

francopic

Decodificatore di tono con l'algoritmo di Goertzel ed un semplice PIC-16F88

2 December 2024

Qualche mese fa, girovagando tra le ormai quasi infinite pagine del web, mi sono imbattuto in diversi articoli che affrontavano il tema di come verificare la presenza di un segnale a frequenza nota, su un canale radio, oppure su un doppino telefonico od altro ancora. Tra le varie soluzioni prese in esame, dai più o meno complessi filtri analogici ai più sofisticati filtri digitali, uno in particolare ha stimolato il mio interesse, poiché in output non restituisce l'ipotetico segnale da noi ricercato, filtrato e ripulito dagli altri segnali indesiderati, ma restituisce semplicemente uno stato logico, alto se nel canale che stiamo analizzando è presente il segnale da noi ricercato, oppure basso se è assente. Per la realizzazione di un tale marchingegno, bisogna ricorrere ad uno strumento matematico noto col nome di "algoritmo di Goertzel". Trattasi di una semplificazione dell'arcinota FFT, ossia della trasformata veloce di Fourier per l'analisi spettrale dei segnali periodici; beato chi ne conosce gli intricati meccanismi di funzionamento, io purtroppo non sono tra questi.

Poiché tale algoritmo interviene esclusivamente sul segnale da ricercare, i cruciali passaggi matematici sono relativamente pochi e semplici, tali da poter essere implementati anche su un piccolo microcontrollore. Questo mi ha invogliato in prima battuta a fare delle simulazioni su un foglio Excel, e successivamente a stilare un programmino che ho realizzato in ASM appositamente per il pic 16F88; microcontrollore prodotto dalla Microchip molto noto per la sua semplicità d'uso. A chi volesse approfondire i vari aspetti teorici e funzionali legati all'algoritmo di Goertzel, consiglio di visitare la pagina al seguente link: <https://www.embedded.com/the-goertzel-algorithm/>. Un ringraziamento va all'autore di questa interessante e stimolante pagina web, che personalmente ho trovato molto chiara ed esplicativa, al punto da permettermi con pochi passaggi di ottenere i risultati sperati.

Dalle prove da me effettuate su dei segnali a frequenza audio, con mio grande stupore ho verificato, come peraltro già anticipato nella pagina web di cui prima, che la risposta dell'algoritmo si può spingere ad una larghezza di banda, fino ad un intervallo di pochi Hertz attorno al segnale ricercato, facendone uno strumento incredibilmente selettivo. Per quanto concerne i nostri interessi, ciò consentirebbe ampiamente di poter realizzare degli ottimi decodificatori di tono del tipo CTCSS oppure dei decoder DTMF od ancora dei radiocomandi; inserendo possibilmente altre funzioni aggiuntive sullo stesso chip, per decodificare ad esempio segnali CW o RTTY.

Il circuito che vi propongo è visibile nella figura 1, e consente la decodifica di toni sub audio, dove oltre al pic è presente anche un filtro RC passa basso del 4° ordine, allo scopo di limitare il passaggio alle sole frequenze sotto i 200 Hz circa ed attenuare i segnali audio a frequenza maggiore normalmente presenti sullo stesso canale. In seguito alle prove da me effettuate, ho notato che il circuito funziona ugualmente bene anche senza il filtro, però credo convenga lasciarlo inserito per non appesantire eccessivamente l'ADC del pic con altri segnali audio in questo caso indesiderati.

[SCHEMA GOERTZEL.jpg](#)

SCHEMA GOERTZEL.jpg

Affinché il circuito operi correttamente, il segnale di bassa frequenza all'ingresso del pic al pin 3, non deve superare 1 Vpp e non deve essere inferiore a 200 mVpp, a tale scopo interverremo sul potenziometro P1 per dosare il segnale opportunamente; il potenziometro P2 invece, stabilisce il valor medio dei segnali d'ingresso, e deve essere regolato in assenza di segnale di bassa frequenza in ingresso, per una lettura col multimetro al pin 3 di circa 0,62 Vcc.

Dato che non ci sono altri punti di taratura, dopo l'accensione il circuito funzionerà immediatamente, e potrete verificare che inviando con un generatore di segnali un tono a 90 Hz al suo ingresso, il led collegato al pin 9 si accenderà, e che variando la frequenza di qualche Hz in più o in meno si spegnerà. Naturalmente questo che condivido con voi è soltanto un circuito di prova, ed intervenendo in modo opportuno sul software di gestione, è possibile restringere ulteriormente la banda passante. Ovviamente per un più opportuno utilizzo, oltre al diodo led si può comandare la chiusura di un relè per gli usi più appropriati.

Coi tre pulsanti presenti nel circuito è possibile modificare in totale autonomia la frequenza del tono da ricercare, e siccome almeno per il momento, non ho previsto un tastierino per l'inserimento diretto del valore del tono, bisogna procedere inviando all'ingresso del decodificatore, un segnale sinusoidale di ampiezza non superiore ai 400 mV circa, con frequenza pari al tono da ricercare; quindi, si continuerà tenendo premuto il pulsante Inc. o Dec. fino a quando non si accenderà nuovamente il led. Tenete presente che tali incrementi o decrementi sono inseriti con step di 0,3 Hz circa e ad intervalli di 180 mS l'uno dall'altro, e se il nuovo valore di frequenza da raggiungere è abbastanza lontano, potrebbero volerci anche alcuni secondi per completare il nuovo inserimento. Le operazioni di modifica termineranno con una breve pressione del pulsante Mem. che salverà i nuovi dati sulla memoria eeprom interna del pic, per renderli utilizzabili alle successive accensioni del circuito; un breve spegnimento del led segnalerà che la registrazione si è conclusa con successo.

Il valore del quarzo da 16 MHz da me utilizzato non è vincolante, infatti, durante le mie prove, erroneamente ho anche inserito un quarzo da 20 MHz, e mi sono accorto dopo un attimo di smarrimento che il circuito stava continuando a funzionare ugualmente, solo

che la frequenza di lavoro si era spostata a circa 110 Hz. A quel punto ho provato anche con un quarzo da 12 MHz, e la frequenza di lavoro si è abbassata a 67 Hz circa. Ciò significa che potrete usare tranquillamente un quarzo intorno ai 16 MHz, tanto poi centerete la frequenza di lavoro coi tre pulsanti come descritto pocanzi.

In breve, il pic acquisisce il segnale d'ingresso grazie al suo convertitore analogico digitale interno, con una frequenza di campionamento esattamente pari a 1 kHz, valore apparentemente basso, ma pur sempre dieci volte maggiore del tono ricercato; pertanto il famoso criterio di Nyquist-Shannon è ampiamente rispettato, che impone una frequenza di campionamento almeno pari al doppio della frequenza da campionare. Ciò significa che tra un'acquisizione e l'altra disponiamo di ben 1000 uS, un tempo relativamente lungo, infatti, considerando che il quarzo utilizzato è di 16 MHz, il tempo occorrente al pic per elaborare ogni acquisizione è di appena 450 uS circa, per cui durante la rimanente parte di tempo di 550 uS circa, il microcontrollore resta per così dire in attesa, con le mani in mano. Tutto ciò conferisce al microcontrollore una buona velocità d'esecuzione delle istruzioni, rendendo il circuito molto efficiente, ed allo stesso tempo anche efficace. Nello stesso tempo raggruppa ed elabora blocchi di 180 campioni alla volta, conferendo al sistema una banda passante di 10 Hz circa, banda passante che oltre a dipendere dai blocchi di campioni elaborati, risente leggermente anche dell'ampiezza del segnale d'ingresso.

Considerando 180 campioni, significa che anche il tempo di risposta del sistema è pari a 180 mS. Volendo ridurre tale tempo dobbiamo ridurre il numero di campioni di ogni blocco, ma così facendo aumentiamo la banda passante del sistema. Diciamo che alla fine come in tutte le cose bisogna trovare un compromesso, oppure passare a microcontrollori più veloci ed aumentare la frequenza di campionamento. Comunque, per le applicazioni in cui spero di utilizzarlo, di cui sicuramente vi terrò informati, va più che bene.

Ovviamente le considerazioni fin qui fatte, valgono se riferite ad un piccolo microcontrollore come il 16F88, che riesce nonostante tutto ed in tempi ragionevoli a svolgere una gran mole di lavoro, e che nonostante le sue ridotte capacità, assolve molto bene al suo compito di decodificatore di tono. Purtroppo ad oggi non esiste un PCB da condividere con voi, tutte le mie prove sono state fatte col circuito realizzato su breadboard, ma visto l'esiguo numero di componenti non vi ci vorrà molto tempo a realizzarne uno.

Se sono stato poco chiaro, oppure ho involontariamente omesso qualcosa nella descrizione del progetto e vi occorrono altri chiarimenti, potete scrivermi al seguente indirizzo e-mail, mira.francesco@gmail.com sarò ben lieto per quanto mi è possibile di rispondervi. Auguro a tutti voi buon lavoro e buon divertimento e di trovare in rete ulteriori spunti sull'argomento trattato, dal fascino intrigante ed accattivante.

Saluti. IT9DPX – #135

Francesco Mira.

Estratto da "<https://www.electroyou.it/mediawiki/index.php?title=UsersPages:Francopic:ctcss-con-l-algoritmo-di-goertzel-ed-un-semplce-immagine-id-21520-name-schema-goertzel-jpg-immagine-id-21520-name-schema-goertzel-jpg-schema-goertzel-jpg-immagine-immagine-16f88>"