Sistemes Encastats

Memòria de les Pràctiques

Realitzat per:

Pavel Macutela

Adrià Auguets

29 de maig del 2017

# Introducció

En les pràctiques de Sistemes Encastats hem realitzat un detector de DTMF utilitzant l’algoritme de Goertzel. La gràcia d’aquestes pràctiques ha estat experimentar quins són els resultats que podem arribar a obtenir si implementem el DTMF en tres dispositius diferents.

Els dispositius que vam tenir oportunitat de programar, van ser:

- Un microcontrolador Arduino.

- Una FPGA (DE0-Nano).

- Un Sistema Operatiu, el nostre portàtil i una Raspberry Pi 3 Model B.

En tots aquests dispositius hem detectat avantatges i inconvenients en el seu ús, tots els resultats es podran comparar més endavant.

## DTMF - Realitzat amb el Microcontrolador Arduino.

La primera pràctica es va realitzar sobre l’entorn Arduino. Com sempre tota primera tasca té les seves complicacions i aquesta no en va ser l’excepció. Nosaltres abans d’assistir a la primera pràctica vam realitzar un estudi previ per familiaritzar-nos amb el que era el Dual-Tone Multi-Frequency signaling. Un cop vam tenir clar quines eren les freqüències que tractava, vam seguir investigant en el mètode proposat a classe per tal de detectar les freqüències que estaven sonant, l’Algoritme de Goertzel.

Un cop assolits aquests coneixements i de fer unes primeres proves en Octave, vam començar a realitzar el nostre codi en C per poder injectar-lo al Arduino. Per tal de realitzar aquesta pràctica vam utilitzar els mòduls entregats pel professor, com per exemple l’ adc.c.

**DTMF.c :**

El nostre mòdul principal, nosaltres considerem que el tenim ben estructurat i fàcil d’entendre. Així com, estructurem el codi de manera que en les primeres línies, com es de costum, tenim els *includes* i les declaracions de les variables amb un nom fàcil d’entendre. Tot seguit es poden veure les funcions que usem i després de les funcions hi trobem el main. Al final del programa, sempre hi deixem lloc per la Interrupció.

Els coeficients que fem servir en l’script els hem calculat prèviament amb Octave. Ho podem veure en la Font 1.

Nosaltres, comencem a tractar les mostres dins la interrupció fins que en tenim 205. Llavors, en calculem la potencia utilitzant vuit filtres i directament anem a una màquina d’estats. Aquesta màquina d’estats, primer s’encarrega de detectar quin número o lletra esta sonant o al contrari, no sona res. Després el mostra per pantalla segons toqui segons la màquina d’estats.

Creiem que un apartat que ens diferència de la resta, es la manera que nosaltres detectem el resultat, és a dir, a partir de quines freqüències estan sonant, saber a quin botó fan referencia. Nosaltres només definim *“resultats[16] = {'1','2','3','A','4','5','6','B','7','8','9','C','\*','0','#','D'};”* i per saber a quina posició de la llista resultats fan referencia les dues freqüències nosaltres hem creat la funció, **uint8\_t detectaResultat(void)**, Font 2.

Aquesta funció mira quines són les freqüències baixes que superen el llindar i actualitza la variable *posició* (+= i\*4) i es posa a true un boleà, depenent de quina superi el llindar. Després fem el mateix amb les altes( += i-4) i també es posa a true un boleà. Alhora per a cada freqüència que superi el llindar, es suma 1 a la variable trobat.

Finalment comparem si els dos booleans donen true i que el valor de trobat es igual a dos. Si això es compleix, anem a buscar el valor que ocupa la posició “*posició*” a la llista resultats.

Que hem aprés en aquesta pràctica d’Arduino?

Com a coneixements que hem assolit realitzant-ho a un microcontrolador, podem destacar que s’ha de tenir sempre en compte els temps d’execució a la hora d’implementar el codi.

També hem aprés a fer servir “càsting” i com tractar amb valors que són decimals, transformant-los en enters, fer càlculs i després tornar a tenir el resultat com si haguéssim fet el càlcul amb floats però amb un temps molt més ràpid.

FPGA:

(PAVELASO)

Hem obtingut una fpga model de0nano...

Hem treballat en Quartus

Ens han donat els mòduls...

Hem implementat....

Quants multiplicadors fa servir?

Ens diferenciem en alguna cosa? La posem. Jo crec que usar 4 cables per transmetre el senyal, esta be + enable + gnd.

(SO, RASP, PYTHON) (ELS DOS).

CONCLUSIONS (ELS DOS)

Diferencies entre els 3

Perquè A B o C depenent de que ens demanin.

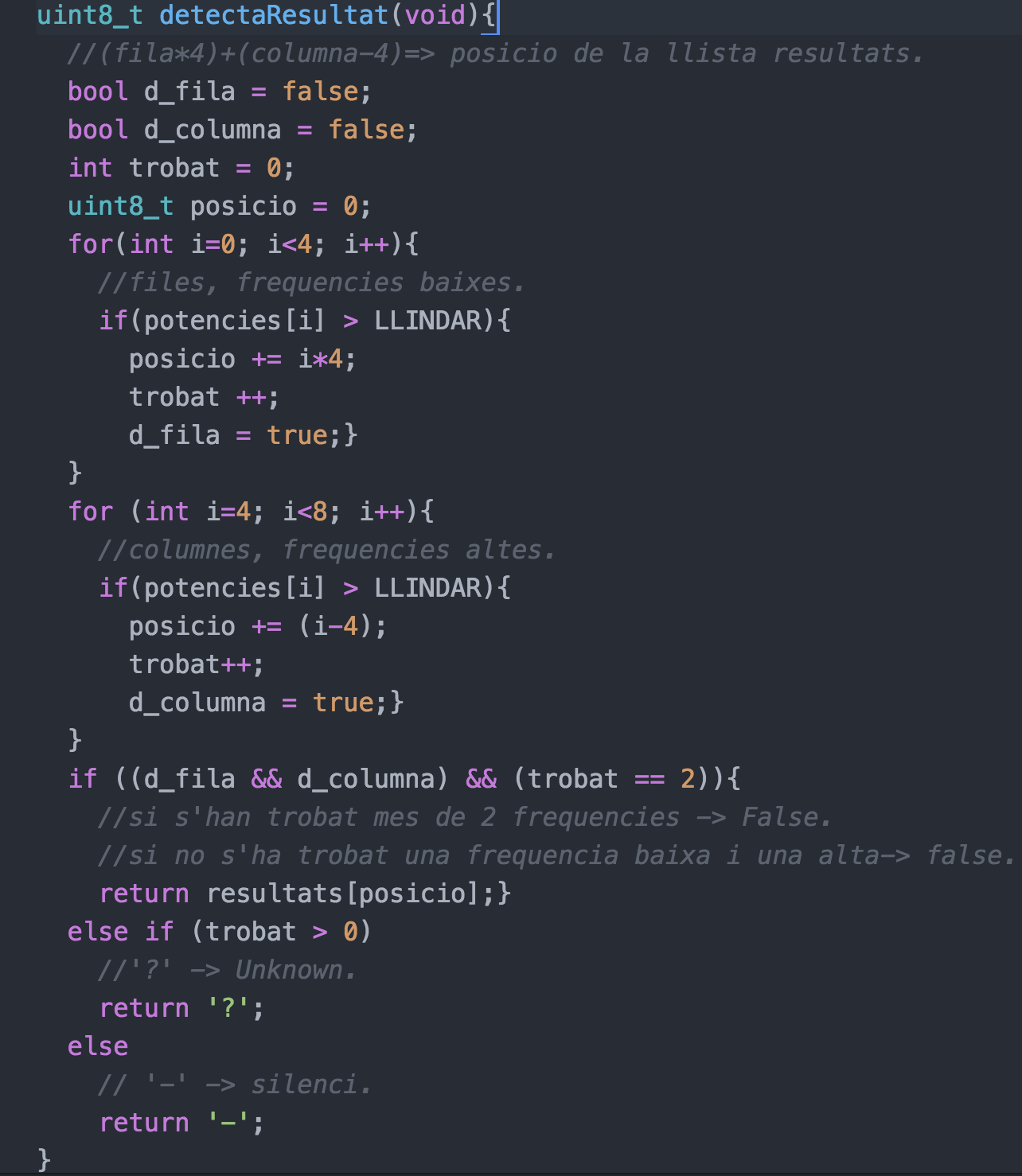
Quin seria el mes fàcil de fer? Optim?

Si vulguessim fer 2 canals quins dispos podríem fer servir? (aquesta no la posem però la pensem)

Imatges:



Font 1: Mostra del càlcul dels Coeficients amb Octave.



Font2: Funció detectaResultat.