

TECNICAS DIGITALES III

1

PRACTICA 2

USO DE QEDESIGN LITE FOR 56800/E EN CONJUNTO CON CODEWARRIOR PARA EL DISEÑO E IMPLEMENTACIÓN DE UN FILTRO DIGITAL

AUTOR: FERNANDO CEAGLIO

OBJETIVOS

2

- Utilizar las herramientas de desarrollo a nivel software y hardware para la implementación de un Filtro Digital Paso Bajos.

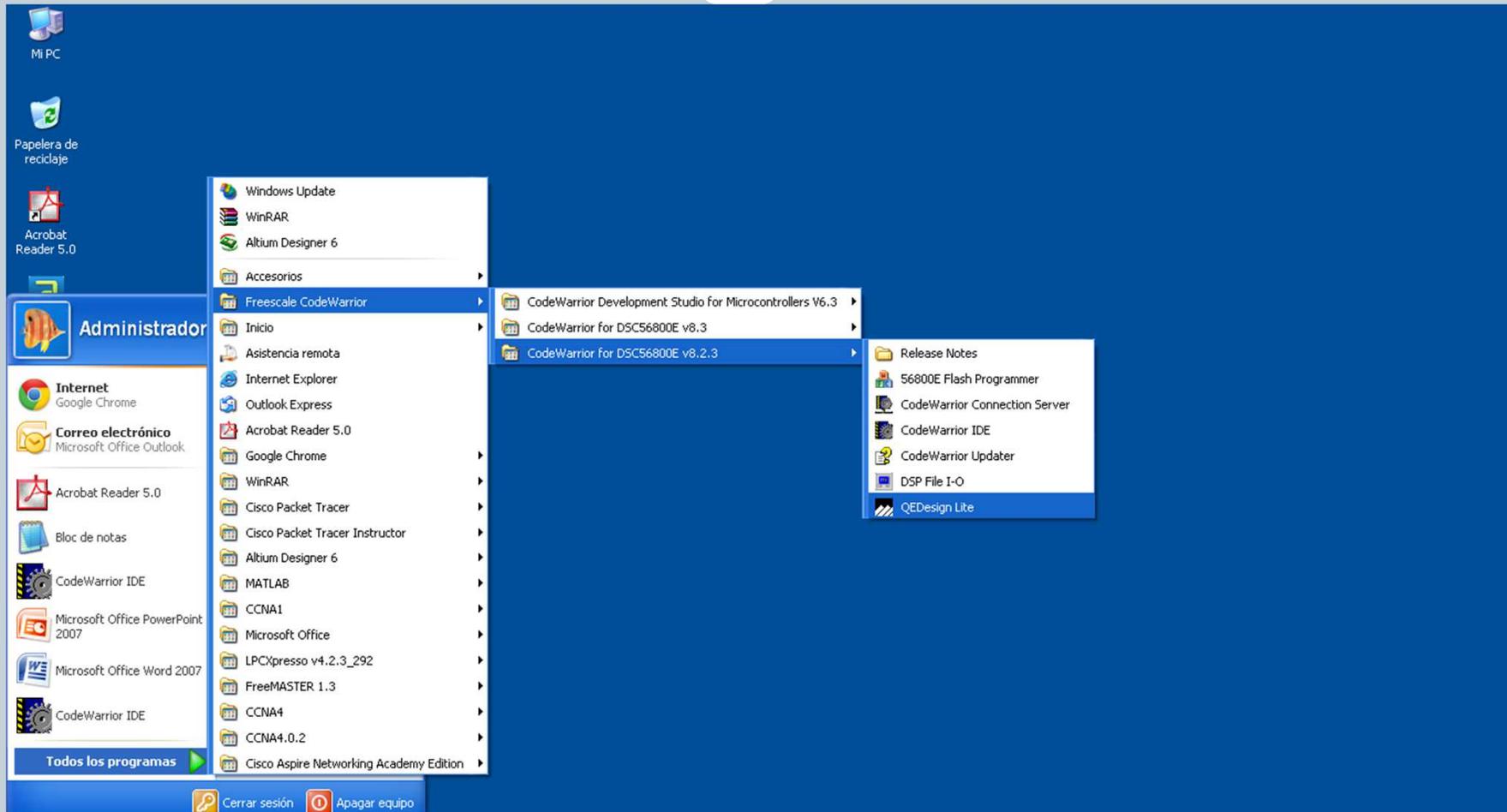
INTRODUCCION

3

- Para esta practica utilizamos en la herramienta QEDesign Lite para 56800/E para generar los coeficientes del Filtro Digital y luego los incorporaremos a nuestro código para poder implementar el mismo.

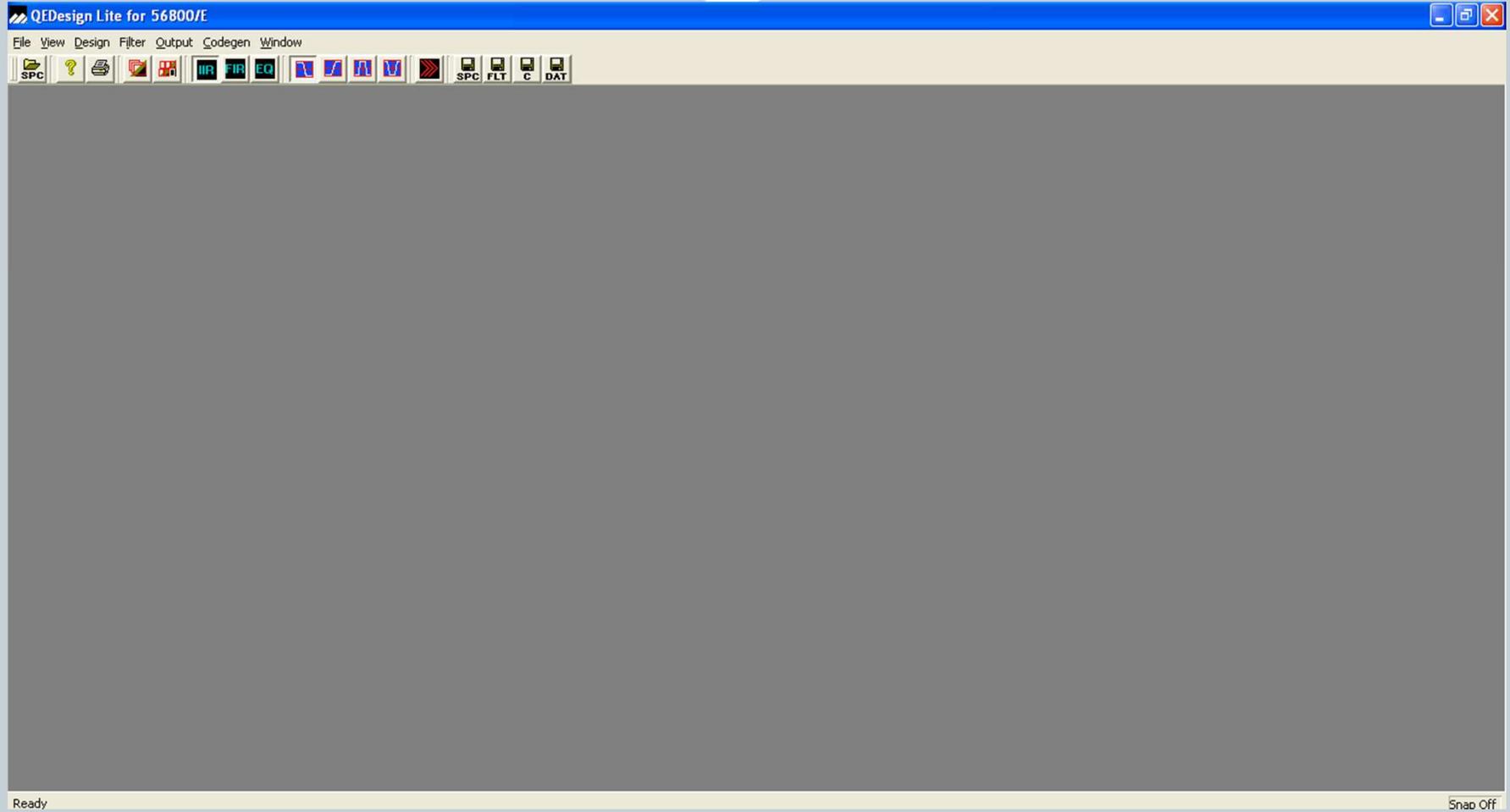
UBICACIÓN DE QEDesign Lite

4



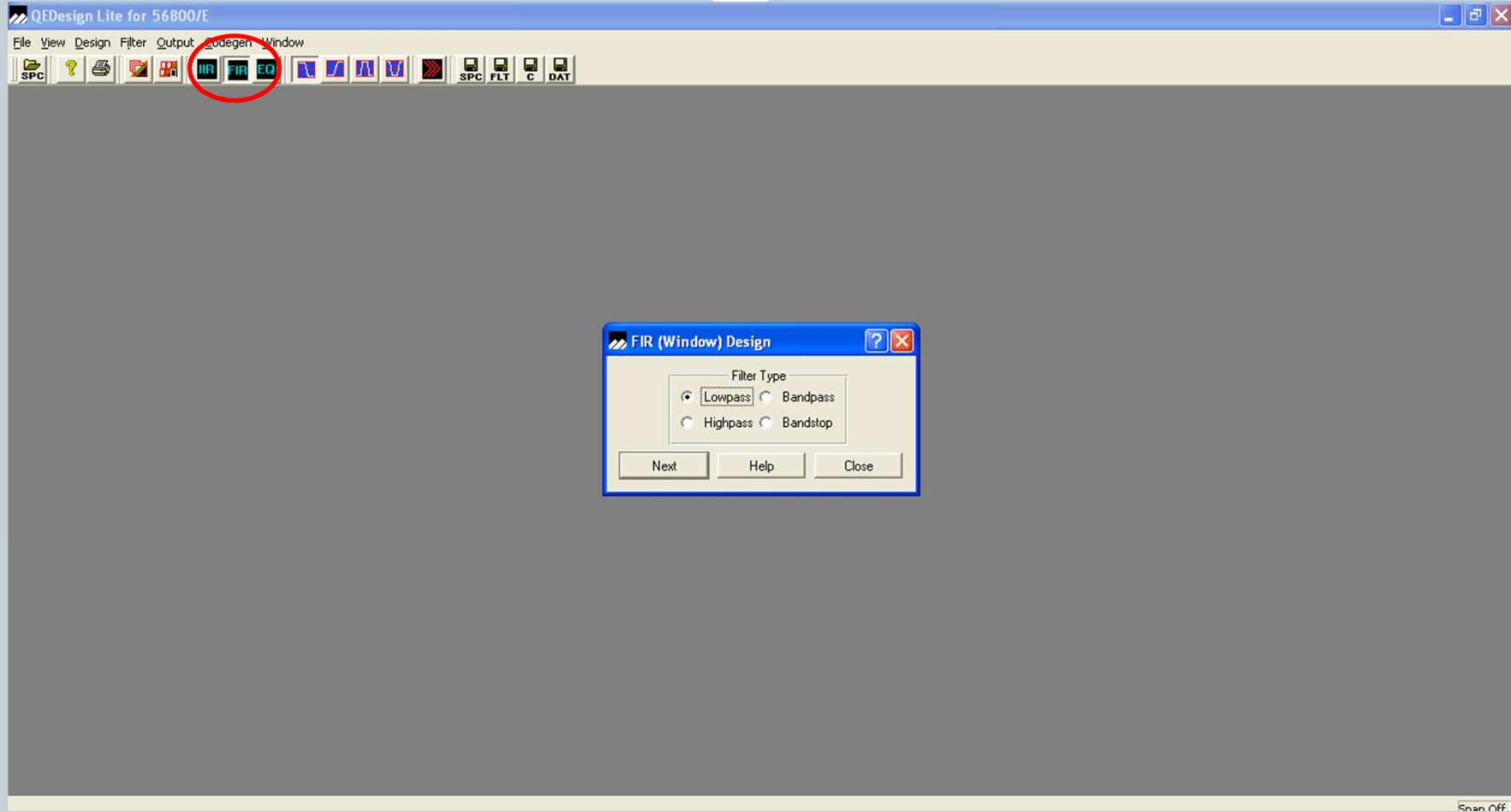
PANTALLA PRINCIPAL

5



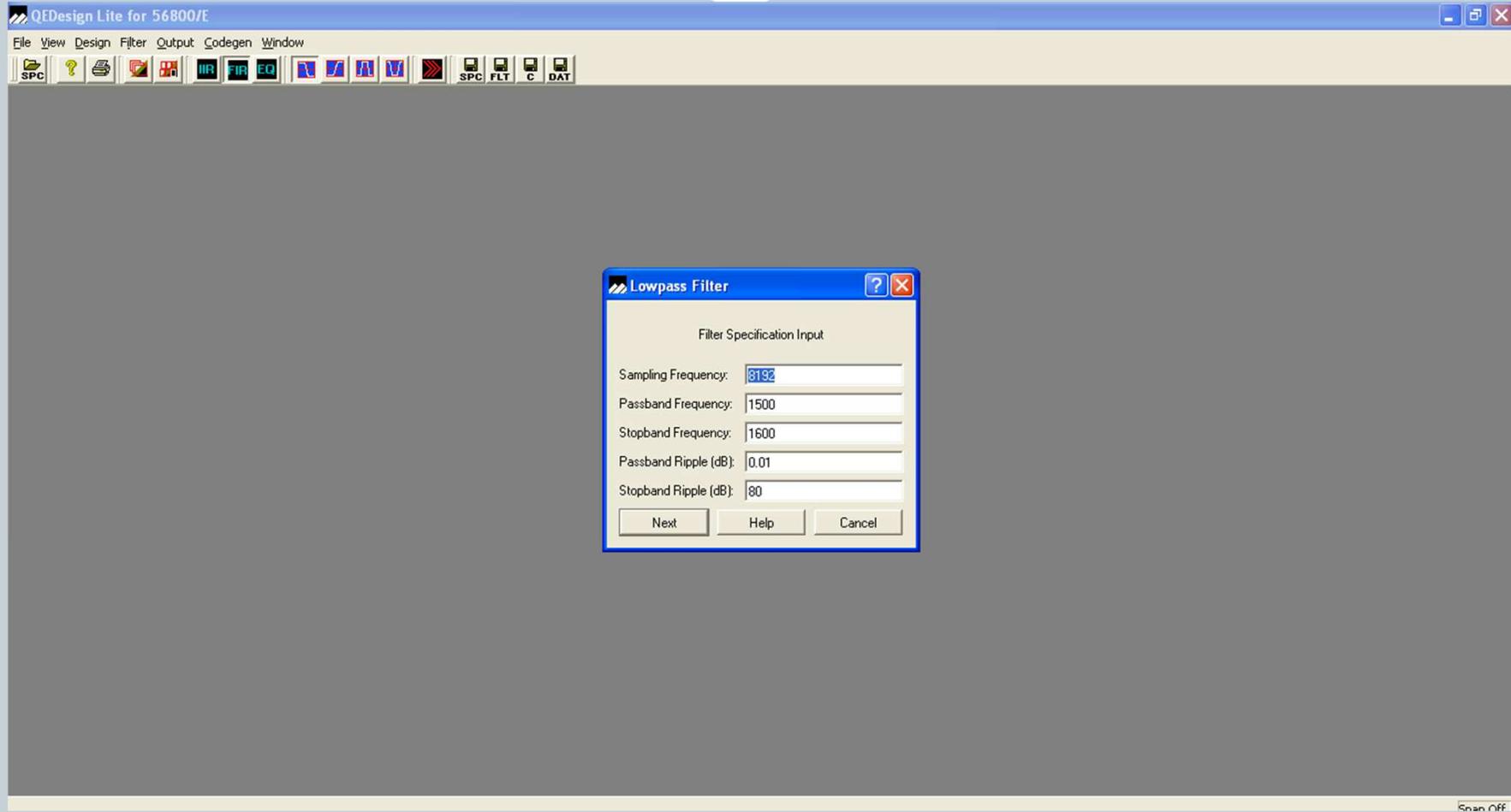
SELECCIÓN DE TIPO DE FILTRO

6



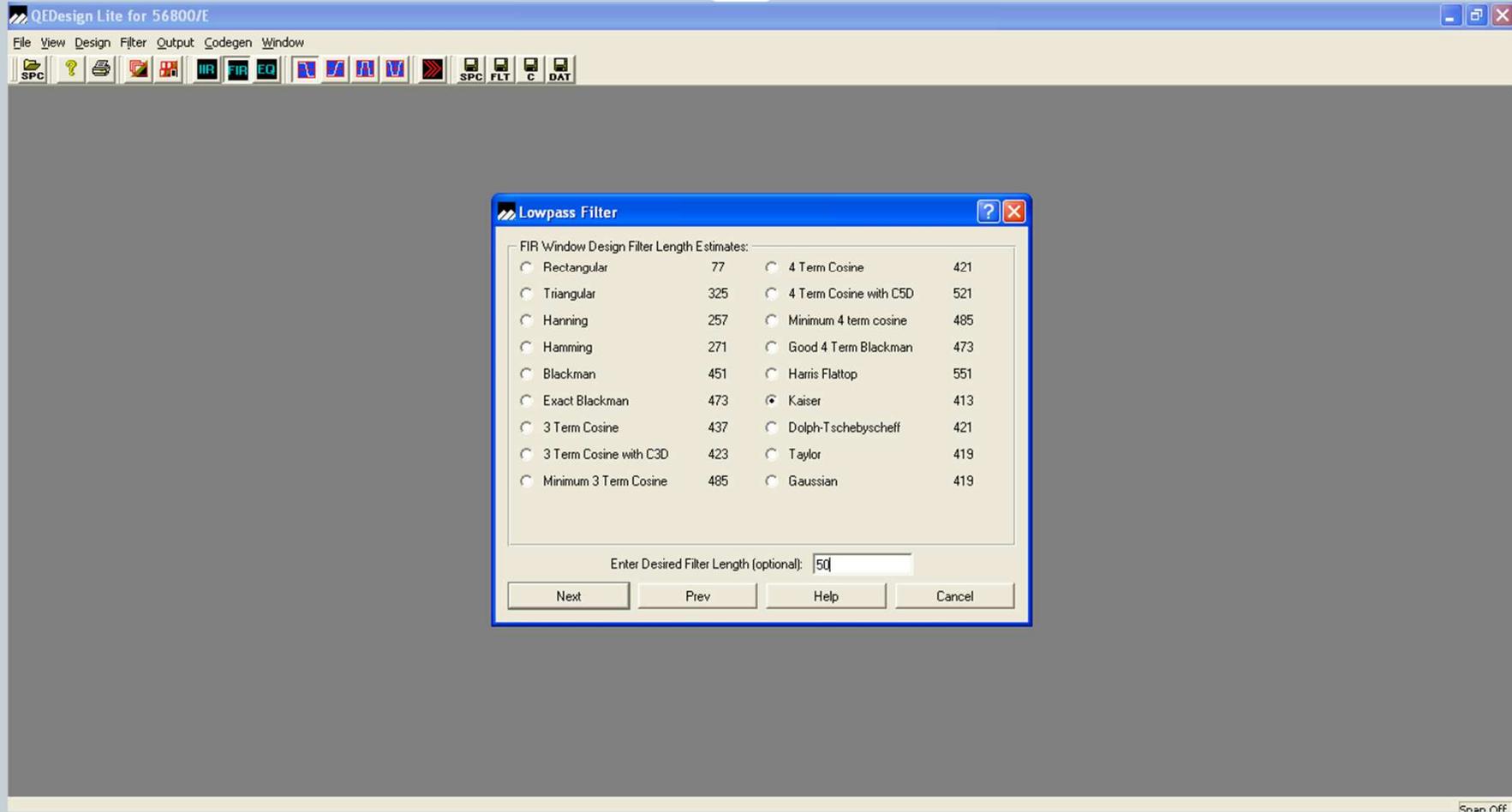
ESPECIFICACIÓN DE ENTRADA DEL FILTRO

7



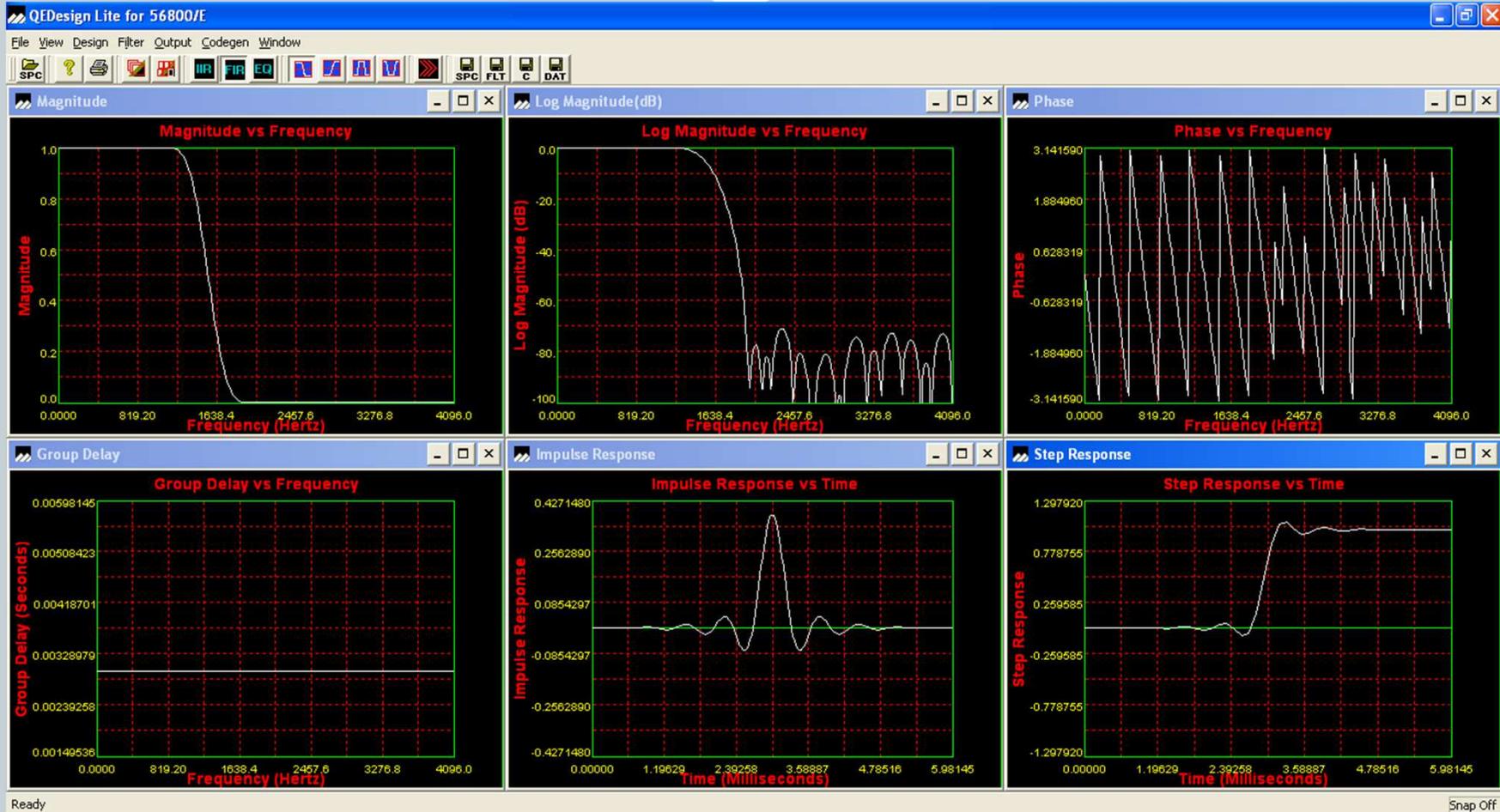
TIPO DE VENTANA Y ORDEN A IMPLEMENTAR

8



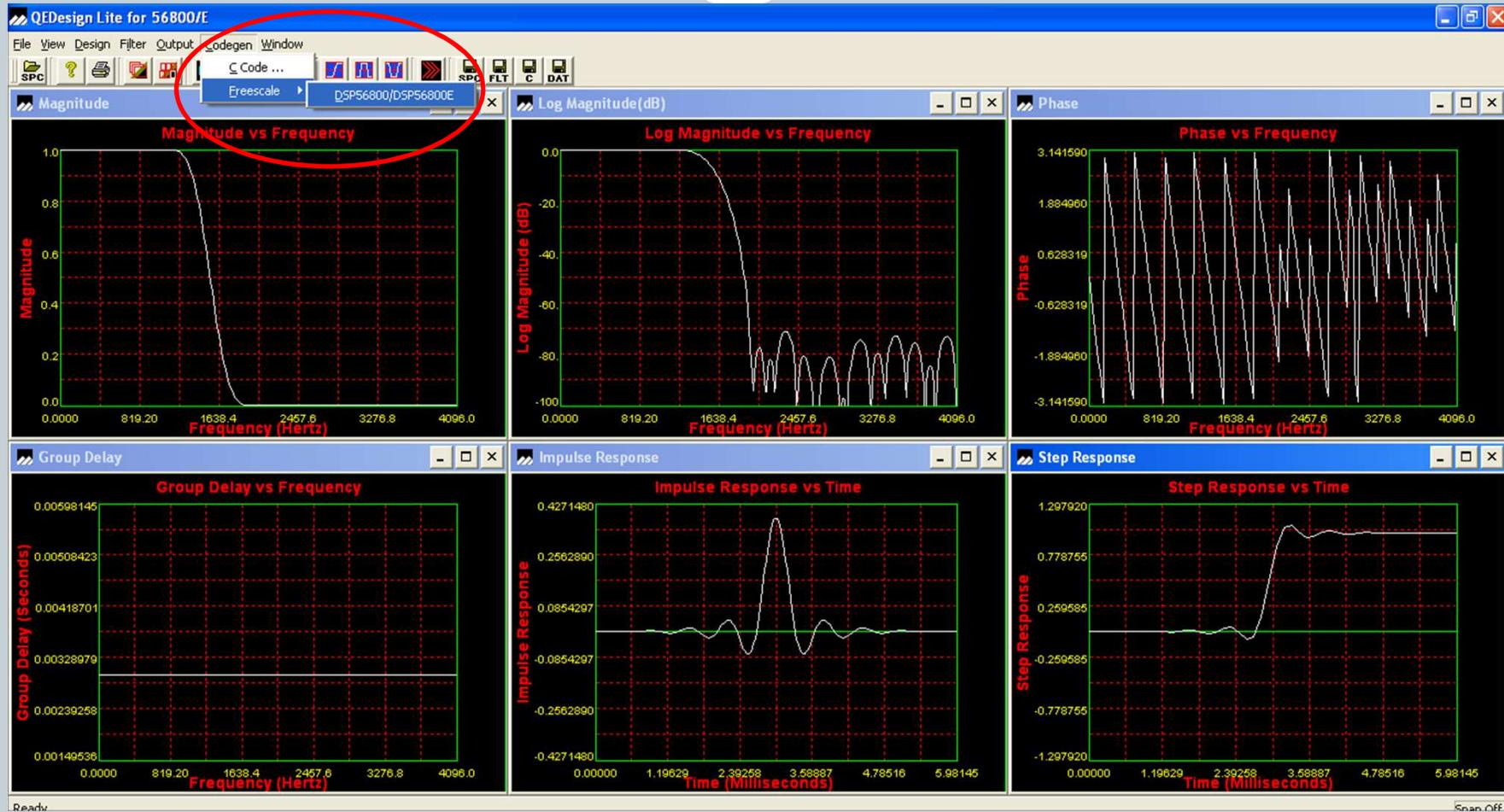
CARACTERIZACION EN TIEMPO Y FRECUENCIA DEL DISEÑO

9



EXPORTANDO LOS COEFICIENTES A NUESTRO CODIGO

10



GUARDANDO ARCHIVO DE COEFICIENTES

11

The screenshot displays the QEDesign Lite for 56800/E software interface. A central dialog box titled "Create DSP56800 Filter Coefficients" is open, showing the file explorer with the "Escritorio" (Desktop) selected. The dialog includes fields for "Nombre:" (Name) set to "CoeefFPB" and "Tipo:" (Type) set to "C Filter Coefficients (*.h)". Buttons for "Guardar" (Save) and "Cancelar" (Cancel) are visible.

Surrounding the dialog are several analysis plots:

- Magnitude vs Frequency:** Shows the magnitude response of the filter across a frequency range from 0.0000 to 4096.0 Hz.
- Phase vs Frequency:** Shows the phase response of the filter across the same frequency range.
- Group Delay vs Frequency:** Shows the group delay in seconds versus frequency in Hertz.
- Impulse Response vs Time:** Shows the impulse response of the filter over time in milliseconds.
- Step Response vs Time:** Shows the step response of the filter over time in milliseconds.

The software interface includes a menu bar (File, View, Design, Filter, Output, Codegen, Window) and a taskbar at the bottom with the Windows Start button and several open applications: "Inicio", "TP7 v1", "DSP - PRACTICA 2", "TP7_KiDemoMotorol...", and "QEDesign Lite for 568...". The system tray shows the time as 6:00.

USO DE LOS COEFICIENTE

12

The screenshot displays the Metrowerks CodeWarrior IDE interface. On the left, a project browser shows the file structure for 'TP7 v1.mcp', including folders like 'support', 'Generated Code', 'Project Settings', 'User Modules', and 'Doc'. The main editor area is split into two windows:

- Events.c**: Shows a C function `TII_OnInterrupt(void)` that is interrupt-driven. It includes a `#pragma interrupt called` directive and uses `periphMemRead` to read from `ADCA_ADRSLT0`. It also contains a check for `ADC_MAX_BUFFER_SIZE` and a comment indicating it was created by a programmer for a Freescale 56800 series processor.
- TP7_v1.c**: Shows global definitions and variables. It includes `#define` for `PBANDA` (1) and `NUM_SAMPLES` (300). It defines prototypes for `TurnOnLed` and `TurnOffLed`. Global variables include `CFrac16` `FFTInplaceBuf`, `UWord16` `AdcBuffer`, and integer variables for `AdcWriteIndex`, `AdcReadIndex`, and `tipoFiltro`. A filter coefficient array `CoefFIR_PasaBajo` is defined with 7 elements, each represented by a hexadecimal value.

CUESTIONES

13

- A partir de lo visto en clase determinar en el fuente adjunto (Anexo) lo requerido en las siguientes preguntas.
- Paso 1:
 - 1) Determine la función del programa.
 - 2) Que entradas/Salidas usa y para qué las utiliza.
 - 3) Mencione cual es la frecuencia de muestreo del ADC y como se fija en el código de ejemplo.
 - 4) ¿En dónde se guardan los resultados de las conversiones del ADC?
. ¿Qué ocurre cuando se llena el buffer?
 - 5) Proponga el cambio necesario en el cuerpo principal del programa “main.c” si se quiere muestrear una señal por la entrada analógica ANA1.
 - 6) Vinculado a la frecuencia del clock del DSC ¿qué limita la frecuencia a la cual muestrear la señal de entrada?
 - 7) Indique cuál es la actividad repetitiva que realiza el DSC en nuestro caso. Justifique el uso de este tipo de procesadores.

DEBUG – PASO 11

14

The screenshot displays the Metrowerks CodeWarrior IDE interface during a debug session. The main window is titled "target.elf" and shows the following components:

- File Explorer (Left):** Shows a project structure for "TP7_1.mcp" with files like "support", "Generated Code", "Project Settings", "Startup Code", "User Modules", "Events.c", "TP7_1.c", "Events.h", and "Doc".
- Source Window (Center):** Displays the source code for "target.elf" with a message: "Program 'target.elf' is executing. Choose Break from the Debug menu to stop it." The status bar indicates "Line 1 Col 1 Source".
- Stack and Variables (Top Right):** Shows a stack window with "No local variables" and a variables window with "No local variables".
- Target CPU View (Right):** Shows a 3D model of the "56F801FA60" CPU chip.
- Bean Selector (Bottom Right):** A panel titled "Bean Selector" with categories "On-Chip Prph", "Alphabet", and "Assistant". It displays a list of beans for the "56F801FA60" target, including "Digital input/output [4 beans]", "Measurement [10 beans]", "Generation of signals [7 beans]", "Timing [18 beans]", "Communication [29 beans]", "Memory [3 beans]", "External devices & peripherals [3 beans]", and "Motor control [13 beans]".

DEBUG (BKP) – PASO 11

15

The screenshot displays the Metrowerks CodeWarrior IDE interface. The main window shows a C program for a 56F801FA60 CPU. The code includes initialization for system clock, PLL, and clock prescaler. The interface also shows a target CPU model, a stack window, and a bean selector for hardware components.

Code:

```
/** !!! Here you can place your own code before PE initialization !!!  
  
/** ### 56F801FA60 "Cpu" init code ... **/  
/** PE initialization code after reset **/  
/* System clock initialization */  
setRegBitVal(DFIU_CNTL, IFREN, C_Cpu_reg_DFIU_CNTL_bit_IFREN);  
setReg(IOSCTL, *(word *)0x103F); /* Set the trim osc freq  
setRegBitVal(DFIU_CNTL, IFREN, C_Cpu_reg_DFIU_CNTL_bit_IFREN_0)  
setRegBitVal(PLLCR, PRECS, C_Cpu_reg_PLLCR_bit_PRECS); /* Select  
while(getRegBit(PLLSR, PRECS)){} /* Wait for clock stabil  
setReg(PLLCR, (PLLCR_LCKON_MASK | PLLCR_ZSRC0_MASK)); /* Enable  
/* CLKOSR: ??=0, ??=0, ??=0, ??=0, ??=0, ??=0, ??=0, ??=0, ??=0, ??=0,  
setReg16(CLKOSR, C_Cpu_reg_CLKOSR); /* CLK0 = ZCLOCK */  
/* PLLDB: LORTP=2, PLLCOD=0, PLLCID=1, ??=0, PLLDB=29 */  
setReg16(PLLDB, C_Cpu_reg_PLLDB); /* Set the clock prescaler  
while(!getRegBit(PLLSR, LCK0)){} /* Wait for PLL lock */
```

Target CPU [Cpu:56F801FA60]

9 Vss_9 Vss_9 GND Ground

Bean Selector

Categories | On-Chip Prph | Alphabet | Assistant | Quick

[START]

What kind of task do you need to handle?

- Digital input/output [4 beans]
- Measurement [10 beans]
- Generation of signals [7 beans]
- Timing [18 beans]
- Communication [29 beans]
- Memory [3 beans]
- External devices & peripherals [3 beans]
- Motor control [13 beans]

Filter: for 56F801FA60 only | Licensed | All questions