

## 5 DESARROLLO DEL PROYECTO

### 5.1 Introducción

En este capítulo se presentan los pasos que se siguieron en cada una de las etapas necesarias para modular en amplitud señales analógicas mediante el procesador TMS320C50. Dado que se puede modificar la frecuencia de modulación (frecuencia de la portadora), se describe el desarrollo de una interfase realizada en java que facilita al usuario modificar la frecuencia de la misma. }

### 5.2 Inicialización de la tarjeta DSK

Antes de ejecutar un algoritmo de procesamiento digital de señales lo más importante es la inicialización de la tarjeta DSK. La inicialización se realiza en software. En caso de que no se inicialicen los parámetros correspondientes las aplicaciones no correrán adecuadamente. Para inicializar la tarjeta se deben llevar a cabo los pasos que se mencionan a continuación [16]:

- Inicializar el procesador digital de señales TMS320C50.
- Inicializar el temporizador y el puerto serial.
- Inicializar el circuito de interfase analógica AIC TLC32040.

#### 5.2.1 Inicialización del procesador digital de señales TMS320C50

Para ejecutar el algoritmo que realiza la modulación, es necesario inicializar el procesador. Generalmente, la inicialización se lleva a cabo cuando el procesador se *resetea* (aplicando un nivel bajo a la entrada *reset*). Los bits del apuntador de vectores de interrupción son

limpiados, es decir, los vectores se mapean a la página 0 en la memoria de programa. Por lo tanto, el vector de *reset* siempre reside en memoria de programa en la dirección 0, en esta locación se encuentra la instrucción que manda a la rutina de inicialización del sistema. Para configurar el procesador después del *reset*, se deben inicializar las siguientes funciones internas [16]:

- Interrupciones
- Memoria
- Registros auxiliares y apuntador de registro auxiliar (ARP)
- Apuntador de página de memoria de datos (DP)

### **5.2.2 Inicialización del temporizador y el puerto serial**

El temporizador se inicializa modificando el contenido del TCR (Timer Control Register). En el caso del modulador en amplitud el temporizador está configurado para una frecuencia de 10.368 MHz.

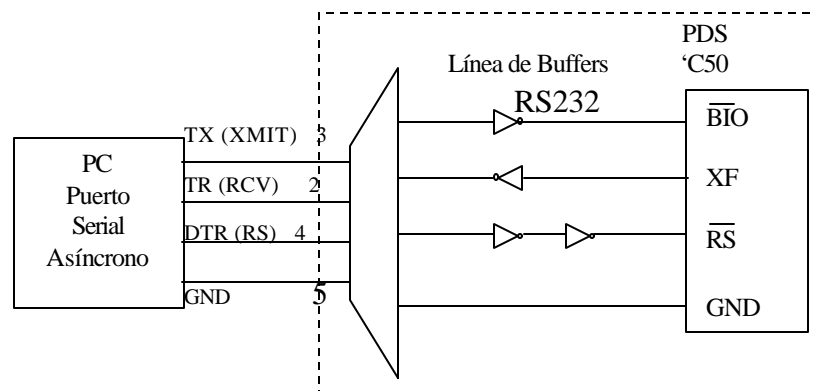
El puerto serial es un puerto *full-duplex* (bidireccional) que provee comunicación directa con dispositivos como codecs, convertidores D/A y A/D y otros sistemas seriales. El puerto serial opera a través de 3 registros mapeados en memoria [14]:

- Registro de control de puerto serial (SPC Serial Port Control).
- Registro de transmisión de datos (DRR Data Receive Register).
- Registro de recepción de datos (DXR Data transmit Register).

El puerto serial del 'C50 se inicializa modificando el registro de control de puerto serial (SPC) del procesador. Para poder recibir y transmitir datos del TLC32040 el bit FSM (Frame Sync Mode) del SPC debe ser 1. Los bits XRST y RRST (*transmit reset* y *receive reset*) *resetean* el transmisor y receptor [16].

La forma en que se comunican la tarjeta DSK y la computadora es utilizando el puerto serial de esta última. El puerto serie de la computadora es compatible con el estándar RS-232. El puerto serial de la computadora es un dispositivo asíncrono. En una transmisión asíncrona, un bit identifica su bit de comienzo y 1 o 2 bits identifican su final, no es necesario ningún carácter de sincronismo. Los bits de datos son enviados al receptor después del bit de *start*. Dependiendo de la configuración de la transmisión un bit de paridad es enviado después de cada bit de datos. Se utiliza para corregir errores en los caracteres de datos. Finalmente 1 o 2 bits de stop son enviados [17].

En la Figura 5-1 se ilustra las conexiones entre la tarjeta DSK y el puerto serial RS-232.



**Figura 5-1 Conexiones de la tarjeta DSK con el puerto RS-232**

### 5.2.3 Inicialización del circuito de interfase analógica TLC32040

El AIC es la interfase del mundo analógico externo con el mundo digital de la tarjeta DSK y se conecta al procesador a través del puerto serial. Para que el procesador se comunique con el AIC se debe inicializar el reloj del procesador y el puerto serial, una vez que se han inicializado los parámetros anteriores se debe *resetear* el TLC32040. Posteriormente se procede a inicializar los siguientes parámetros:

- Registro de control: para establecer la ganancia, sincronización, realimentación, etc.

A continuación se muestra la definición del registro de control:

G1	G0	SY	AX	LB	BP	xx	xx
----	----	----	----	----	----	----	----

G1/G0: bits de control de ganancia

SY: 0/1 para transmisión y recepción asíncrona/síncrona

AX: 0/1 deshabilita/habilita las entradas auxiliares AUX IN+ y AUX IN-

LB: 0/1 deshabilita/habilita la función de realimentación

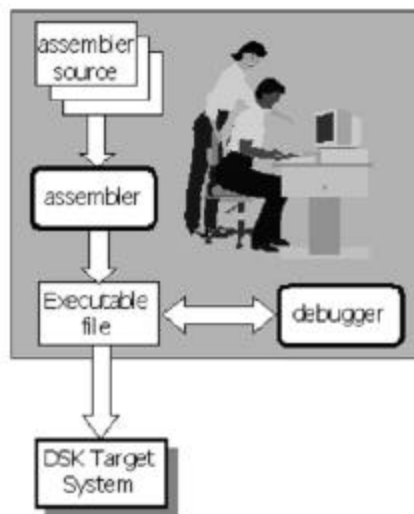
BP: 0/1 deshabilita/habilita el filtro pasabanda

- registros TA y RA del contador A (si es modo síncrono RA se deshabilita)
- registro TB y RB del contador B (si es modo síncrono RB se deshabilita)

Los valores de los registros de transmisión TA y TB determinan la frecuencia de conversión del convertidor D/A, mientras que los registros de recepción RA y RB determinan la frecuencia de conversión del convertidor A/D.

### 5.3 Desarrollo del programa que realiza la modulación en amplitud

La tarjeta DSK incluye el ensamblador y depurador (*debugger*) que le permite al usuario crear sus propias aplicaciones y correrlas en la tarjeta. El ensamblador es una interfase sencilla y fácil de usar que traduce archivos fuente en lenguaje ensamblador a un archivo ejecutable (lenguaje de máquina) para ser usado por la tarjeta. Una vez que se tiene el archivo ensamblado se procede a correrlo en la tarjeta [14]. La Figura 5-2 muestra el proceso anterior [18].



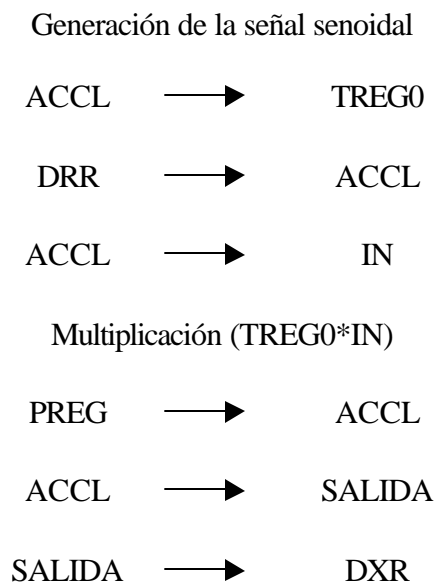
**Figura 5-2 Proceso para correr una aplicación en la tarjeta DSK**

El programa del modulador en amplitud se escribe en lenguaje ensamblador en un archivo de texto, el cual se guarda con extensión .asm para que posteriormente se ensamble.

Una vez que se han inicializado todos los parámetros explicados en la sección 5.2 se procede a llevar a cabo la modulación en amplitud que como se vio en el capítulo 2, consiste en multiplicar la portadora (señal senoidal) por la señal de información. Por lo tanto, el primer paso es generar en el procesador una señal senoidal de frecuencia variable

cuyo algoritmo se obtuvo de un generador de funciones proporcionado por Texas Instruments.

El procedimiento para realizar la modulación en amplitud se presenta a continuación:



Como se observa, una vez que se ha generado la señal senoidal se almacena en la parte baja del acumulador (ACCL), posteriormente se manda al registro TREG0 que almacena temporalmente el multiplicando. Una vez hecho lo anterior se limpia el contenido del ACC y del registro PREG para prepararlos para la multiplicación. Para recibir la señal a modular (entrada) se usa el puerto serial. La señal se recibe en el registro DRR que posteriormente se guarda en un registro de memoria de datos inicializado como IN. La multiplicación se lleva a cabo con la instrucción *MPY IN* que realiza la multiplicación del

contenido del registro TREG0 por el contenido del registro IN. El producto se almacena en el registro PREG. En este momento ya se tiene la señal modulada pero no se puede mandar al DXR para transmitir la señal ya que el PREG es de 32 bits y el DXR de 16 bits, por lo que se procedió a almacenar los bits más significativos del contenido del registro PREG en otro registro de memoria de datos llamado SALIDA. Finalmente, la señal modulada se manda al convertidor a través del registro DXR.

Para variar la frecuencia de la señal portadora se modifica el valor de COEFF. Con los valores de  $TA=TB=18$  y  $RA=RB=15$  y con un valor del registro de control  $AIC\_CTR=8h$  se obtiene un rango de frecuencias de 1428Hz-2856Hz, el archivo que contiene el programa con los valores de registros mencionados se llama MOD1.asm. En la Tabla 5-1 se muestran las frecuencias obtenidas variando el valor de COEFF en el archivo MOD1.asm. En la Tabla 5-2 se presentan los valores de COEFF para un rango de frecuencias de 2857Hz-6666Hz modificando los valores de los registros a  $TA=RA=9$  y  $TB=RB=15$  que se encuentran en el archivo MOD2.asm.

En el apéndice A se incluyen los archivos MOD1.asm y MOD2.asm.

Cabe destacar que en la Tabla 5-1 se tomó el intervalo de 13000 a 27000 ya que en éste la frecuencia tiene un comportamiento aproximadamente lineal, de igual manera, en la Tabla 5-2 se toma el intervalo de 14000 a 29000.

Para calcular el valor de COEFF para una determinada frecuencia de modulación ( $f_c$ ) se aproximó cada intervalo a una línea recta como se observa en la Figura 5-3.

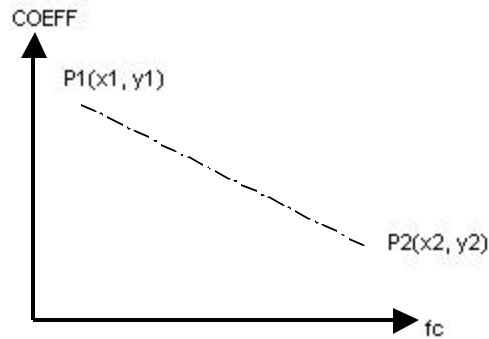
<b>COEFF</b>	<b>Frecuencia (Hz)</b>
31000	3030
30000	3125
29000	1250
28000	1428
27000	1428
26000	1666
25000	1666
24000	1818
23000	1923
22000	2083
21000	2173
20000	2272
19000	2380
18000	2500
17000	2500
16000	2631
15000	2702
14000	2777
13000	2857
12000	2941
11000	3030

**Tabla 5-1 Valores de COEFF en MOD1.asm**

<b>COEFF</b>	<b>Frecuencia (Hz)</b>
29000	2857
28000	3225
27000	3571
26000	3846
25000	4166
24000	4347
23000	4761
22000	5000
21000	5263
20000	5555
19000	5555
18000	5882
17000	6060
16000	6451
15000	6451
14000	6666
13000	6493
12000	7142

**Tabla 5-2 Valores de COEFF en MOD2.asm**





**Figura 5-3 Recta para calcular COEFF a la frecuencia de modulación deseada**

Como se observa en la Figura 5-3  $x_1$  y  $x_2$  son los valores de frecuencia límite en cada intervalo, mientras que  $y_1$  y  $y_2$  corresponden a los valores de COEFF. Por ejemplo, si se desea modular a una frecuencia de 2000 Hz, se eligen de la tabla 5-1 las coordenadas de los puntos 1 y 2, los cuales serían P1(1923, 23000) y P2(2083, 22000). Se calcula la pendiente de la recta ( $m$ ) con la ecuación siguiente:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (5.1)$$

Aplicando la fórmula anterior se obtiene una pendiente de  $-6.25$ .

El siguiente paso es obtener el valor de COEFF que se obtiene sustituyendo en la ecuación de la línea recta (ecuación 5.2) el P1( $x_1$ ,  $y_1$ ),  $x=fc$  y  $y=COEFF$ , con lo cual se obtiene la fórmula para calcular COEFF (ecuación 5.3):

$$y - y_1 = m(x - x_1) \quad (5.2)$$

$$COEFF = m(fc - x_1) + y_1 \quad (5.3)$$

Una vez que se tiene la fórmula para obtener el valor de COEFF se procede a sustituir en la misma el P1(1923, 23000), la pendiente y la frecuencia de modulación ( $fc$ ) como se muestra:

$$\text{COEFF} = -6.25(2000-1923) + 23000$$

$$\text{COEFF} = 22518.75$$

Por lo tanto, redondeando COEFF, el valor que se va a colocar cuando se inicialice COEFF es 22519. Con el valor mencionado se va a obtener una señal modulada a 2000 Hz. Para poder observar la señal modulada se debe ensamblar el archivo y posteriormente correrlo en la tarjeta.

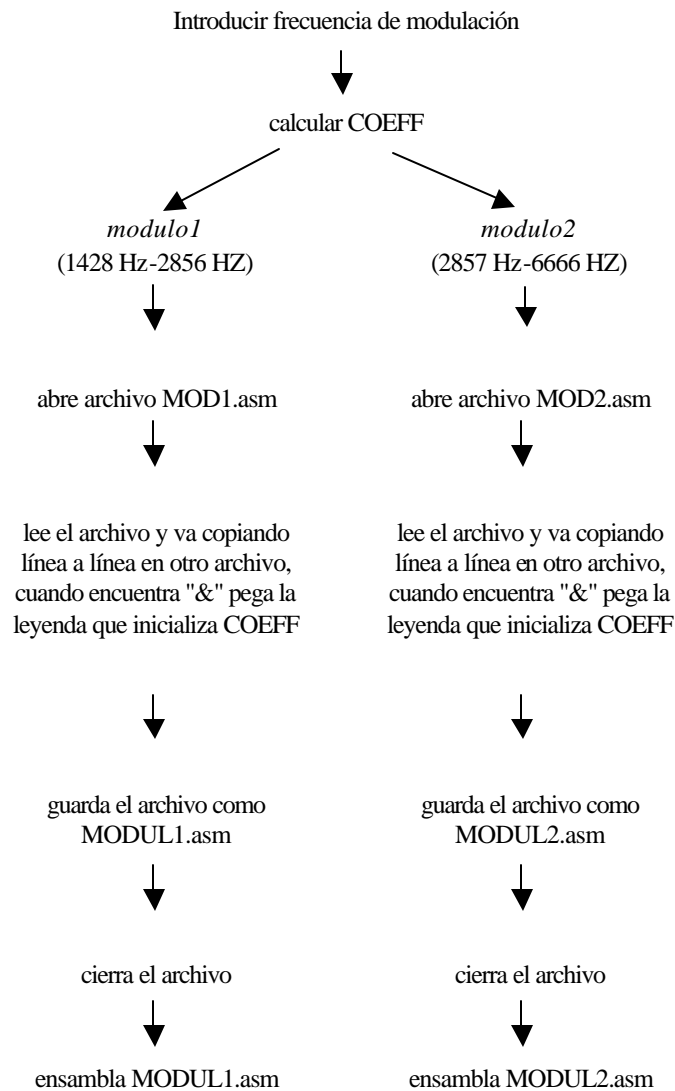
El procedimiento que se realizó anteriormente se debe llevar a cabo cada vez que se desee cambiar la frecuencia de modulación, es decir, se debe buscar en qué intervalo se encuentra la fc deseada para calcular COEFF, después pegarlo en el archivo y por último ensamblarlo y correrlo; por lo que para evitar al usuario todo el trabajo que implica variar la frecuencia de la portadora se desarrolló una interfase en java que simplifica el proceso anterior. En la siguiente sección se detalla la interfase.

#### **5.4 Desarrollo de la interfase en java**

La interfase que facilita al usuario cambiar la frecuencia de modulación se realizó en java, que fue desarrollado por la compañía Sun Microsystems. Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. Actualmente, es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Una de las razones por las cuales se eligió java es que es un lenguaje de programación potente y universal, además de que es un software gratuito que se puede obtener de Internet [19].

Como se ha mencionado, el programa de java simplifica el procedimiento para variar la frecuencia de la portadora, para esto, se realizó una ventana que pide la frecuencia de modulación y que permite calcular el valor de COEFF, después se hace una comparación del valor introducido por el usuario ya que si se cumple alguna de las siguientes condiciones  $f_c = 1427 \text{ Hz}$  o  $f_c = 6667 \text{ Hz}$  se encuentra fuera del rango de modulación.

A continuación se muestra el diagrama que se siguió para la interfase en java:



**Diagrama 5-1 Diagrama que muestra el proceso para realizar la interfase en Java**

Si 1428 fc 2856 (rango 1) se va a la subclase *modulo1* que calcula el valor de COEFF de la misma forma en que se explicó en la sección anterior, por lo que existe un método llamado *calculaPendiente* que calcula la pendiente tomando como base los valores mostrados en la tabla 5-1, después se calcula el valor de COEFF con la ecuación 5-3. Para el caso en el que 2857 fc 6666 (rango 2), se manda a la subclase *modulo2* pero en este caso se toman como base los valores de la Tabla 5-2.

Hasta el momento las subclases *modulo1* y *modulo2* sólo calculan COEFF por lo que el siguiente paso es pegar ese valor en la sección del programa donde se inicializan los valores de memoria de datos. Para hacer lo anterior, se toman como base los archivos MOD1.asm y MOD2.asm que contiene el programa que realiza la modulación y que se muestra en el apéndice A. Como se observa en el apéndice A, los archivos MOD1.asm y MOD2.asm no inicializan COEFF sino que en su lugar hay un “&” que sirve para identificar en dónde se pegará la leyenda que inicializa COEFF y que se muestra a continuación:

COEFF .word (valor calculado)

Ya sea que COEFF haya sido calculado por *modulo1* o *modulo2*, se manda a la subclase *escribir*, la cual abre el archivo MOD1.asm para el caso de *modulo1* o MOD2.asm para *modulo2*, se realiza una lectura del archivo línea por línea, de tal manera que si no encuentra “&” va copiando línea a línea en otro archivo hasta que encuentra, en donde pegará la leyenda para inicializar COEFF, posteriormente termina de copiar el archivo. Una vez que se ha terminado de copiar el archivo se procede a guardarlo con el nombre de

MODUL1.asm (para el rango 1) o MODUL2 (para el rango 2) y finalmente se cierra el archivo.

Como se mencionó en la sección 5-2, para ver la señal modulada se debe ensamblar, lo cual también se realizó en java para evitar teclear el comando que ensambla cada vez que se modifica la frecuencia de modulación. El método que ensambla se obtuvo de un método encontrado en Internet [20] que permite acceder el sistema operativo del sistema. El método escribe el comando siguiente:

*dsk5a MODUL1.asm* o *dsk5a MODUL2.asm* con lo que se ensambla dicho archivo.

En el apéndice B se anexa el programa en java.