interfaccia SPI consente di collegare diversi dispositivi mentre il controllore seleziona uno di loro con il segnale CS (Chip Select), l’underline significa che il chip selezionato è attivo se il segnale è basso, cioè uguale a zero.

SPI usa un modello Master-Slave, 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 SCKL sono output. Il dispositivo Slave riceve il clock e il chip select dal Master, perciò CS e SCKL sono input. Ciò significa che c’è un Master, mentre il numero di Slave è limitato solamente dal numero di chip select.

Il Bus SPI consiste di quattro linee di segnale:

1. Master Out Slave In (MOSI), il segnale di dati è generato dal Master e recepito dallo Slave, quindi il segnale MOSI è definito come output dal dispositivo Master e come un input nei dispositivi Slave. Il trasferimento dei dati è solamente in una direzione, dal Master ad uno Slave.

2. Master In Slave Out (MISO), i segnali di dati sono generati da Slave, quindi il segnale MISO è definito come un input nel dispositivo Master e come un output in un dispositivo Slave. Il trasferimento dei dati è solamente in una direzione, da uno Slave al Master.
3. Serial Clock (SCLK), è generato dal Master, quindi è un input in tutti i dispositivi Slave, per sincronizzare i dati di input ed output trasferiti.
4. Chip Select (CS) per la periferica, è generato dal Master per selezionare un dispositivo o chip Slave. Alcuni controllori hanno un chip select dedicato per l'interfacciamento SPI chiamato Slave Select (SS) (rif. bibl. 22 e 23).

Ciò è mostrato nella seguente figura 2.16:

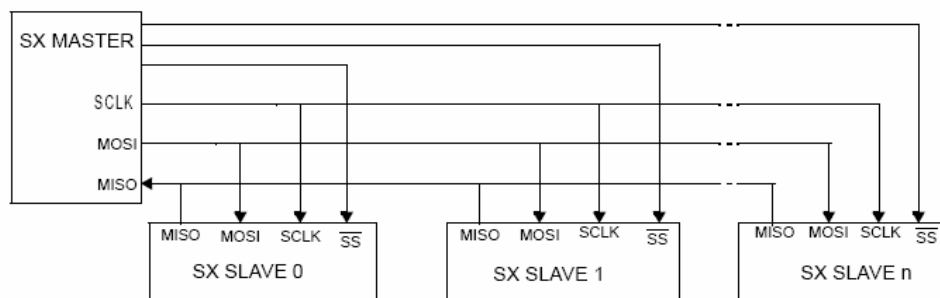


Figura 2.16 : Configurazione del sistema Master-Slave

Nella nostra realizzazione usiamo, però, solo il segnale MOSI per il trasferimento dei dati poiché vengono inviati dal personal computer, tramite il modulo UM245R, che funge da Master, agli altri chip costituenti il generatore di segnale, ovvero l'AD9834 ed il DAC, che rappresentano gli Slave, mentre non c'è nessun segnale che è d'input per il PC, ovvero inviato da uno degli Slave al Master.

Vediamo come tali collegamenti sono stati effettuati praticamente. Abbiamo già detto che nella costruzione del generatore di segnali il

ruolo del Master è svolto dal modulo UM245R, di cui usiamo, come accennato, solo alcuni terminali. Difatti abbiamo utilizzato il pin DB0 per la trasmissione del segnale di dati (MOSI), il pin DB1 per il segnale di clock (SCLK) ed i restanti pin, DB2-DB7, per il segnale CS. Indi, per mettere in comunicazione il chip AD9834 con il modulo UM245R, il terminale SDATA (pin 13) è stato collegato al pin DB0, il terminale SCLK (pin 14) al pin DB1, ed il terminale FSYNC (pin 15) viene connesso al pin DB7 in quanto si individua tale terminale dell'UM245R come pin di selezione del chip Slave AD9834. I restanti terminali dell'AD9834 sono stati collegati come mostra la seguente figura 2.17, rappresentante una connessione di funzionamento del circuito integrato.

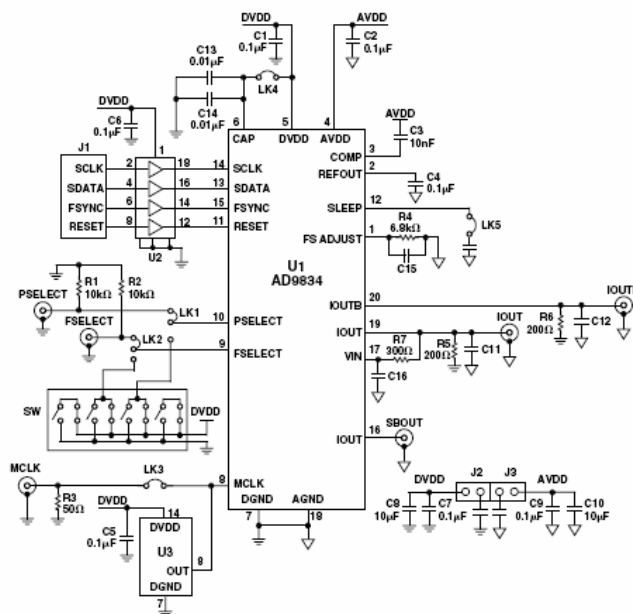


Figura 2.17 : Connessioni AD9834

Però fatta eccezione per alcuni pin; infatti il terminale MCLK (pin 8) è stato connesso ad un oscillatore quarzato che genera una frequenza a 20MHz che, quindi, sarà la frequenza di riferimento. I terminali FSELECT (pin 9), PSELECT (pin 10), RESET (pin 11) e SLEEP

(pin 12) sono tutti messi a massa in quanto pilotiamo il dispositivo tramite i bit della word di controllo e non attraverso i pin. I terminali di uscita IOUT ed IOUTB (pin 19 e 20) li colleghiamo all'ingresso differenziale del moltiplicatore per ottenere un segnale centrato sullo zero senza offset in continua che altrimenti sarebbe scalato dall'ampiezza.. Infatti le due uscite dell'AD9834 variano tra 0V e 5V con una componente continua non centrata sullo zero e non assumendo mai valori negativi proprio perché il range di alimentazione dell'integrato varia tra 0V e 5V.

Invece, per connettere il DAC col modulo UM245R, sono stati effettuati i seguenti collegamenti, il terminale SDI (pin 5) è stato connesso al pin DB0, il terminale SCK (pin 4) al pin DB1, ed il terminale CS (pin 3) viene collegato al pin DB4 in quanto si indica tale terminale dell'UM245R come pin di selezione del dispositivo Slave DAC. I rimanenti terminali del DAC sono stati opportunamente collegati per ottenere un corretto funzionamento del dispositivo. Quindi il terminale V_{DD} (pin 1) è stato connesso a 5V, il terminale LDAC (pin 8) è stato messo a massa perché tale configurazione ci permette di aggiornare contemporaneamente i due DAC, come anche il terminale AV_{SS} (pin 12). Il terminale SHDN (pin 9) è stato collegato a 5V in quanto non ci serve spegnere il DAC quando non lo usiamo. I terminali V_{REFA} (pin 13) e V_{REFB} (pin 11) sono stati connessi al TL431, un diodo Zener di precisione, che permette di fornire una tensione costante e stabile, di cui non è importante conoscere il valore perché è tramite un Trimmer che regoliamo l'uscita al valore voluto. Infatti il terminale V_{OUTA} (pin 14), utilizzato per fornire in uscita la tensione di controllo dell'ampiezza del segnale generato, è collegato al pin Y1 del moltiplicatore tramite un Trimmer. Quest'ultimo è stato opportunamente regolato per far sì che l'uscita del DAC sia resa conforme al range di tensione nominale

in ingresso al moltiplicatore a cui è connesso (i dettagli di come ciò è stato realizzato saranno spiegati nella descrizione del programma di pilotaggio del DAC). Inoltre il Trimmer permette un ulteriore 'piccolo' aggiustamento per compensare la tolleranza delle due resistenze che collegano W e Z nel moltiplicatore, e le tolleranze del guadagno dello stadio amplificatore d'uscita. Infine il terminale V_{OUTB} (pin 10) del DAC, utilizzato per fornire in uscita la tensione di controllo dell'offset del segnale generato, è collegato al pin Z del moltiplicatore. Con l'analisi dei collegamenti dei due Slave, abbiamo anche già visto come sono stati collegati alcuni pin del moltiplicatore. In particolare il terminale Y1 (pin 1) è collegato alla V_{OUTA} del DAC tramite il Trimmer, i terminali X1 e X2 (pin 8 e 7) sono connesse alle uscite dell' AD9834, il terminale Z (pin 4) è collegato alla V_{OUTB} del DAC.

Vediamo, quindi, come sono stati collegati i restanti pin. Il terminale Y2 (pin 2) è stato messo a massa cosicché il valore moltiplicante il segnale generato dall'AD9834 sia proprio quello passato dal DAC. I terminali VN e VP (pin 3 e 6) sono connessi, rispettivamente, alla tensione di alimentazione negativa (-5V) ed alla tensione di alimentazione positiva (+5V). Il terminale d'uscita W (pin 5) è collegato all'amplificatore d'uscita.

Il generatore di segnale così ottenuto è mostrato nella seguente figura 2.18:

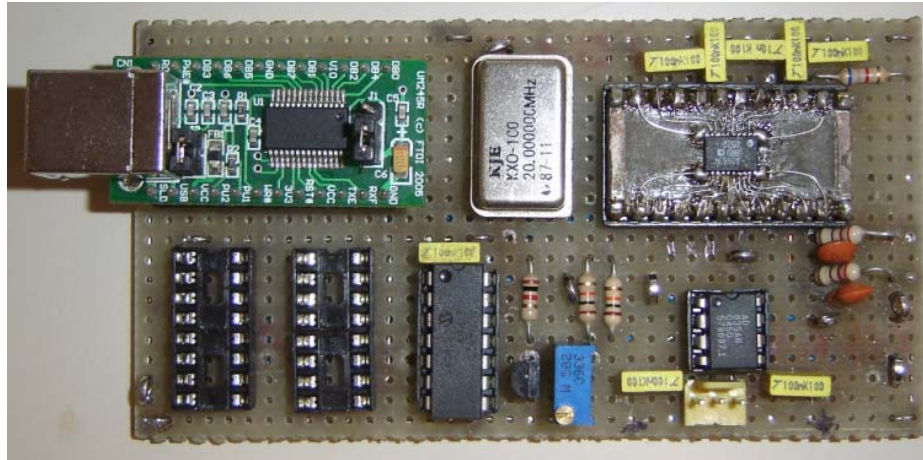


Figura 2.18: Generatore di segnale portatile controllato via USB

2.6 Risultati

Lo sviluppo di questo elaborato di laurea è stato incentrato sulla realizzazione prototipale di un generatore di segnale portatile controllato via USB.

In particolare ci siamo soffermati sulla generazione di forme d'onda di tipo sinusoidali e triangolari variabili in ampiezza e frequenza, come mostrato nelle seguenti figure.

Sullo schermo dell'oscilloscopio è mostrata sia l'uscita diretta del DDS (quella in alto), che ha sempre la stessa ampiezza, sia l'uscita del moltiplicatore (quella in basso), la cui ampiezza è quella settata.

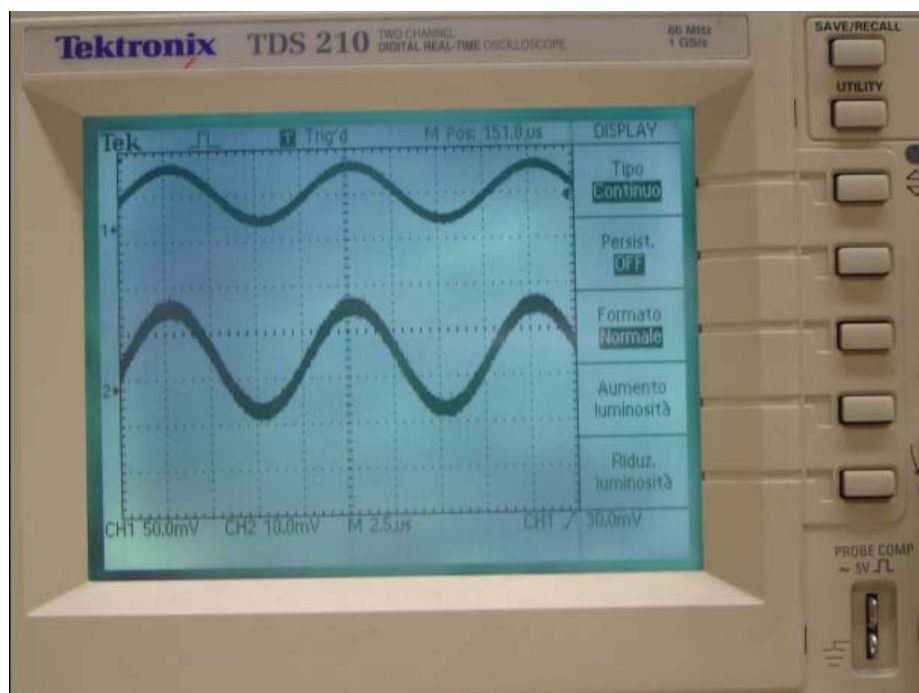


Figura 2.19 : Onda sinusoidale a frequenza 100kHz ed ampiezza 2V

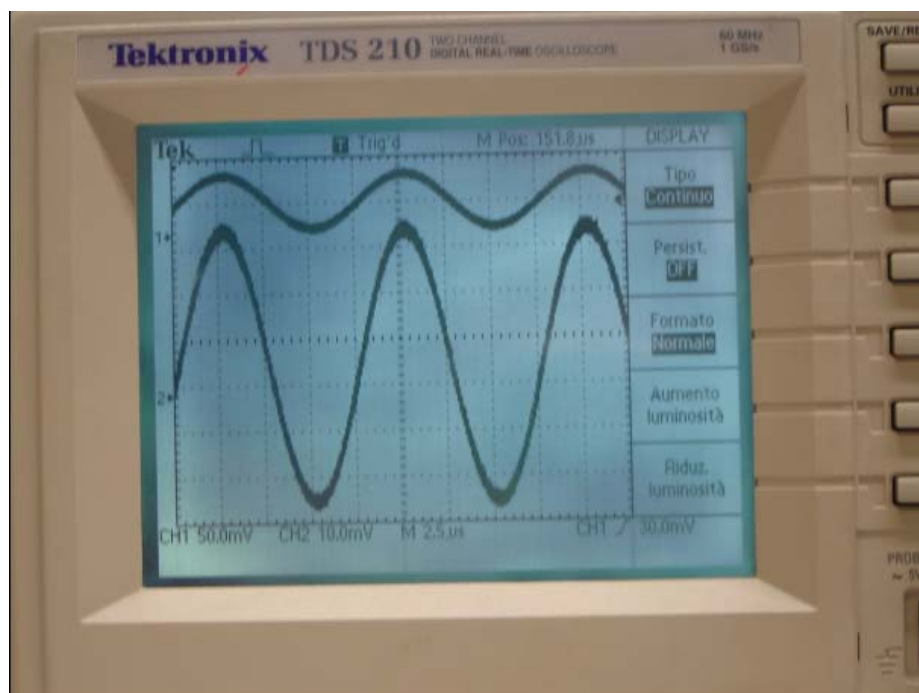


Figura 2.20 : Onda sinusoidale a frequenza 100kHz ed ampiezza 5V

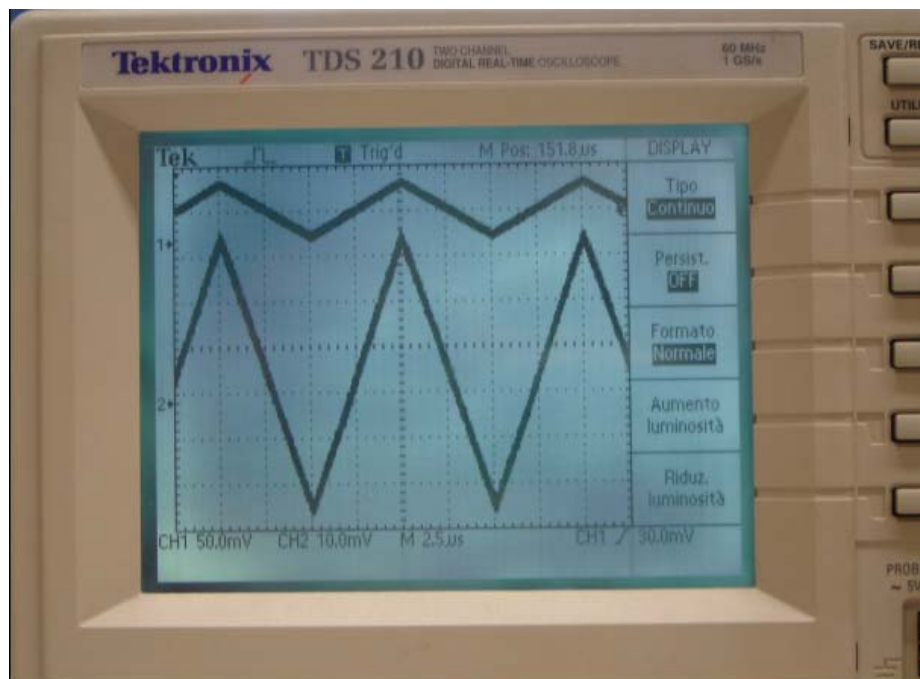


Figura 2.21 : Onda triangolare a frequenza 100kHz ed ampiezza 5V

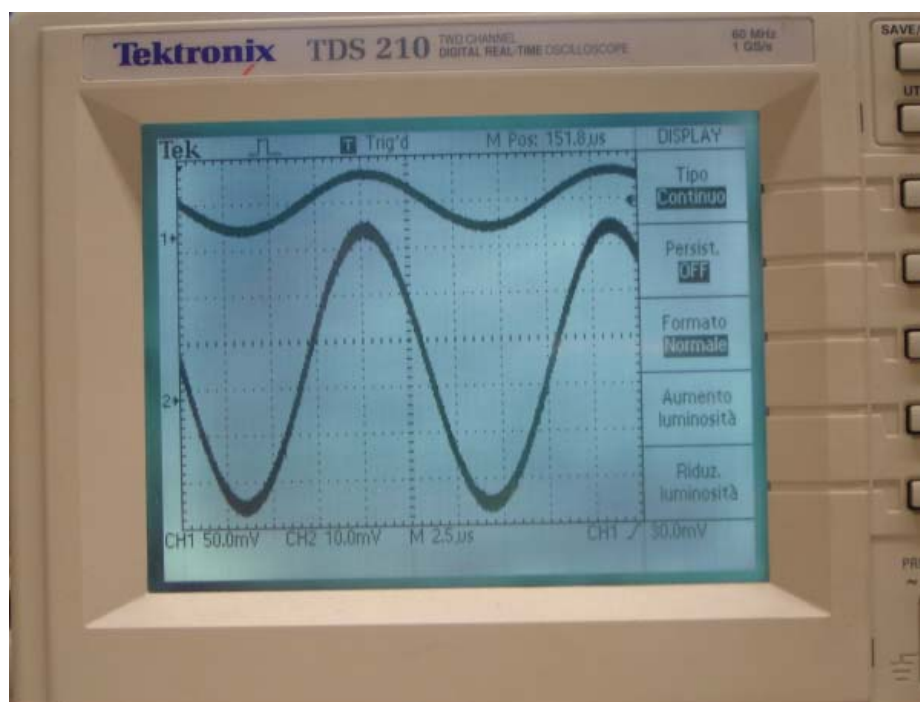


Figura 2.22 : Onda sinusoidale a frequenza 75kHz ed ampiezza 5V

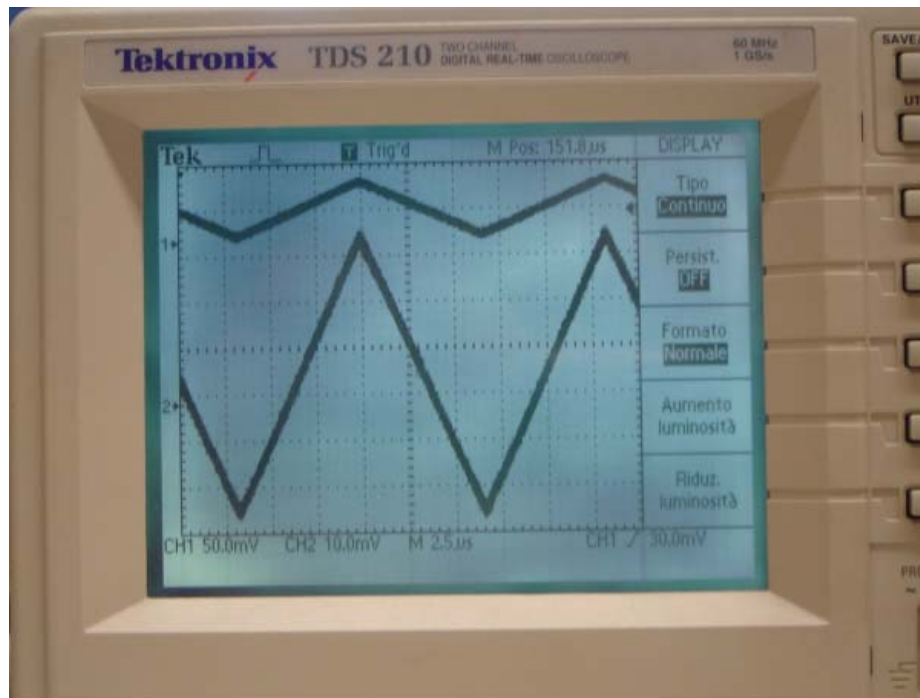


Figura 2.23 : Onda triangolare a frequenza 75kHz ed ampiezza 5V



Figura 2.24 : Onda triangolare a frequenza 2,5MHz ed ampiezza 5V

Come si può notare dalla figura 2.24 quando siamo in corrispondenza di frequenze prossime a quella del Master clock (20MHz) si ottiene

una forma d'onda distorta rispetto alle altre, ciò deriva da un problema di sovracampionamento.

Difatti aumentando la frequenza, indirettamente aumenta anche il ΔPhase e quindi diminuisce il numero di campioni del singolo periodo. Ovvero al crescere della frequenza, aumenta il ΔPhase e diminuisce il numero di campioni presi da una singola forma d'onda perché cambiano i valori di fase istantanea che vengono calcolati.

Capitolo 3

SOFTWARE PER IL PILOTAGGIO E CONTROLLO DEL GENERATORE VIA USB

3.1 Breve premessa

Il capitolo seguente mostra prima una panoramica dell’ambiente di sviluppo in cui si è lavorato: il LabVIEW, spiegandone inoltre il funzionamento, ed in un secondo momento le diverse funzioni che sono state realizzate in tale linguaggio di programmazione grafica per consentire il pilotaggio dei vari integrati costituenti il generatore di segnali costruito.

3.2 Introduzione a LabVIEW

In questo paragrafo descriveremo le caratteristiche fondamentali dell’ambiente di sviluppo LabVIEW, analizzando gli aspetti generali relativi al suo ambito di utilizzo ed alla sua interfaccia utente e descrivendo gli elementi fondamentali per la programmazione.

3.2.1 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 (rif. bibl. 20).

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.

Contemporaneamente questo ambiente di sviluppo software presenta anche alcune 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 segue 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 seriali oppure mediante il protocollo di comunicazione TCP/IP. E' possibile inoltre definire nuove funzioni ed arricchire le librerie in dotazione a 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...

3.2.2 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 (rif. bibl. 20).

3.2.3 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 di uno strumento dotato di display, indicatori, manopole, tasti, eccetera. Nel pannello frontale trovano posto tutti i *controllori* e gli *indicatori* dello strumento virtuale: per *controllore* si intende una variabile di ingresso che può essere modificata agendo sul pannello frontale, 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 uniscono i nodi stessi.

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, detto Sub-VI, cioè un VI utilizzato all'interno di un altro VI come sottoprogramma.

L'icona è un simbolo grafico di piccole dimensioni che rappresenta simbolicamente il VI stesso e che permette di trasformare il programma in un oggetto.

Il connettore stabilisce la corrispondenza tra aree dell'icona e controllori / indicatori del pannello frontale (rif. bibl. 20).

3.3 Funzioni di pilotaggio dei vari moduli

Conosciuto il linguaggio di programmazione grafica LabVIEW, possiamo introdurre le diverse funzioni, ovvero i VI, realizzate per pilotare da PC i vari dispositivi costituenti il generatore di segnale. Alcuni di tali VI sono stati trasformati in Sub-VI sia perchè alcune funzioni sono comuni a più VI, sia per rendere più veloce e chiaro il comprendere il loro scopo ed utilizzo.

3.3.1 VI : PilotUSB

Questo VI è stato realizzato tramite un'opportuna concatenazione delle primitive, citate nel secondo capitolo, per ottenere un nuovo programma che consenta di settare da personal computer quali pin del modulo UM245R sono utilizzati come input e quali come output. Nella sua implementazione la prima funzione impiegata è la **FT_Open_Device_By_Index.vi** che apre la comunicazione col dispositivo (UM245R). Essa riceve in ingresso il parametro Device Index, un controllore che indica il numero di dispositivi connessi al PC, nel caso ce ne sia uno solo occorre porlo uguale a 0, se invece sono due lo dovremmo settare a 1 e così via. Quindi, dato che nella nostra realizzazione usiamo un solo dispositivo, questo parametro avrà come valore 0. La funzione ritorna il valore del puntatore Handle, un indicatore che mostra un sorta di “indirizzo” identificativo del dispositivo collegato e che sarà usato per accessi successivi, ed il parametro FT_Status, un indicatore che fornisce lo stato della funzione. In seguito si connette tale funzione con la Flat Sequence Structure, una struttura che permette la sequenzializzazione dei sub-diagrammi contenuti in essa, o per meglio dire con le funzioni presenti al suo interno, che nel nostro caso, sono tre. In pratica abbiamo collegato l'Handle generato dalla **FT_Open_Device_By_Index.vi** con la funzione **FT_Set_Bit_Mode.vi** che setta il valore del bit “Mode”, cioè indica quali bit del byte DB0-DB7 sono usati come input e quali come output. Tale funzione riceve in ingresso oltre all'Handle della FT_Open, che quindi diventa un controllore, il valore della maschera “Bit Mode Mask”, un controllore che indica quali bit sono settati per l'output e quali per l'input, in particolare se il valore del bit è 0 setta il corrispondente pin come input, se il valore del bit è 1 setta il corrispondente pin come

output. Nel nostro caso il “Bit Mode Mask”, indicato semplicemente come “bitMask”, è un byte, rappresentato nel pannello frontale del VI in binario per consentire all’utente di comprendere subito come sono stati utilizzati i vari pin, che viene settato in tal modo: 10011111 ed è posto come valore di default nel pannello frontale. Difatti usiamo il pin DB0 per trasmettere il Serial Data Input, il pin DB1 per trasmettere il Serial Clock Input ed i restanti pins come CS. Inoltre, un altro input della funzione è il “Bit Mode”, un controllore che mostra il tipo di configurazione del dispositivo da usare, posto da noi, come costante, pari a 1, in quanto indica la configurazione di tipo “Bit Bang Mode” asincrona, quella da noi usata. In uscita alla funzione abbiamo il parametro FT_Status, un indicatore che fornisce lo stato della stessa ed indicato col nome FT_Status 2 per differenziarlo dagli altri FT_Status presenti nel VI. La successiva funzione connessa è **FT_Write_Byte_Data.vi** che scrive i dati sul dispositivo ed è utilizzata per indicare sui bit selezionati per l’output il loro valore che può essere alto (1) o basso (0), rispettivamente per convenzione 5V o 0V. Riceve in ingresso, analogamente alla funzione precedente, l’Handle generato dalla FT_Open, il parametro “Buffer”, un controllore che contiene i dati che devono essere scritti e nella nostra realizzazione l’abbiamo usato per indicare quali tra i bit di output sono alti e quali sono bassi, per tal motivo, nel pannello frontale, è espresso in binario. Un ulteriore parametro d’ingresso è il “Bytes To Write”, un controllore che indica il numero di byte che devono essere scritti, nella nostra realizzazione è settato a 1 perché abbiamo otto pin da manovrare. In uscita alla funzione abbiamo il parametro FT_Status, un indicatore che fornisce lo stato della funzione e da noi indicato col nome FT_Status 3 per distinguerlo dagli altri FT_Status forniti dalle altre funzioni, ed il parametro “Bytes Written”, un indicatore in cui verrà indicato il numero di byte effettivamente scritto. Infine, l’ultima

funzione connessa è **FT_Close_Device.vi** che chiude la comunicazione con un dispositivo precedentemente aperto. Riceve in ingresso l’Handle generato dalla FT_Open e restituisce il parametro FT_Status che abbiamo indicato con FT_Status 4. Questo è il VI realizzato per pilotare il modulo UM245R, attraverso cui inviamo i dati ai vari integrati costituenti il generatore di segnali; il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.1 e 3.2:

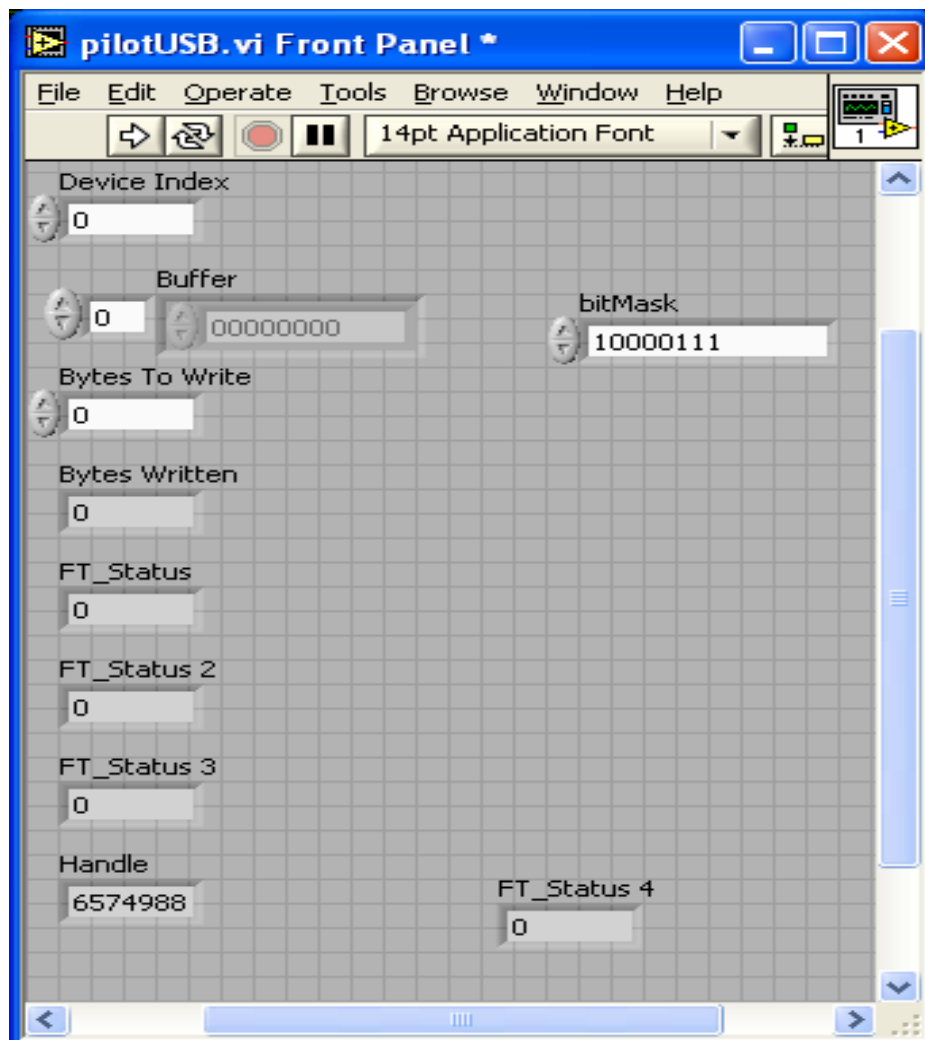


Figura 3.1 : Front Panel del VI PilotUSB

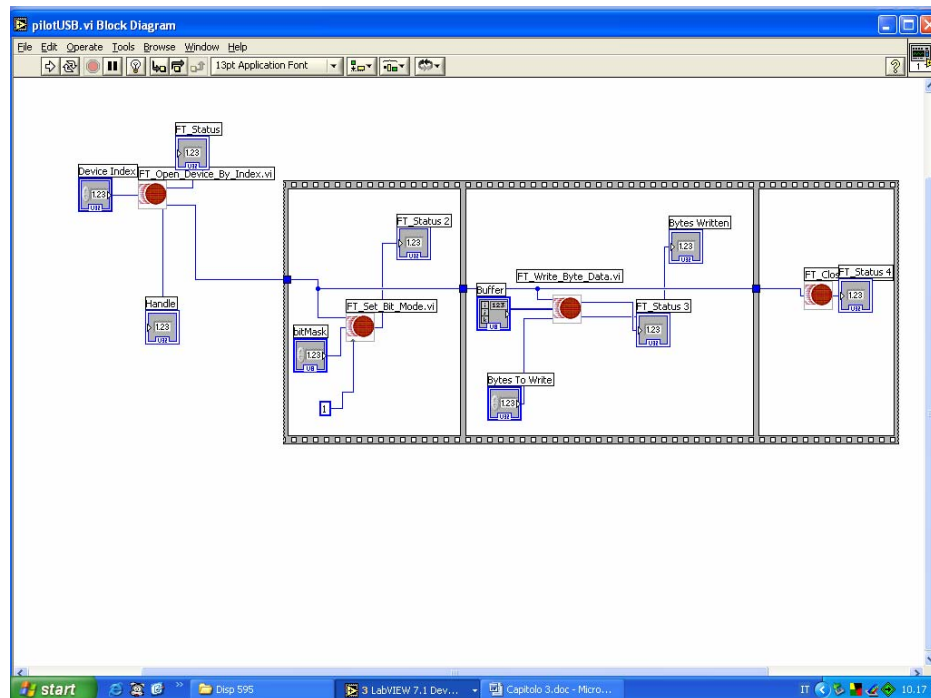


Figura 3.2 : Block Diagram del VI PilotUSB

3.3.2 VI : Word in bit

Questo VI 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). I dati possono, però, essere trasferiti nel dispositivo solamente quando il segnale di Chip Select è basso, in quanto ciò rende attivo l'integrato. Vediamo, quindi, il diagramma di tempismo per questa operazione in entrambi i circuiti integrati. Partiamo dall'analizzare quello dell'AD9834 mostrato in figura 3.3:

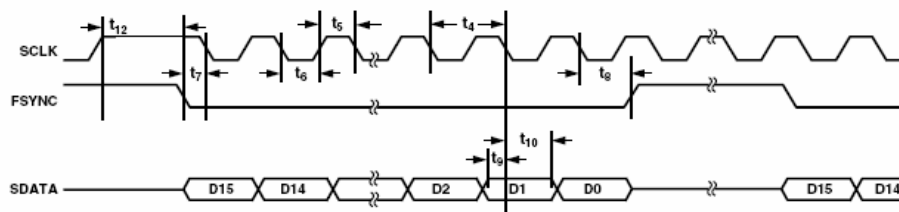


Figura 3.3 : Diagramma di tempismo del protocollo seriale dell'AD9834

Per iniziare il trasferimento seriale dei dati, FSYNC dovrebbe essere messo basso, osservando un tempo minimo, t_7 , tra FSYNC ed il bordo di discesa del SCLK. Dopo che FSYNC diventa basso, i dati saranno spostati nel registro a scorrimento di input del dispositivo in corrispondenza dei bordi di discesa del SCLK per 16 impulsi di clock. Ciò vuol anche dire che 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. Ovviamente nelle due scritture il valore del bit da inviare deve mantenersi costante, mentre quello che varia è solo SCLK. Dopo il sedicesimo bordo di discesa del SCLK, FSYNC può essere portato alto, osservando un tempo minimo, t_8 , tra il bordo di discesa del SCLK e FSYNC. Un aspetto importante è che SCLK può essere continuo, o alternativamente, può essere alto o basso tra operazioni di scrittura, ma deve essere assolutamente alto quando FSYNC va basso (t_{12}). Ora esaminiamo il diagramma di tempismo del DAC mostrato in figura 3.4:

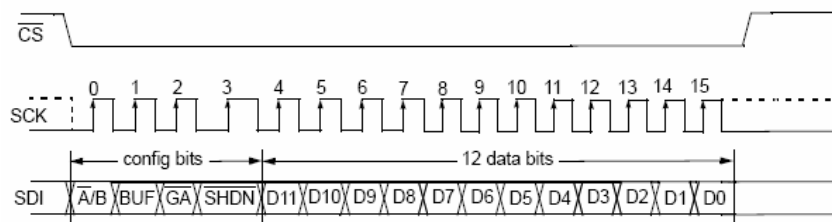


Figura 3.4 : Diagramma di tempismo del protocollo seriale del DAC

La scrittura seriale dei dati nel dispositivo inizia portando in basso $\overline{CS}$. Dopo di che i dati saranno trasferiti nell'integrato in corrispondenza dei bordi di salita del SCK per 16 impulsi di clock. Quindi, anche in tal caso, il passaggio del singolo bit lo si ottiene in due scritture, però nella prima l'impulso di SCK passa da 0 a 1, permettendo il trasferimento del bit nel dispositivo, nell'altra l'impulso di SCK passa da 1 a 0 per consentire la prossima trasmissione. Ciò perché SCK deve essere basso quando $\overline{CS}$ va basso. Ovviamente nelle due scritture il valore del bit da inviare deve mantenersi costante, mentre quello che varia è solo SCK. Il $\overline{CS}$ ritorna alto dopo i 16 impulsi di SCK.

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. Il primo che analizzeremo, indicato con “Word in bit”, è quello usato per il DAC, l'altro, detto “Parola in bit per AD9834”, è utilizzato dall'AD9834, che usa “Word in bit” come Sub-VI, esaminato successivamente. Ora, introdotta la modalità di trasferimento dei dati per il DAC, vediamo com'è stato realizzato il VI “Word in bit”. La sua implementazione richiede un controllore, indicato col nome di Numeric, in cui inserire i dati da inviare e che viene rappresentato come Unsigned Word e con formato binario, pur essendo un numero intero, unicamente per permettere

all'utente la visualizzazione dei 16 bit da trasferire. Ricordando che i dati vengono inviati serialmente come parole di 16 bit, che un bit è inviato in corrispondenza del bordo di salita del SCK, che il tutto avviene in corrispondenza del CS basso e che i dati sono trasmessi settando opportunamente i pin DB0-DB7 dell'UM245R; abbiamo collegato il controllore con la funzione LabVIEW “Number To Boolean Array”, che permette di convertire un numero intero in un array booleano di 8, 16, o 32 elementi, in base al numero di bit nel numero intero, dove l'elemento più a destra corrisponde al bit meno significativo della rappresentazione binaria del numero intero stesso. L'array così ottenuto è posto in ingresso alla funzione LabVIEW “Boolean To (0,1)”, che converte un valore booleano vero o falso in un intero a 16 bit con un valore di 1 o di 0, rispettivamente. Si ottiene così la trasformazione dalla word da trasferire in un array di 16 bit per poter isolare i singoli bit da trasmettere serialmente. L'array risultante va in ingresso alla funzione LabVIEW “Reverse 1D Array”, che inverte l'ordine dei suoi elementi, poiché dobbiamo trasmettere a partire dal bit più significativo. A tal punto dobbiamo ricordarci che la trasmissione avviene serialmente e che per ogni bit inviato dobbiamo settare anche i restanti 7 bit usati per pilotare i corrispondenti pin dell'UM245R.

L'array ottenuto è posto in ingresso a due cicli For dato che il passaggio del singolo bit lo si ottiene in due scritture in cui il suo valore si deve mantenere costante. Col primo ciclo For ricavo i singoli bit della Word da inviare, ecco perché N è posta pari a 16. Col secondo ciclo For duplico il valore del singolo bit per mantenerlo costante nelle due scritture, ottenendo un array di 32 elementi rappresentante la Word da trasmettere. Inoltre nel segnale da trasmettere, per ogni bit, bisogna settare anche il valore dei bit che pilotano i pin. In particolare dobbiamo settare SCK che cambia ad

ogni scrittura passando da 0 a 1 nella prima e da 1 a 0 nella seconda. Quindi nel secondo ciclo For sono state necessarie due operazioni matematiche: una moltiplicazione tra l'indice “i” del For, che varia tra 0 e 1 essendo N pari a 2, ed una costante pari a 2, poiché la sua rappresentazione in binario su 8 bit è 00000010, ed una somma tra il risultato della moltiplicazione, che può essere 0 o 2, ed il singolo bit da trasmettere.

Otteniamo, quindi che se sommo 0 (così rappresentato in binario su 8 bit 00000000) il bit SCK non cambia, se sommo 2 (così rappresentato in binario su 8 bit 00000010) il bit SCK cambia riproducendo le sue variazioni nelle due scritture.

I valori risultanti sono posti in ingresso alla funzione LabVIEW “Reshape Array”, che cambia le dimensioni di un array conformemente al valore di dimensione inserito, nel nostro caso 32.

L'array ottenuto è ulteriormente modificato tramite la funzione LabVIEW “Insert Into Array”, che permette d'inserire un nuovo elemento nell'array nella posizione indicata. Difatti inseriamo due nuovi elementi, uno in testa e l'altro in coda, rappresentanti il settare basso e alto del CS che avviene ad inizio e fine trasmissione. Ciò è fatto settando opportunamente il controllore “PinDiSelezionePeriferica” utilizzato per indicare quale integrato è attivo. Questo è il VI realizzato per costruire l'effettivo segnale inviato; il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.5 e 3.6:

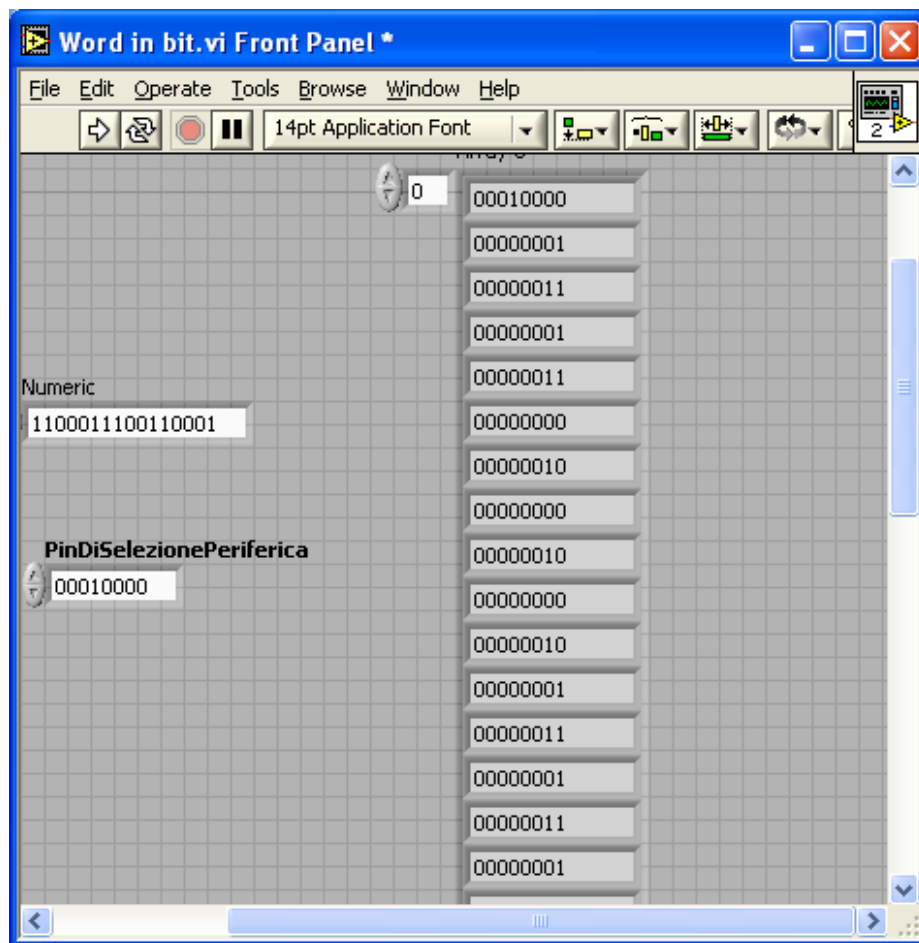


Figura 3.5 : Front Panel del VI Word in bit

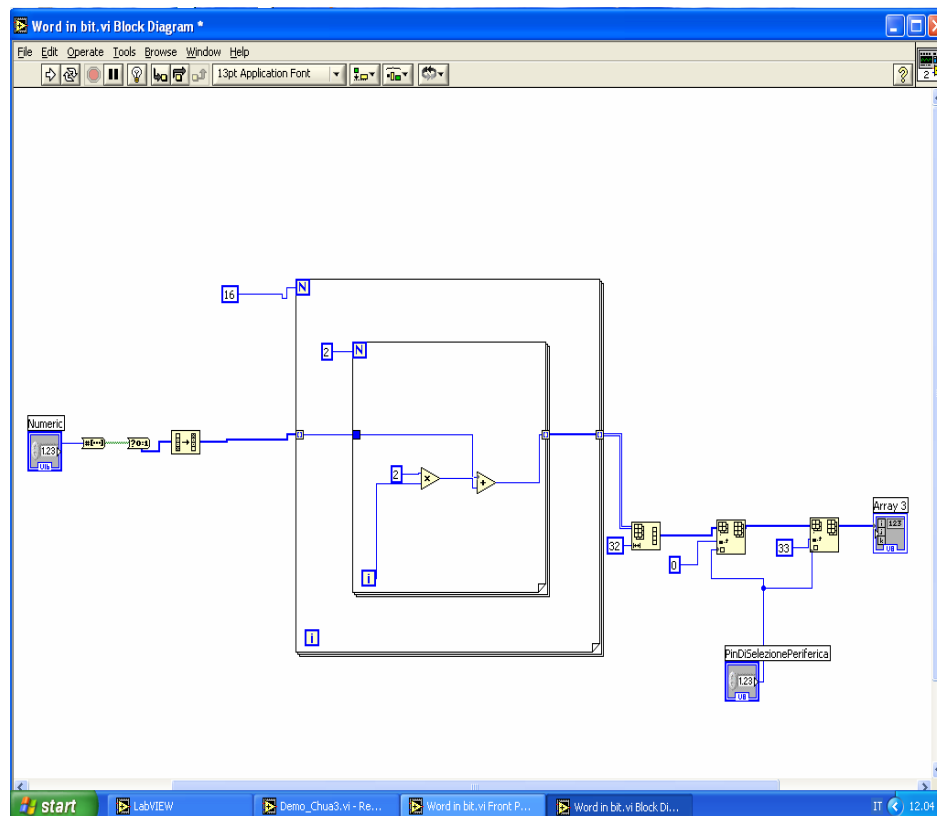


Figura 3.6 : Block Diagram del VI Word in bit

3.3.3 VI : ImpostazioneFreqReg

Questo VI è realizzato per ricavare il valore da caricare nel registro di frequenza dell'AD9834 selezionato, nota la frequenza da generare, indicata con f_{OUT} .

Ricordando che l'output analogico dall'AD9834 è:

$$f_{OUT} = f_{MCLK} / 2^{28} * \text{FREQREG}$$

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:

$$\text{FREQREG} = f_{OUT} * 2^{28} / f_{MCLK}$$

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

Il valore ricavato viene poi suddiviso in 2 parti, poiché una word completa viene caricata in un registro di frequenza in due scritture consecutive. La prima carica i 14 bit meno significativi della word di frequenza e la seconda quelli più significativi. Ciò spiega la divisione per 16384 (cioè 2^{14}), il cui quoziente e resto, rappresentanti rispettivamente i bit meno significativi e quelli più significativi, sono sommati ad una costante pari a 4000, espressa in esadecimale, indicante l'indirizzo del registro di frequenza usato. Infatti i primi 2 bit di ogni parola di 16 bit definiscono il registro di frequenza in cui la parola è caricata.

Questo VI ha, quindi, come controllori f_{OUT} e f_{MCLK} , come indicatori RegFreqLow e RegFreqHigh contenenti, rispettivamente, i bit meno significativi e quelli più significativi. Il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.7 e 3.8:

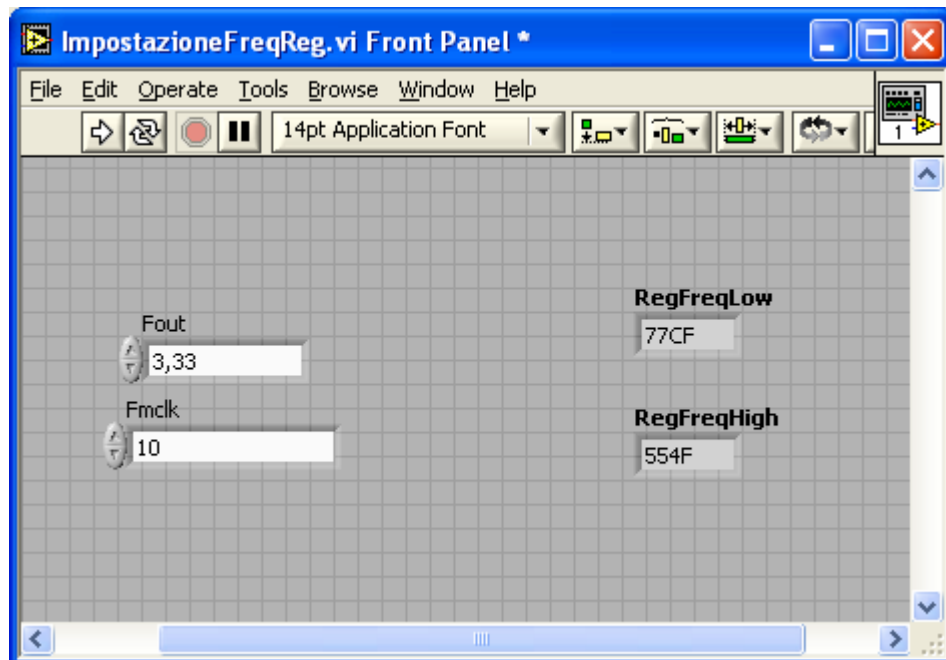


Figura 3.7 : Front Panel del VI ImpostazioneFreqReg

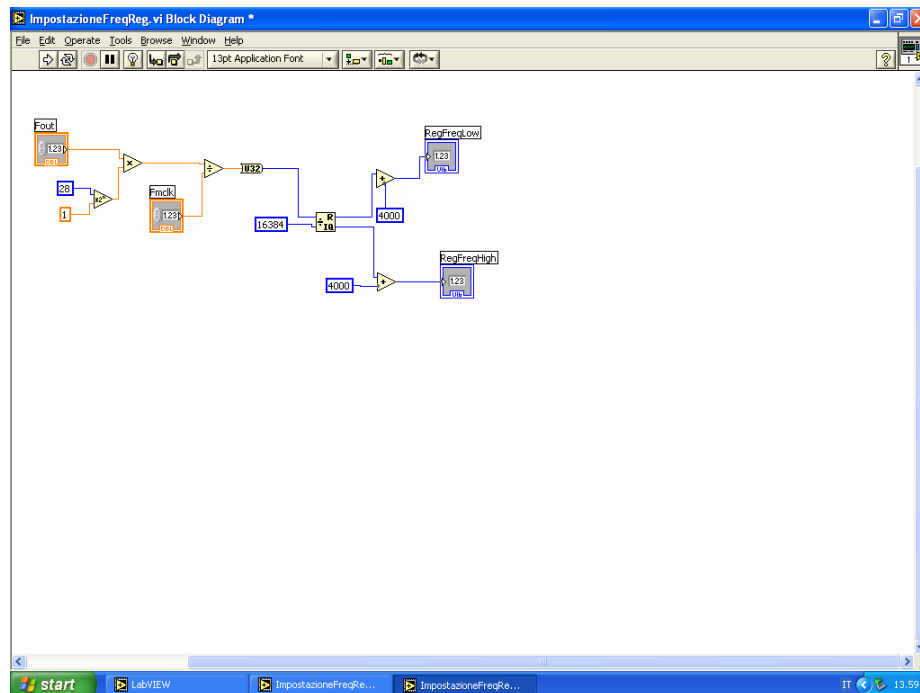


Figura 3.8 : Block Diagram del VI ImpostazioneFreqReg

3.3.4 VI : Parola in bit per AD9834

Questo VI consente la realizzazione del segnale effettivamente inviato all'AD9834 a partire dai dati che si vogliono trasmettere. Come già detto precedentemente utilizza il “Word in bit” come Sub-VI poiché la modalità di scrittura dei dati nell'AD9834 e nel DAC sono simili, cambia solo il Serial Clock Input richiesto. Infatti in questo integrato il SCLK passa da 1 a 0 nella prima scrittura e da 0 a 1 nella seconda. Quindi gli accorgimenti apportati nella sua implementazione sono stati effettuati per poter far fronte a tale cambiamento rispetto al VI “Word in bit”. Inoltre è stata apportata un'ulteriore modifica all'array ottenuto per risolvere un problema d'interazione tra i due integrati, che nasceva in corrispondenza di un uso congiunto dei VI che li pilotano. Problema risolto settando sempre alto (a 1) il pin di selezione del DAC nel VI tramite un ciclo For.

Il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.9 e 3.10:

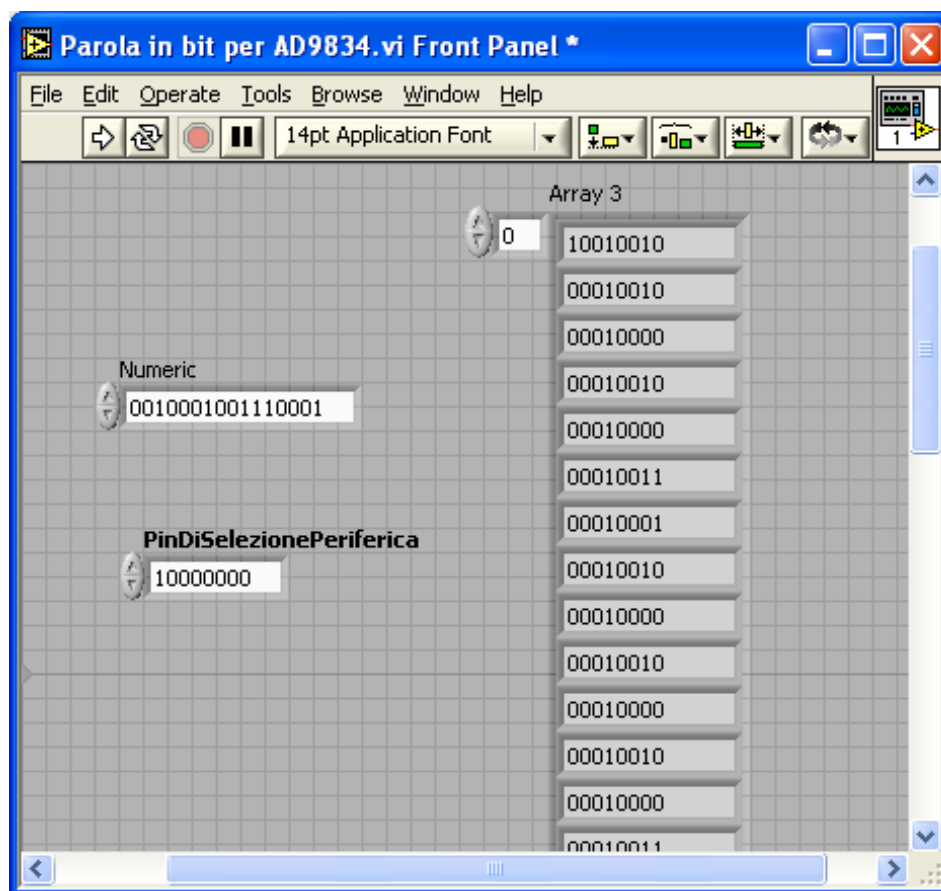


Figura 3.9 : Front Panel del VI Parola in bit per AD9834

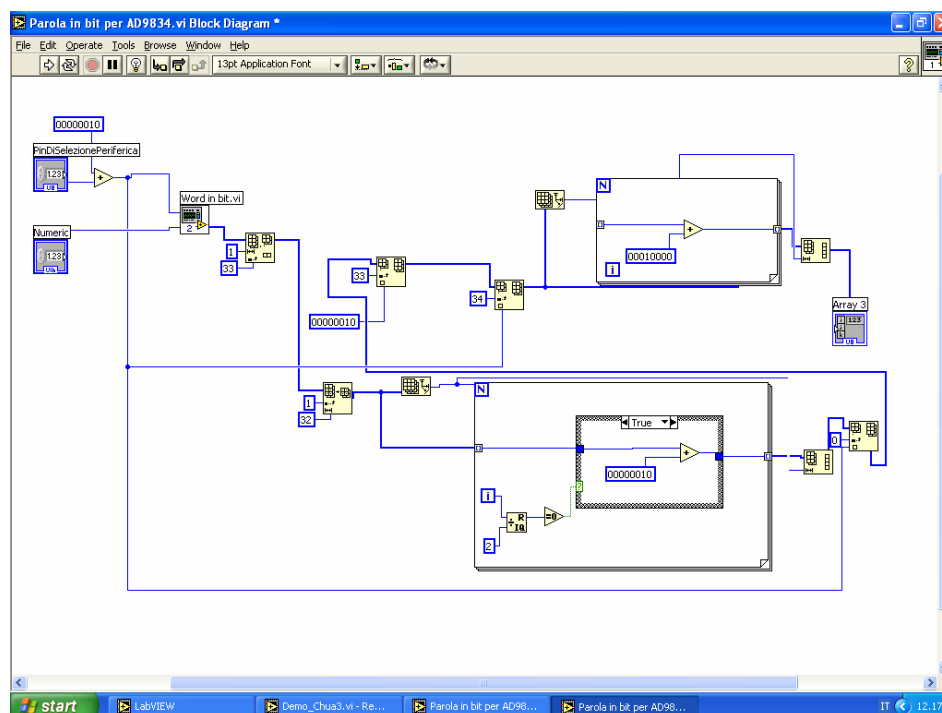


Figura 3.10 : Block Diagram del VI Parola in bit per AD9834

3.3.5 VI : SendAD9834Frequenza

Questo VI è realizzato per effettuare la trasmissione dei dati all'AD9834, in particolare della f_{OUT} desiderata. La sua implementazione è data dalla concatenazione di due VI: “Parola in bit per AD9834”, che realizza il segnale effettivamente inviato a partire dal dato che si vuole spedire, e “pilotUSB”, che trasmette praticamente il segnale così ottenuto. Poiché per trasmettere f_{OUT} dobbiamo effettuare due scritture consecutive, una per i bit meno significativi ed una per quelli più significativi, abbiamo utilizzato una Stacked Sequence Structure, una struttura che consiste di uno o più sub-diagrammi che esegue sequenzialmente. Nella nostra implementazione è costituita da 2 sub-diagrammi, poiché sono necessarie due scrittura consecutive, al cui interno abbiamo posto la concatenazione dei VI prima citata, ma con controllori diversi. Infatti

al primo passiamo, tramite il controllore “RegFreqLow”, i bit meno significativi e al secondo, tramite il controllore “RegFreqHigh”, quelli più significativi.

Il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.11 e 3.12:

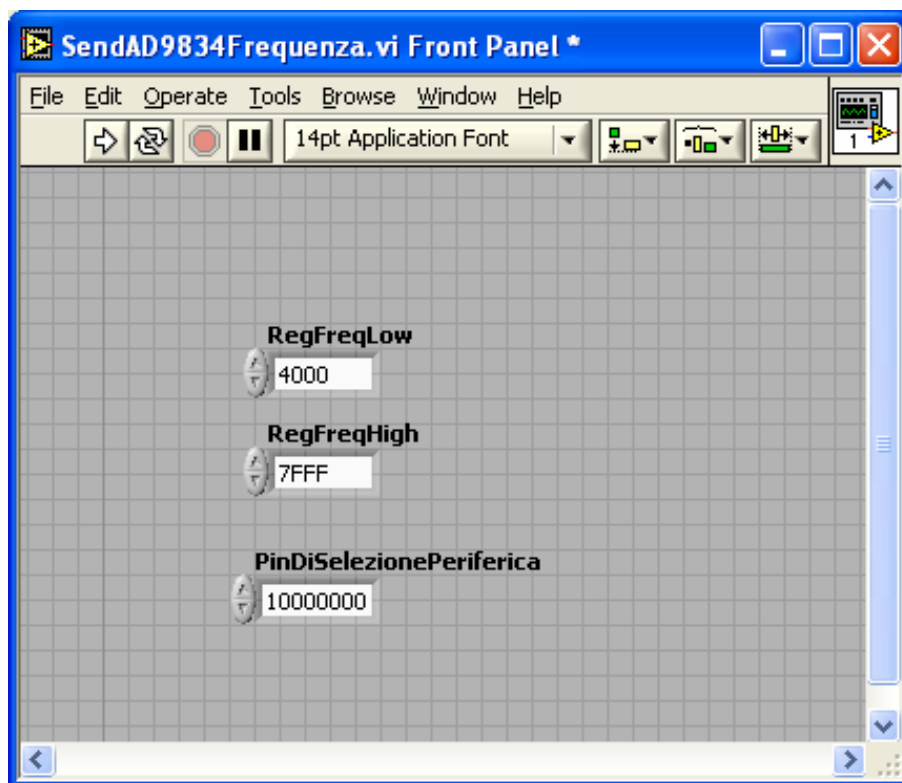


Figura 3.11 : Front Panel del VI SendAD9834Frequenza

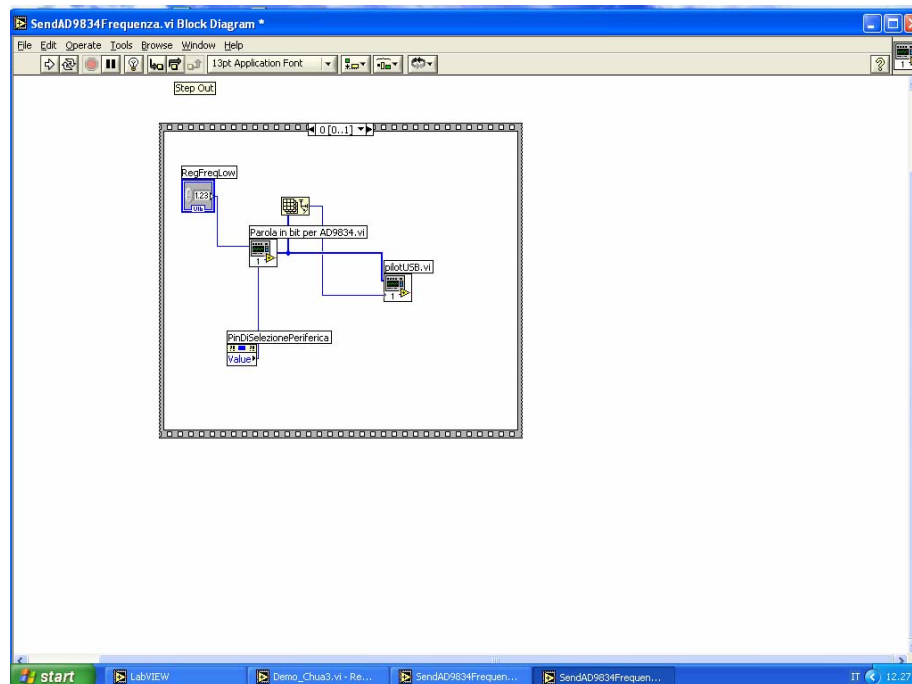


Figura 3.12 : Block Diagram del VI SendAD9834Frequenza

3.3.6 VI : ComandoFreqcompleto

Questo VI è 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} .

Il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.13 e 3.14:

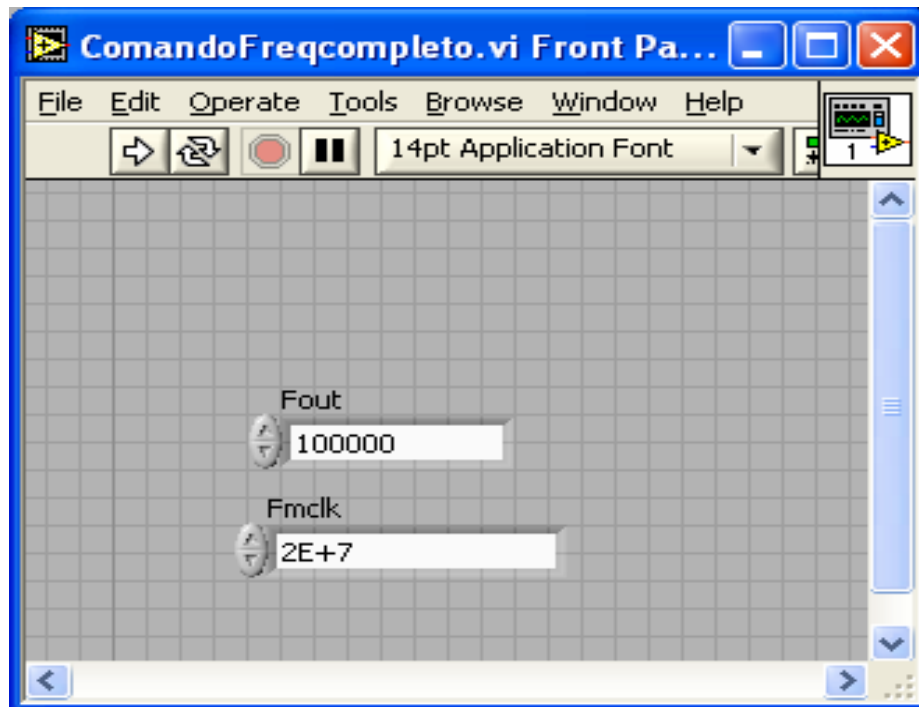


Figura 3.13 : Front Panel del VI ComandoFreqCompleto

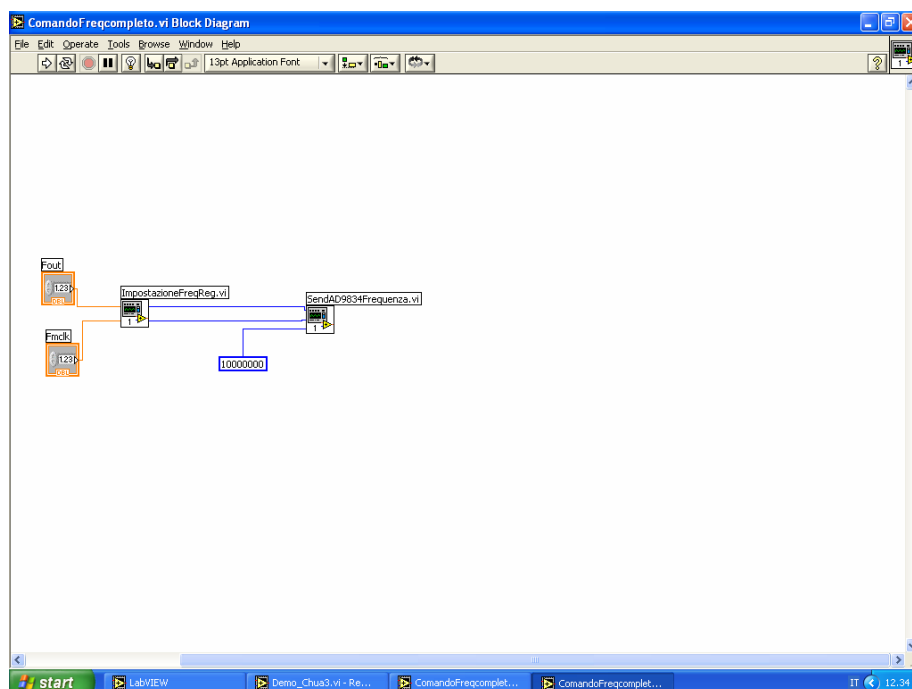


Figura 3.14 : Block Diagram del VI ComandoFreqCompleto

3.3.7 VI : SendAD9834Controllo

Questo VI è realizzato per effettuare la trasmissione dei dati all'AD9834, in particolare della forma d'onda che si vuole generare. La sua implementazione è la stessa del “SendAD9834Frequenza”, l'unica cosa che cambia è il controllore che conterrà il tipo d'onda selezionata.

Il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.15 e 3.16:

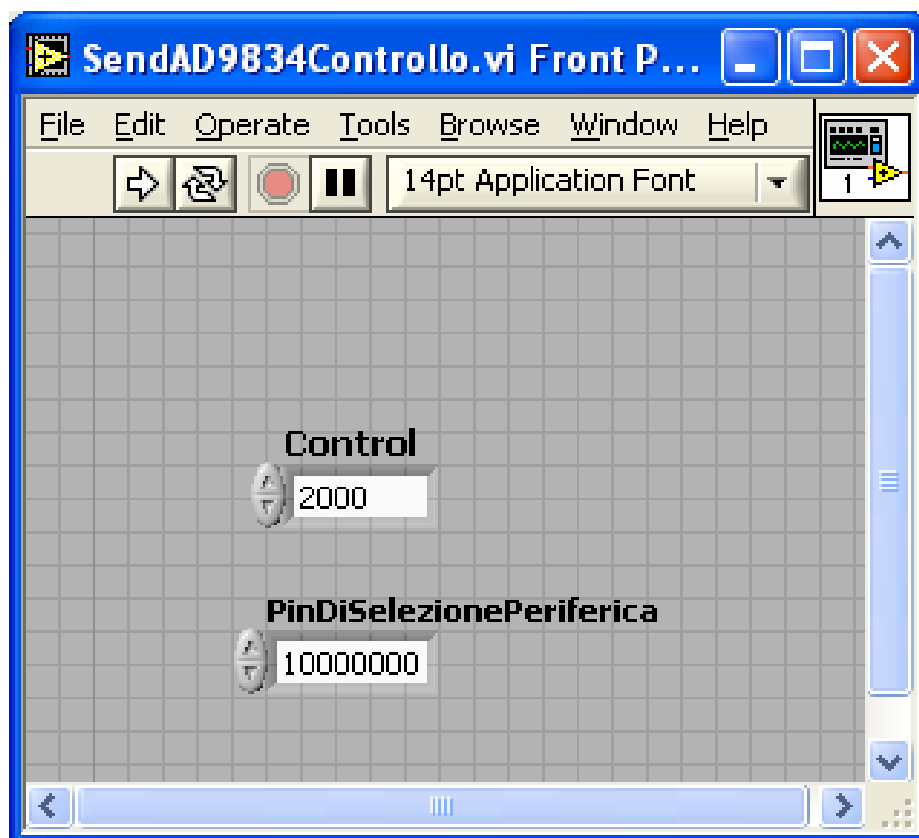


Figura 3.15 : Front Panel del VI SendAD9834Controllo

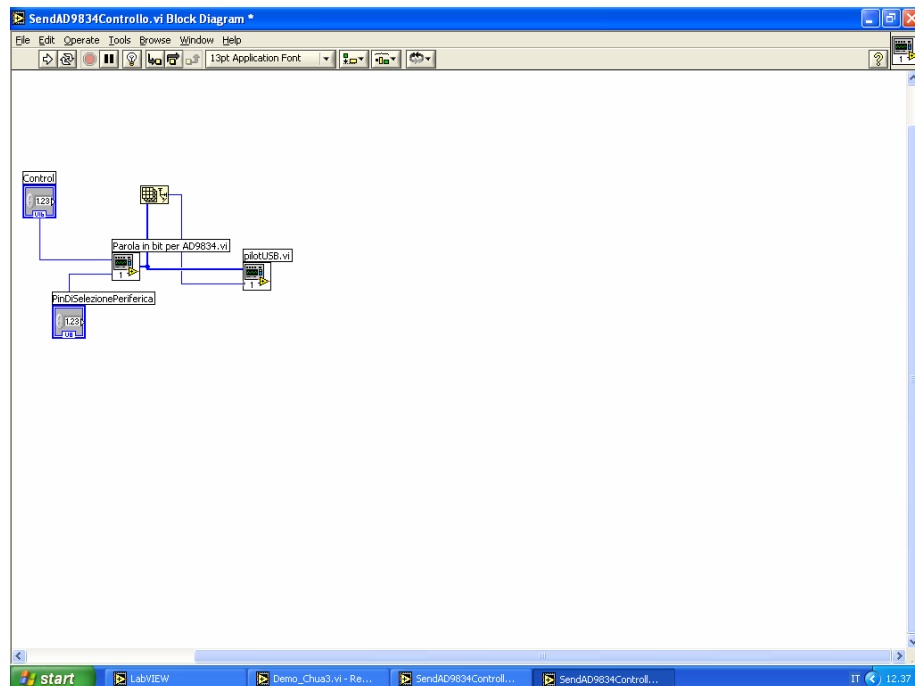


Figura 3.16 : Block Diagram del VI SendAD9834Controllo

3.3.8 VI : DAC

Questo VI è realizzato per passare al DAC il parametro di ampiezza della forma d'onda. Difatti la sua implementazione è data dalla concatenazione di due VI: “Word in bit” che realizza il segnale effettivamente inviato a partire dal dato che si vuole spedire, e “pilotUSB”, che trasmette praticamente il segnale così ottenuto. Ovviamente ci sono degli opportuni accorgimenti dovuti al principio di funzionamento del DAC. Difatti il valore posto in input viene moltiplicato per una costante pari a 400. Ciò permette al range richiesto per l’ampiezza del segnale di essere convertito nel range di valori che effettivamente il DAC può ricevere in input. Per poter comprendere ciò dobbiamo fare una breve digressione. Nella realizzazione del generatore di segnale siamo interessati ad ottenere 10V come valore di picco-picco dell’onda da generare su di un carico

di 50Ω , quindi il range di valori che possiamo passare al DAC va da 0V a 10V. Invece il valore d'ampiezza richiesto viene inviato al DAC tramite un'opportuna word, di cui però solo 12 bit conterranno i dati veri e propri, mentre i primi 4 bit più significativi rappresentano i bit di configurazione del DAC. In particolare il bit 15 seleziona il DAC da scrivere, se è settato alto scriverà sul DAC_B , se è settato basso sul DAC_A . Il bit 14 controlla l'input della V_{REF} che può essere passato direttamente al DAC vero e proprio oppure inviato in un buffer che funziona da separatore. Se è settato alto viene bufferizzato, se è settato basso no. Il bit 13 seleziona il tipo di guadagno utilizzato; se è settato alto il guadagno è pari a 1, se è settato basso il guadagno è pari a 2. Poiché il moltiplicatore utilizzato ha un range d'ingresso compreso tra 0 e 1V, abbiamo scelto un guadagno unitario. Il bit 12 indica la sorgente di controllo della disabilitazione del DAC, se è settato alto la disabilitazione è gestita dall'apposito pin, se è settato basso il DAC è spento.

Quindi il DAC può ricevere in input valori compresi tra 0 e 4095 (cioè 2^{12}). In realtà i valori d'input del DAC considerati sono quelli tra 0 e 4000, in quanto 4000 è molto prossimo a 4095, quindi permette di sfruttare al massimo la dinamica del DAC, ed è un numero “tondo”, dunque ci permette di ottenere un intervallo di tensione “esatto”. Indi, la moltiplicazione inserita ci permette di passare dal range che va da 0V a 10V a quello che va da 0 a 4000.

Mentre l'inserimento del Trimmer, di cui si è accennato nel secondo capitolo, ha permesso di adattare quest'ultimo range con quello di tensione nominale in ingresso al moltiplicatore che va da 0V a 1V. Infatti agendo sul Trimmer si configura l'uscita del DAC e del Trimmer in modo tale che per il valore numerico 0 la loro uscita fosse 0V, mentre per il valore numerico 4000 la loro uscita fosse 1V.

Dopo tale digressione ritorniamo all’analisi del VI, in cui il valore ottenuto dalla moltiplicazione viene poi sommato una word costante rappresentante la configurazione di utilizzo del DAC così settata: 0111000000000000.

Un ulteriore accorgimento apportato è la presenza di un ciclo For per risolvere il problema d’interazione tra i due integrati, AD9834 e DAC, che nasce in corrispondenza di un uso congiunto dei VI che li pilotano. Problema risolto settando sempre alto (a 1) il pin di selezione dell’AD9823 nel VI tramite un ciclo For.

Il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.17 e 3.18:

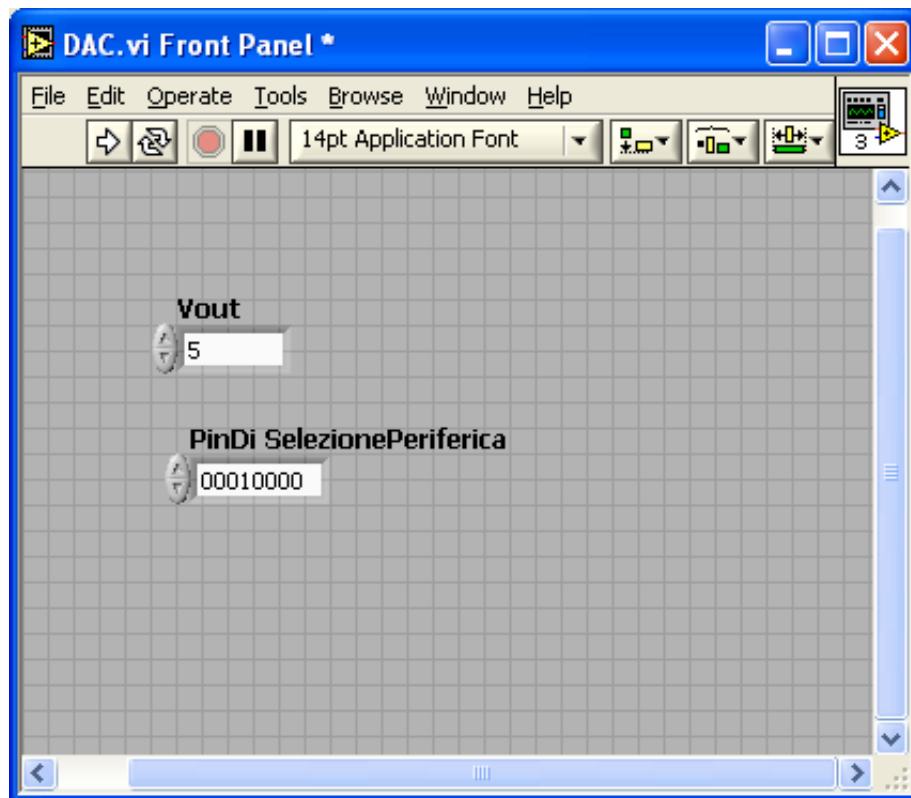


Figura 3.17 : Front Panel del VI DAC

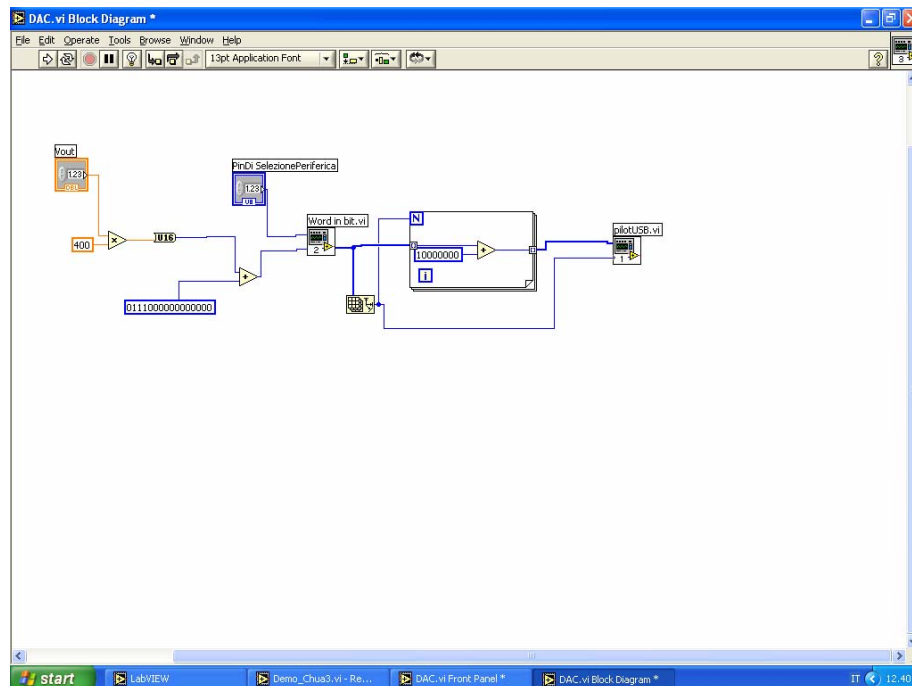


Figura 3.18 : Block Diagram del VI DAC

3.4 Pannello frontale del generatore di segnale

Questo VI è quello che l'utente effettivamente utilizza per pilotare il generatore di segnali realizzato tramite PC. Nell'implementazione del suo Front Panel si è fatto riferimento all'interfaccia dei generatori di segnali esistenti in commercio. Infatti abbiamo inserito un controllore per selezionare la frequenza che si vuole generare, uno per l'ampiezza del segnale da generare, uno per il tipo di forma d'onda ed uno per l'offset. Inoltre c'è un ulteriore controllo che ci permette di stoppare il funzionamento del generatore .

Vediamo, ora, l'implementazione del Block Diagram realizzata mediante un'opportuna concatenazione dei VI esaminati. Si è utilizzato un ciclo While per far sì che in una trasmissione venga inviato solo il valore cambiato, mentre gli altri no. Questo è il motivo per cui abbiamo utilizzato gli “Shift Register”, delle variabili locali

che trasferiscono il valore assunto da una variabile a fine iterazione alla iterazione successiva, la funzione LabVIEW “Not Equal?” e le “Case Structure”, struttura che permette l'elaborazione condizionata di sub-diagrammi a seconda del valore assunto da una variabile di controllo.

Al di fuori del ciclo While c'è il VI “SendAD9834Controllo” con gli opportuni input poiché bisogna inizializzare il chip.

Il Front Panel ed il Block Diagram di tale VI sono mostrati, rispettivamente, in figura 3.19 e 3.20:

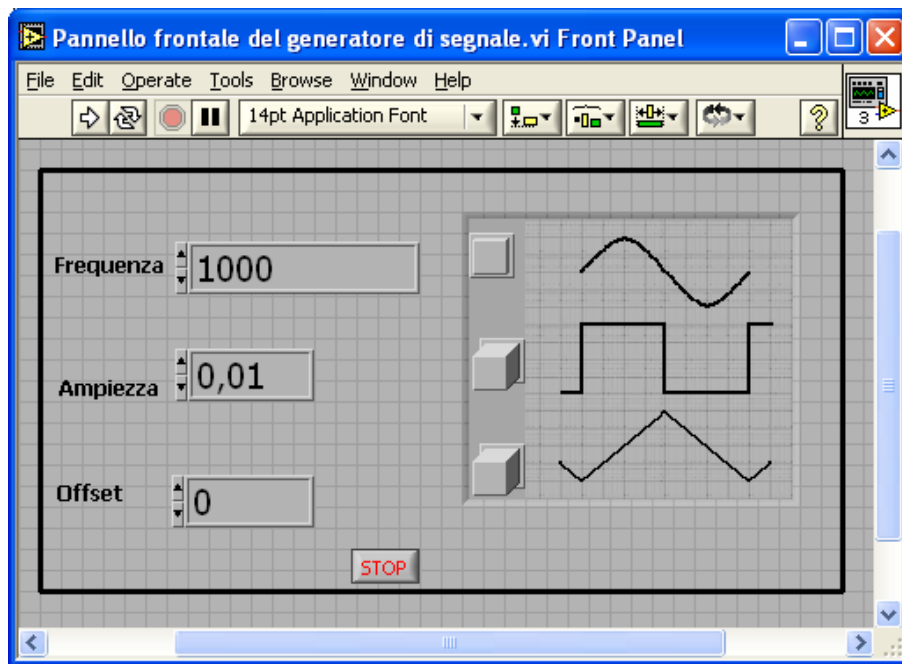


Figura 3.19 : Front Panel del generatore di segnale

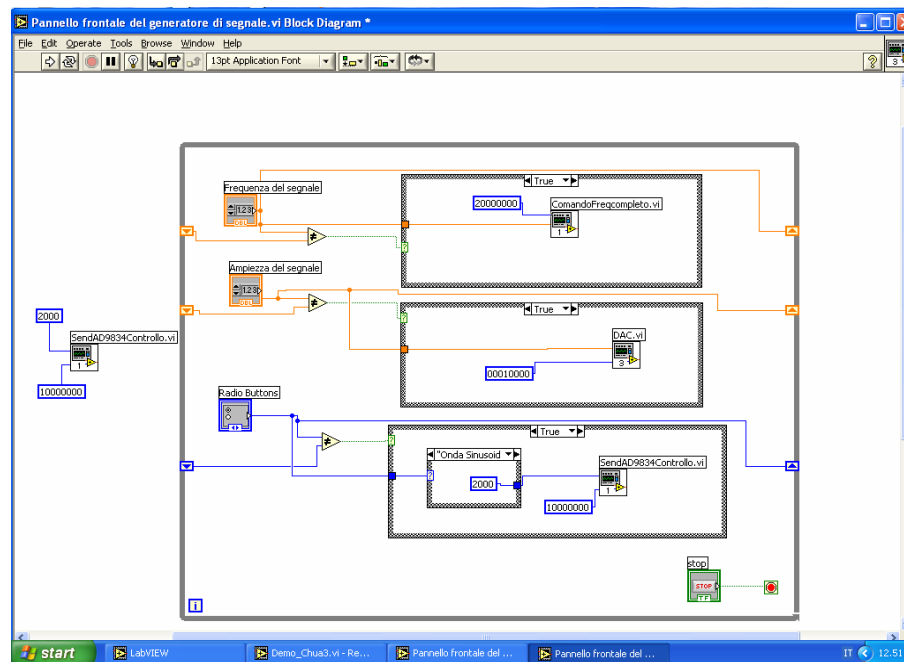


Figura 3.20 : Block Diagram del generatore di segnale

3.5 Conclusioni

Il lavoro svolto in tale elaborato di laurea è stato incentrato principalmente sul pilotaggio del generatore di segnale tramite lo sviluppo di un apposito programma in LabVIEW.

Ovviamente per effettuare ciò è stato necessario un attento studio dei circuiti integrati utilizzati per la realizzazione del prototipo per comprenderne funzionamento ed utilizzo. La scelta di tali integrati è stata dettata dalle nostre esigenze di realizzazione, cioè dimensione, costo ed interfacciabilità, a cui hanno pienamente assolto.

Il prototipo di generatore così realizzato ha soddisfatto tutte le nostre necessità. Difatti ha dimensioni notevolmente ridotte presentando un'altezza pari a 2,5 cm, una lunghezza di 5,3 cm ed una larghezza di 11 cm per un peso di 50 g. Inoltre ha un costo molto basso e consente di realizzare forme d'onda sinusoidali e triangolari con una frequenza

che varia tra 1Hz e 4MHz, anche se il range di funzionamento ottimo è tra 1Hz e 1MHz, ed ampiezza che va da 0,01V a 10V.

Altro aspetto importante è la sua interfacciabilità che permette di poter pilotare il generatore di segnale tramite PC grazie alla realizzazione di un apposito VI LabVIEW che consente all'utente di poter settare la frequenza, l'ampiezza ed il tipo di forma d'onda da generare.

Questi aspetti visti portano a poter utilizzare il prototipo di generatore realizzato nell'ambito della didattica sperimentale, motivazione che ci ha spinto allo sviluppo di questo progetto.

3.6 Sviluppi futuri

Il prototipo del generatore di segnale realizzato in questo elaborato di tesi può essere ulteriormente migliorato. Difatti devono essere ancora realizzate alcune applicazioni “base” che possono essere utilizzate nella didattica sperimentale per i circuiti e che sono presenti nei generatori di segnali in commercio.

In particolare manca un VI per il pilotaggio dell'offset del tipo di forma d'onda che si vuole generare, avendo già realizzato i collegamenti necessari per un suo controllo.

Mancano le connessioni necessarie e gli opportuni accorgimenti da apportare agli integrati utilizzati per la realizzazione di un'onda quadra, con l'eventuale inserimento di altra circuiteria.

Infine il prototipo di generatore di segnale realizzato è privo di un amplificatore d'uscita del segnale generato che, quindi, dovrà essere realizzato e collegato.

Bibliografia

- [1] Ing. Marco Colandrea. Realizzazione di un dimostratore didattico per circuiti caotici (Tesi di Laurea) ;
- [2] www.ni.com;
- [3] Ing. Dario Acanfora. Remotizzazione WEB di esperimento sul circuito caotico di Chua (Elaborato di Laurea) ;
- [4] Introduzione ai circuiti. Anno accademico 2002/2003.
Prof. Giovanni Miano;
- [5] www.agilent.com;
- [6] www.picotech.com/highperformance-oscilloscopes.html;
- [7] www.intersil.com;
- [8] Intersil. ICL8038 Datasheet;
- [9] www.exar.com;
- [10] Exar. XR2206 Datasheet;
- [11] Maxim. MAX038 Datasheet;
- [12] www.ftdichip.com;
- [13] FTDI. FT245R Datasheet;
- [14] FTDI.UM245R Datasheet;
- [15] www.microchip.com;
- [16] Microchip. MCP4922 Datasheet;
- [17] www.analog.com;
- [18] Analog Devices. AD835 Datasheet;
- [19] Analog Devices. AD9834 Datasheet;
- [20] M. Bertocco, Introduzione a LabVIEW, Università di Padova Facoltà di Ingegneria.
- [21] www.wikipedia.it;
- [22] www.scienceprog.com;
- [23] www.mct.net;