

Введение

В современных радиотехнических системах выделяют два класса систем – это аналоговые и цифровые.

В аналоговых радиотехнических системах процессы приема, связи и передачи основаны на аналоговых сигналах, например, в телевизионном вещании, радиовещании, телефонии.

В цифровых системах процессы основаны на цифровых (дискретных) сигналах. Примерами таких процессов являются современные объекты связи, цифровая телефония, цифровое телевидение.

Сигналы могут быть обработаны с использованием аналоговых методов (аналоговой обработки сигналов, или ASP), цифровых методов (цифровой обработки сигналов, или DSP) или комбинации аналоговых и цифровых методов (комбинированной обработки сигналов, или MSP). Выбор методов осуществляется в зависимости от поставленной задачи.

Что касается DSP, то главное отличие его от традиционного компьютерного анализа данных заключается в высокой скорости и эффективности выполнения сложных функций цифровой обработки, таких как фильтрация, анализ с использованием быстрого преобразования Фурье (БПФ) и сжатие данных в реальном масштабе времени.

Система комбинированной обработки может быть реализована в виде печатных плат, гибридных интегральных схем (ИС) или отдельных кристаллов с интегрированными элементами. АЦП и ЦАП рассматриваются как устройства комбинированной обработки сигналов, либо просто на одном кристалле.

DSP позволяет выполнять операции в режиме реального масштаба времени.

1. Применение DSP микропроцессоров

DSP микропроцессоры используются в модемах удаленного доступа (RAS), цифровых сотовых телефонах, при управлении электродвигателями, в кодеках и процессорах обработки сигналов голосовых приложениях.

А. Цифровые сотовые базовые станции

Сотовые телефонные базовые станции формируют основу современной беспроводной сотовой инфраструктуры, кроме того, они должны удовлетворять нескольким стандартам одновременно, например AMPS и CDMA.

Гибкость, высокая производительность и низкая стоимость канала являются основными требованиями к современным базовым станциям. Максимальное использование DSP микропроцессоров в приемопередатчиках позволяет обрабатывать несколько стандартов без необходимости замены аппаратных средств. На примере цифрового приемника: узкополосный или широкополосный (упр. Стр. схема которого показана на рисунке 1.1) DSP микропроцессоры используются для управления АЦП, ЦАП для реализации алгоритмов ЦОС.

Б. Многоканальная интернет телефония (ADSL)

ADSL представляет собой новую технологию высокоскоростной цифровой коммутации и маршрутизации и обработки сигналов. Области их применения для построения, высокоскоростных систем передачи данных, являются одними из самых быстроразвивающихся (рисунке 1.2).

ADSL может передавать данные по обычным телефонным линиям почти в 200 раз быстрее, чем самые современные модемы, и в 90 раз быстрее, чем системы ISDN.

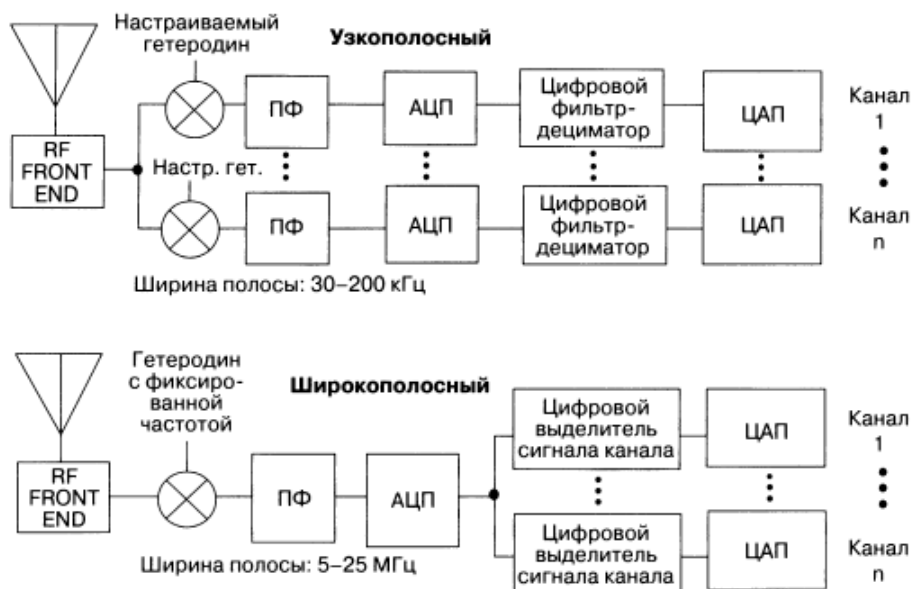


Рисунок 1.1– Узкополосные и широкополосные цифровые приемники для базовых радиостанций



Рисунок 1.2 Структурная схема модема

Ключом к повышению скорости передачи данных по стандарту ADSL является использование усовершенствованных аналоговых решений и сложных алгоритмов цифровой обработки сигналов позволило реализовать данный процесс с помощью DSP **микропроцессоров**.

Таким образом, актуальность использования DSP процессоров в повседневной жизни не оспорима. Для более удобной реализации различных проектов на DSP процессорах используется большое количество программных обеспечений. Самыми распространенными из которых являются C++, VisualDSP, MPLAB. Каждая из них имеет свои особенности: удобство использования в том или ином случае. Так для удобства формирования сигналов предлагается использовать программное обеспечение dsPICWorks.

2. Программное обеспечение для обработки сигналов dsPICWorks

Интегрированная среда программирования dsPICWorks – удобный пакет анализа сигналов для операционных систем Microsoft Windows 9x, NT, 2000 и XP, который предназначен для формирования сигналов различной формы с различными параметрами и упростить работу с DSP **микروпроцессорами**. dsPICWorks поддерживает следующие функции:

- Обширный набор генерируемых сигналов, включая синусоидальный, прямоугольные и треугольные импульсы, ступенчатая и дельта функции, экспоненциальных и шумовых функций. Шум с указанным распределением может быть добавлен к любому сигналу. Полученные сигналы могут быть представлены как 32-разрядные числа с плавающей точкой или 16-разрядные с фиксированной точкой для любой частоты дискретизации. Длина генерируемого сигнала ограничена только доступным дисковым пространством. Сигналы могут быть импортированы или экспортироваться в среду разработки MPLAB и Matlab.

Многоканальные данные могут быть представлены набором функций;

- Арифметические операции и операции цифровой обработки сигналов dsPICworks имеет широкий набор функций цифровой обработки сигналов и арифметических функций. Стандартные функции цифровой обработки сигналов – быстрое преобразование Фурье (FFT) и дискретное косинусное преобразование (DCT), операции свертки и корреляции, децимация сигнала, преобразование частоты дискретизации, интерполяция и цифровая фильтрация. Цифровая фильтрация – важная часть dsPICworks. Здесь используются фильтры, разработанные дочерней программой – dsPICFilterDesign, которые применяются к синтезированным или импортированным сигналам. dsPICworks, также позволяет выполнять такие

преобразования, как ограничение, масштабирование и квантование сигналов, которые важны в реальном практическом анализе алгоритмов цифровой обработки сигналов;

- Возможность одно-, двух- и трехмерного отображения сигналов. Значение сигналов могут быть измерены с помощью курсора мыши. В окне отображаются текущие координаты курсора, а также изменения от предыдущего положения. Спектральная мощность сигнала может быть измерена в определенном диапазоне частот. Имеется поддержка отображения многоканальных и мультиплексированных данных. Есть возможность масштабирования изображений. Кроме того, существует возможность гибкой настройки интерфейса, в частности цветов отображения.

- Импорт/Экспорт данных в формат, совместимый с IDE MPLAB и ассемблером MPLAB ASM30, что позволяет импортировать внешние данные из ASCII или бинарных файлов и экспортировать данные в файл. dsPICworks поддерживает все форматы файлов диалога «Import/Export-Table» в среде MPLAB. Это позволяет вводить реальные данные из MPLAB в dsPICworks для проведения анализа. dsPICworks может также создать файлы ассемблера ASM30, которые могут быть включены в MPLAB.

Особенности

- Большое количество генерируемых сигналов: синус, прямоугольник, треугольник, оконная функция, шум;
- Большое количество алгоритмов ЦОС: БПФ, ДКП, фильтрация, операция свертки, интерполяция, и др.;
- Арифметические функции: алгебраические выражения, масштабирование, ограничение и т.д.;
- 1-, 2- и 3-мерные графики;
- Квантование данных;
- Отображение многоканальных данных;

- Поддержка файлов сценария, позволяющих автоматизировать действия;
- Импорт/экспорт файлов, совместимый с MPLAB IDE;
- Цифровая фильтрация с помощью фильтров, созданных в dsPICDigitalFilterDesign;
- Создание ассемблерного файла ASM30 для экспорта полученных данных непосредственно в ОЗУ dsPIC30F.

3. Среда обработки сигналов dsPICWorks

3.1 Запуск среды проектирования

Запустить dsPICWorks можно через меню программ (Пуск -> Программы -> dsPICWorks-v2.1 -> dsPICWorks-v2.1.exe) или через пиктограмму на рабочем столе.

После запуска отобразится основное окно программы dsPICWorks (рисунок 3.1)

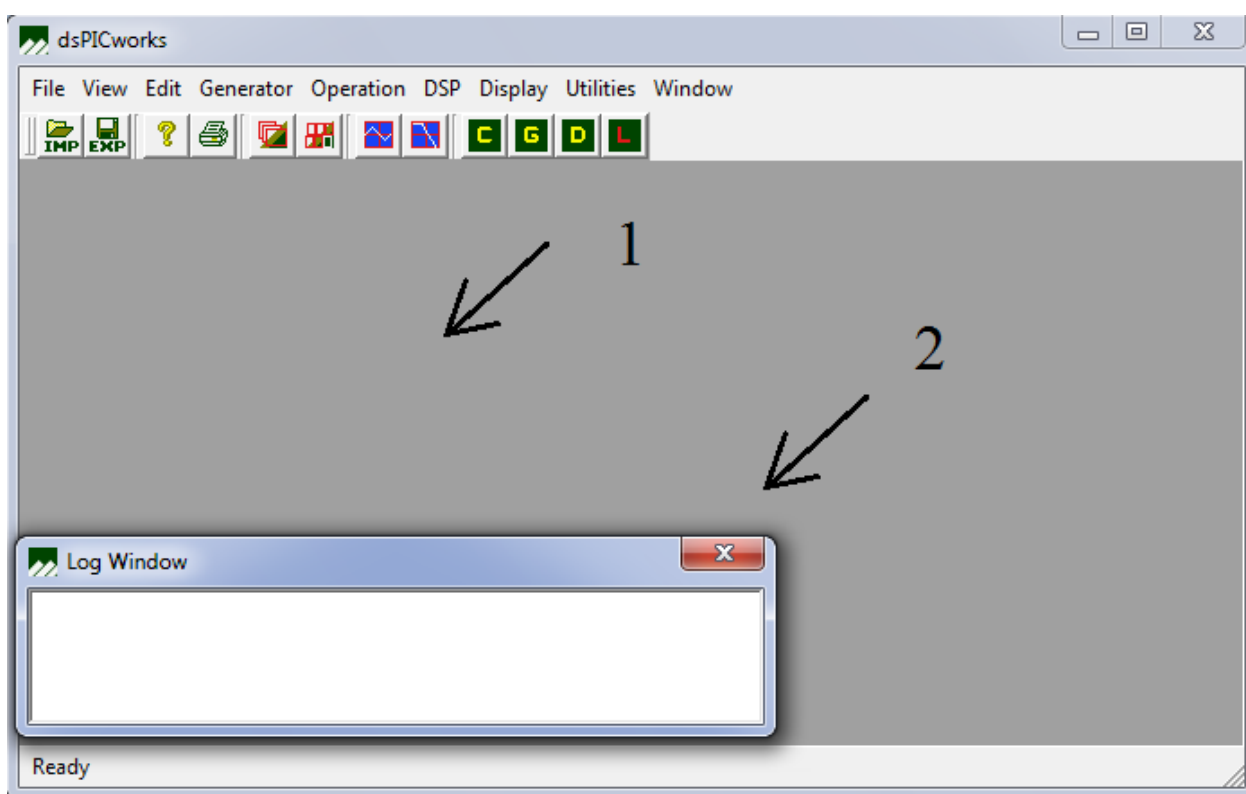


Рисунок 3.1

Рабочий стол среды содержит:

1. Рабочую область, в которой размещаются открытые окна с файлами, диалогами или другой информацией.
2. Окно LogWindows, в котором отображаются все данные о процессе компиляции проекта.

3.2. Описание меню

3.2.1. Вкладка **File/Файл** содержит стандартные файловые операции. В том числе запись/воспроизведение проекта, ввод/вывод данных, печать, выход и информация о программе.

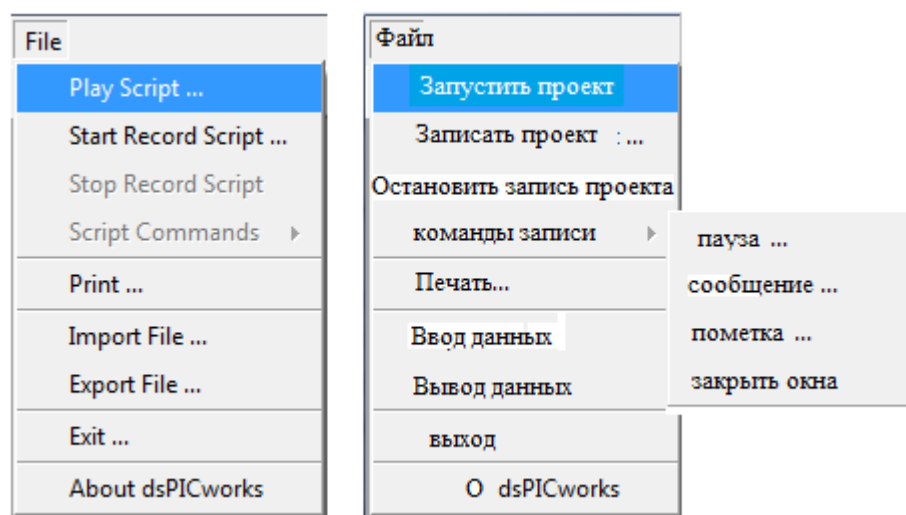


Рисунок 3.2.1

3.2.2 Вкладка **View/Вид** содержит инструменты и строки состояния.



Рисунок 3.2.2

3.2.3 Вкладка **Edit/Редактирование** содержит функции, редактирующие выбранные участки сигнала, по средствам вырезания, копирования, вставки и удаления.

Edit		Редактирование	
Undo	Ctrl+Z	Отменить	Ctrl+Z
Cut	Ctrl+X	Вырезать	Ctrl+X
Copy	Ctrl+C	Копировать	Ctrl+C
Paste	Ctrl+V	Вставить	Ctrl+V
Paste to New File		Вставить в новый файл	
Delete	Del	Удалить	Del
Select All	Ctrl+A	Выбрать все	Ctrl+A

Рисунок 3.2.3

3.2.4 Вкладка **Generator/Генерирование** помогает сформировать сигналы стандартного вида.

Generator	Генерирование
Sinusoidal ...	Синусоидальный
Square ...	Квадратичный
Triangular ...	Треугольный
Swept Sine ...	Охваченный синус
Unit Sample ...	Простой модуль
Unit Step ...	Шаг модуля
Window Functions ...	Функциональное окно
Sinc Function ...	Функция Sinc ...
Ramp Function ...	Функция Ramp ...
Exp Function ...	Функция Exp ...
Noise Function ...	Функция шума ...

Рисунок 3.2.4

3.2.5 Вкладка **Operation/Операции** содержит математические операции во временной области сигнала,

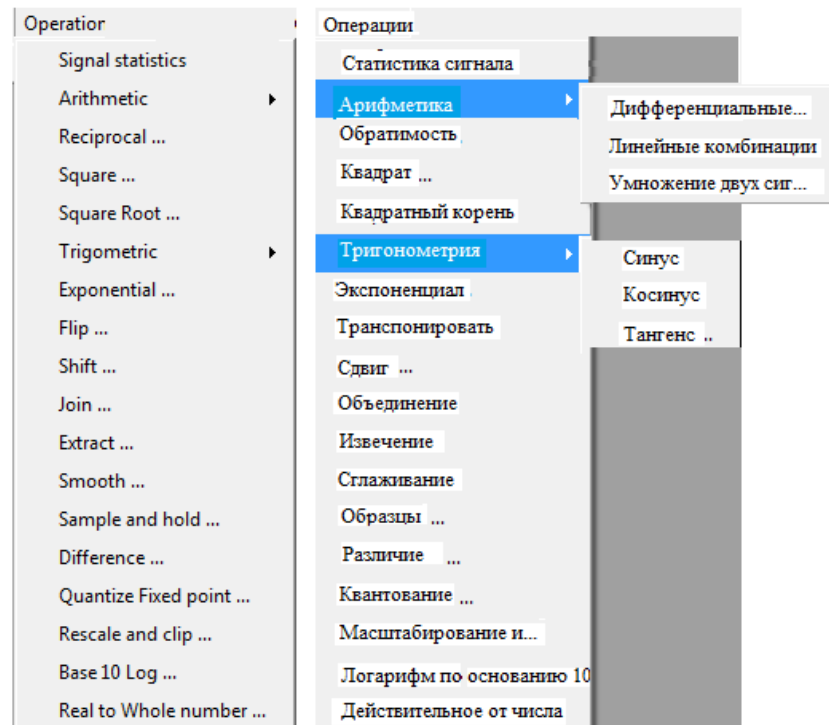


рисунок 3.2.5

3.2.6 Вкладка **DSP** содержит редактирующие операции для временной и частотной областей.

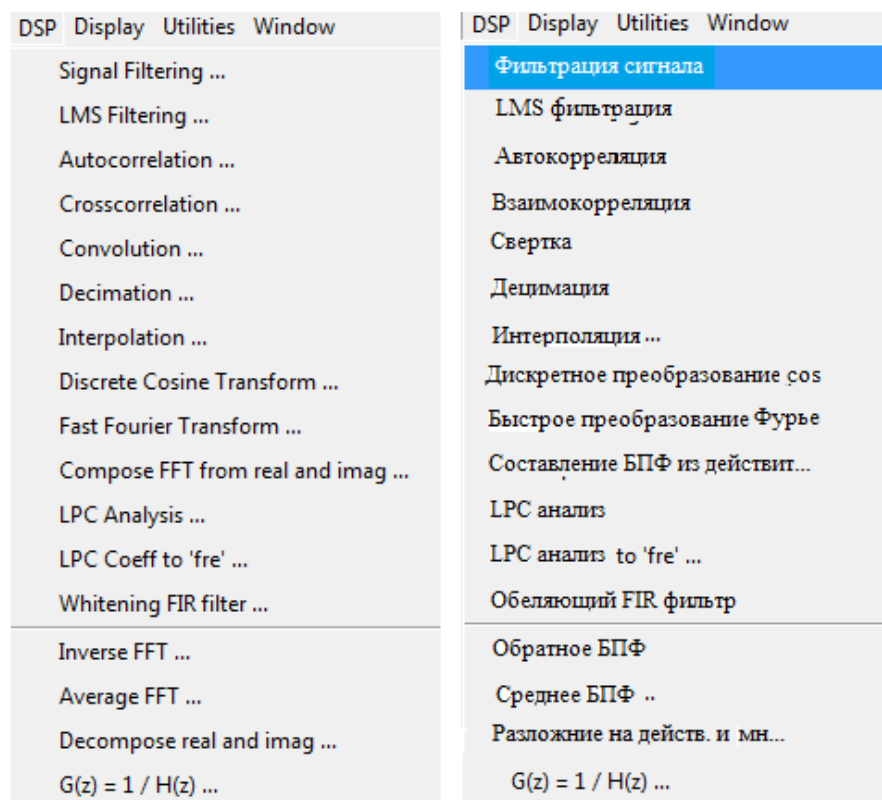


Рисунок 3.2.6

3.2.7 Вкладка Display/Отображение содержит функции отображающие сигнал в более удобном виде для определенных исследований.

Display	Utilities	Window	Отображение
Time file ...			Временной файл
Frequency file, Mag ...			Частотный файл, величина
Frequency file, Power ...			Частотный файл, мощность
Frequency file, Phase ...			Частотный файл, фаза .
Frequency file, Mag 2D ...			Частотный файл, вел 2D .
Frequency file, Mag 3D ...			Частотный файл, вел 3D

Рисунок 3.2.7

3.2.8 Вкладка Utilits/Утилиты содержит системные функции, такие как преобразование формата файла, преобразование типа числа, управление графической частью.

Utilities	Window	Утилиты
Control Center ...		Контрольный центр
<input checked="" type="checkbox"/> Saturate		<input checked="" type="checkbox"/> Насыщать
Wraparound		Полный оборот
ASCII outputs		ASCII вывод :
<input checked="" type="checkbox"/> Binary outputs		<input checked="" type="checkbox"/> Бинарный вывод
ASCII and Binary conversion ...		ASCII и бинарный...
Fixed and float conversation ...		Фиксир. и плавающ.
Demultiplex Signal ...		Демультимплексный сигнал
Multiplex Signals ...		Мультимплексный сигнал

Рисунок 3.2.8

1.2.9 Вкладка **Windows/Окно**

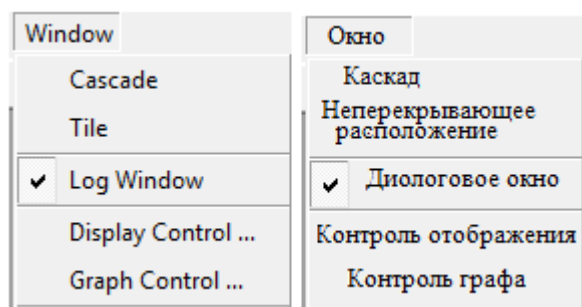


Рисунок 1.2.9

4. Формирование сигналов в dsPICWorks

Для того чтобы начать работу, нужно создать новый или открыть существующий проект.

Рассмотрим этот процесс на примере синусоидального сигнала и манипуляций с ним.

Пример:

Сформируем синусоидальный сигнал с частотой 80 Гц, **обязательно указав имя файла**, в котором будет сохраняться все промоделированные результаты.

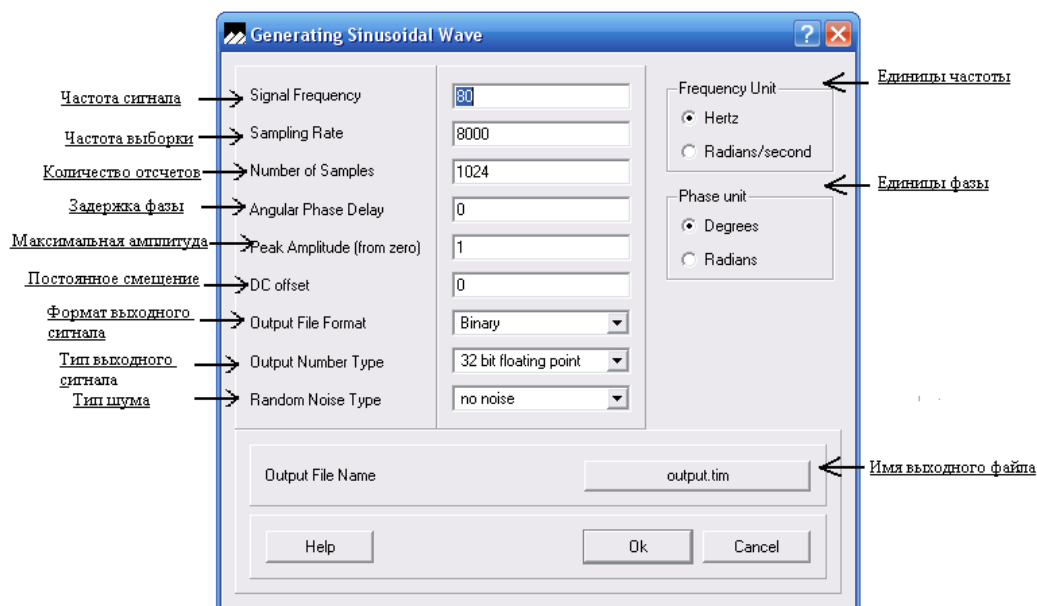


Рисунок 4.1

Выходной сигнал может быть не только 32-битным с фиксированной запятой, но и также 16-битным дробным с плавающей запятой.

Получаем сформированный сигнал:

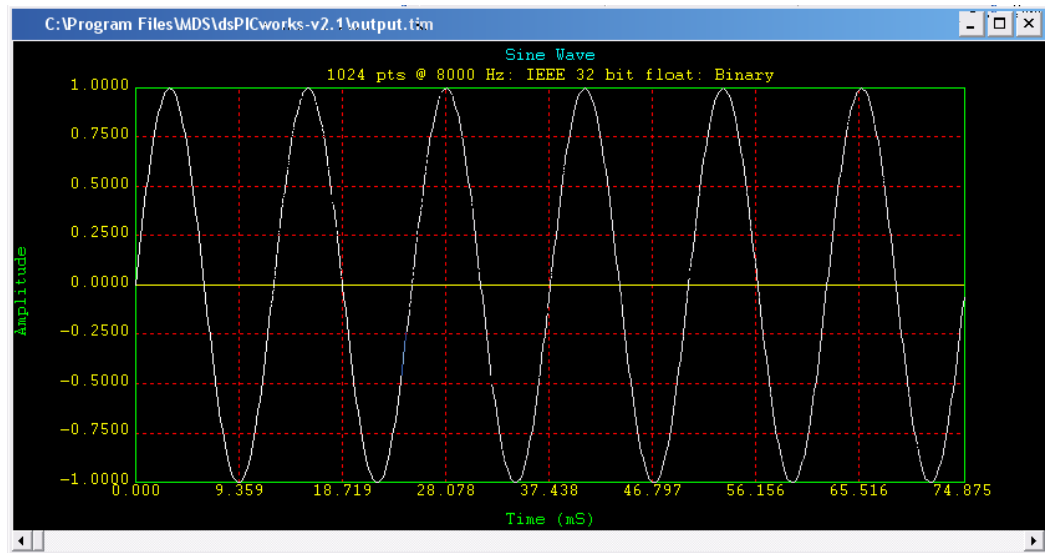


Рисунок 4.2

Перед подачей на микроконтроллер с сигналами можно провести ряд различных операций, таких как суммирование сигналов разных типов, нахождение корреляционных функции, использование преобразований Фурье, интерполяция, свертка и т.д.

5. Синтез сигналов в dsPICWorks

1) Сформируем автокорреляционную функцию синусоидального сигнала, для чего выбираем пункт меню «DSP->Autocorrelation» Вводим необходимые параметры:

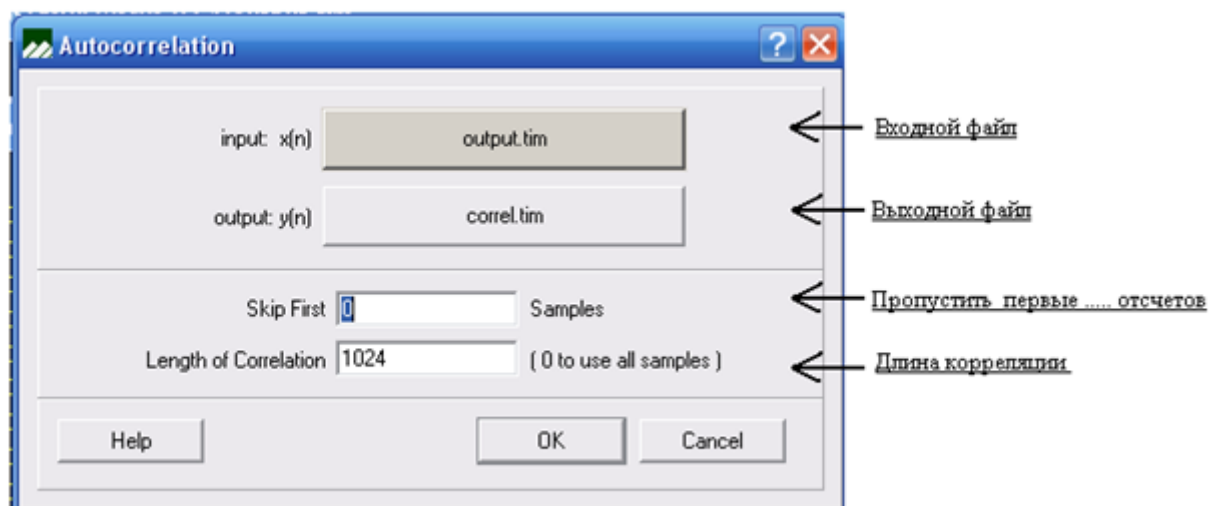


Рисунок 5.1.1

Получаем ее изображение:

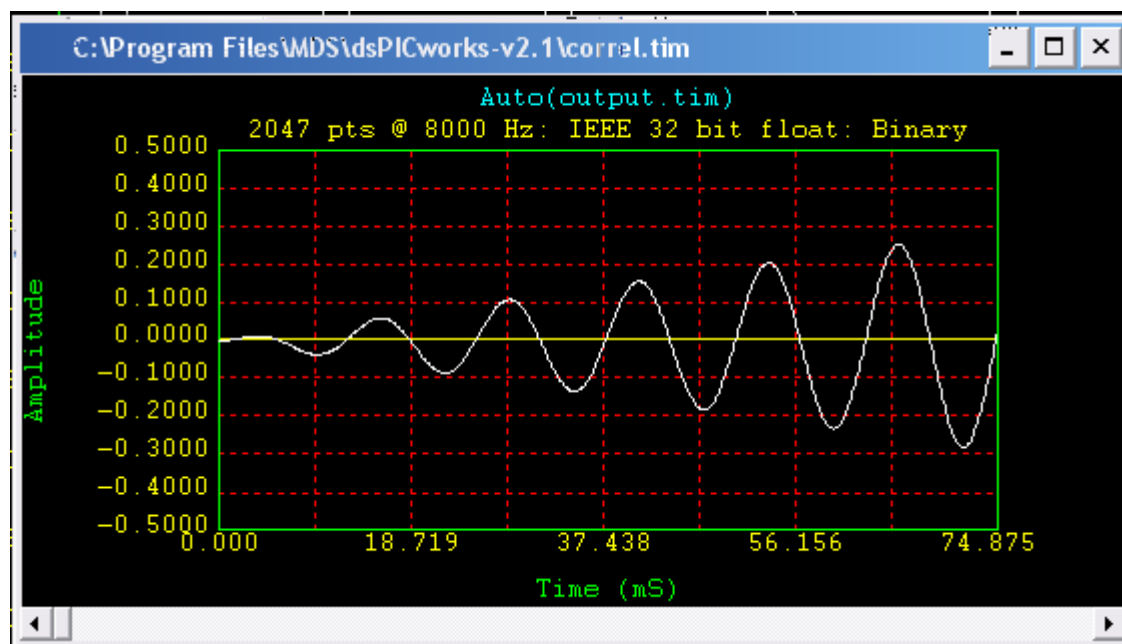


Рисунок 5.1.2

2) Смешаем сигнал с шумом. Для чего открываем «Generator - >NoiseFunction»

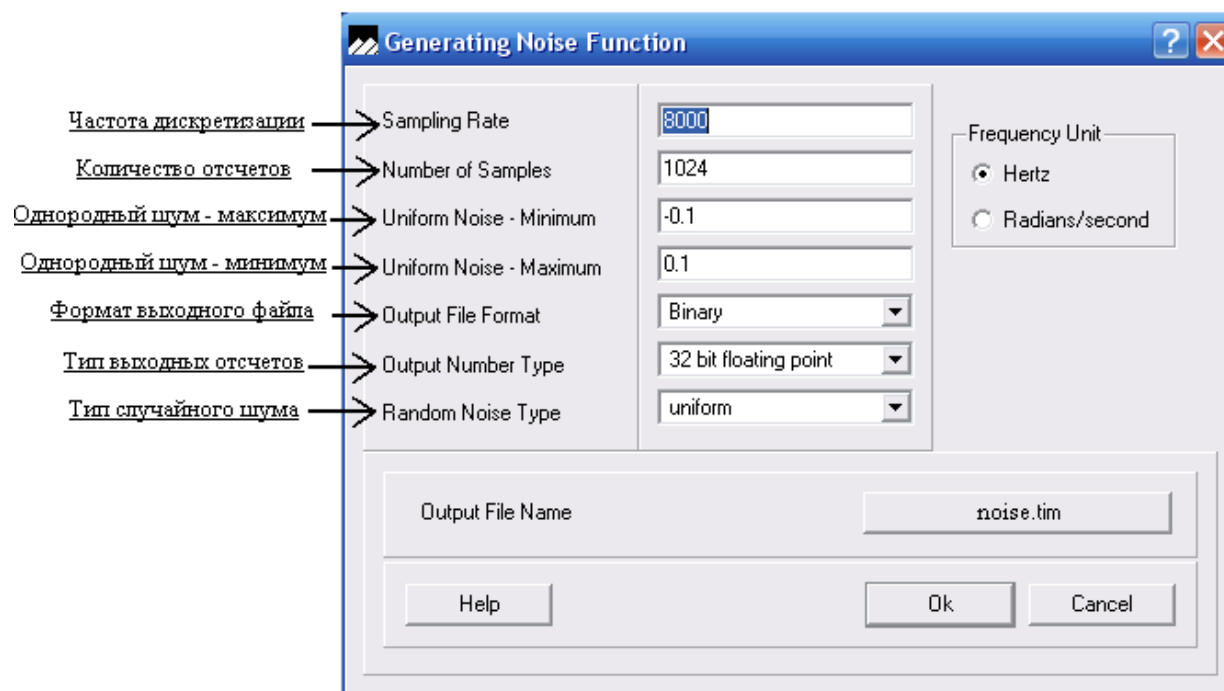


Рисунок 5.2.1

Получаем изображение:

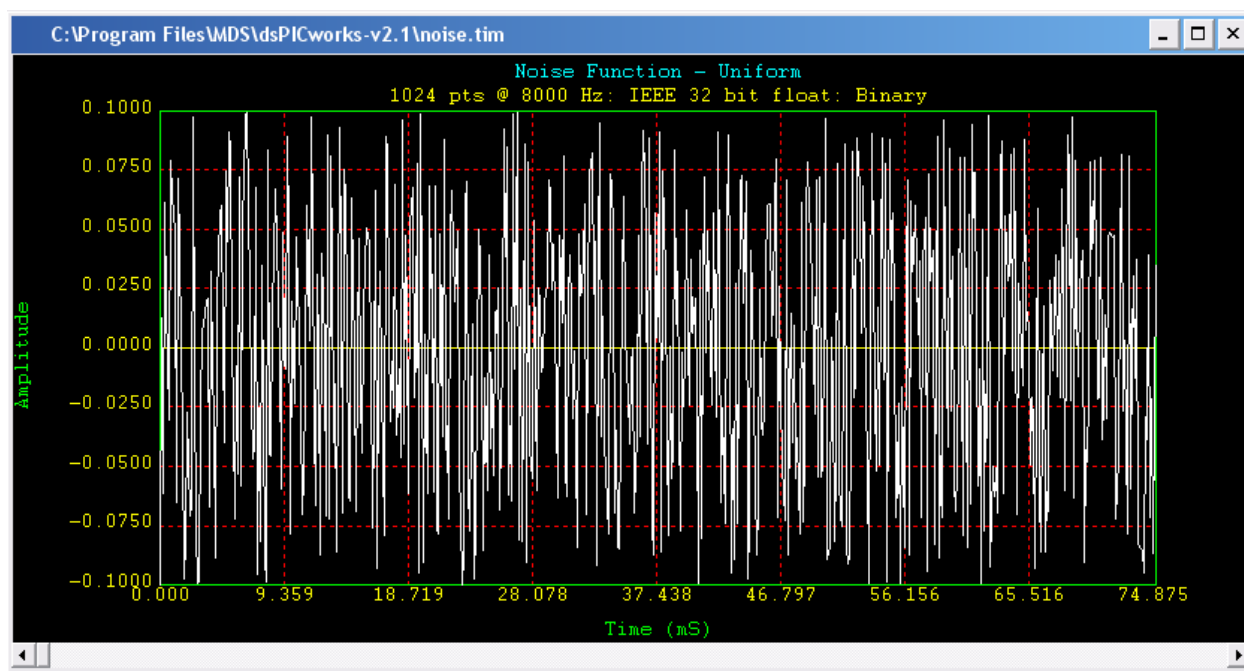


рисунок 5.2.2

б) Для получения смеси шума и синусоидального сигнала выбираем «Operation->Arithmetic->LinearCombo...»

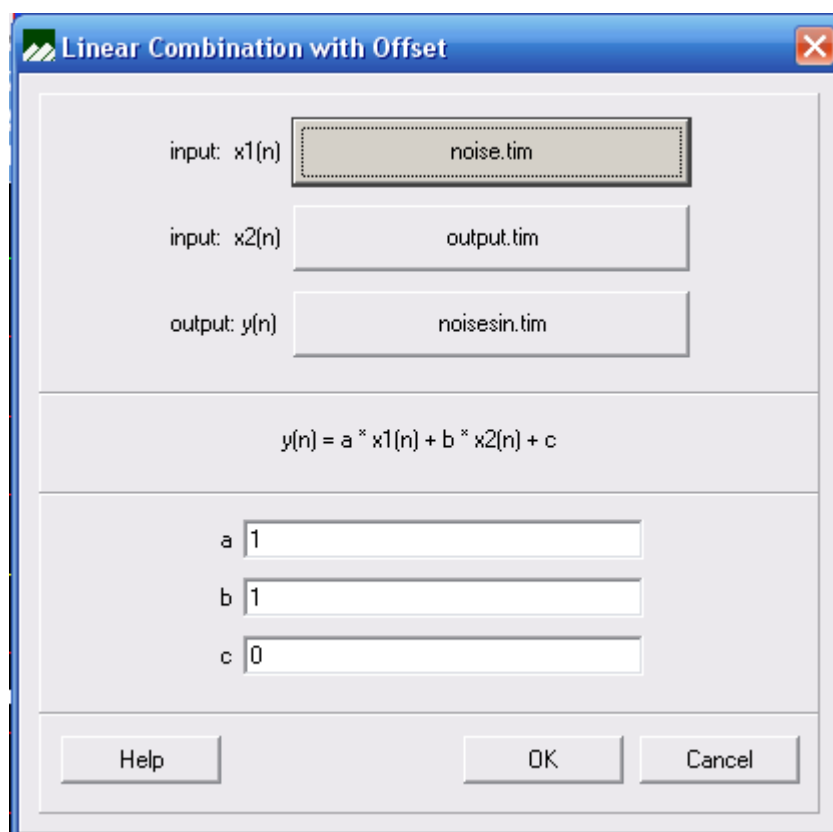


Рисунок 5.2.3

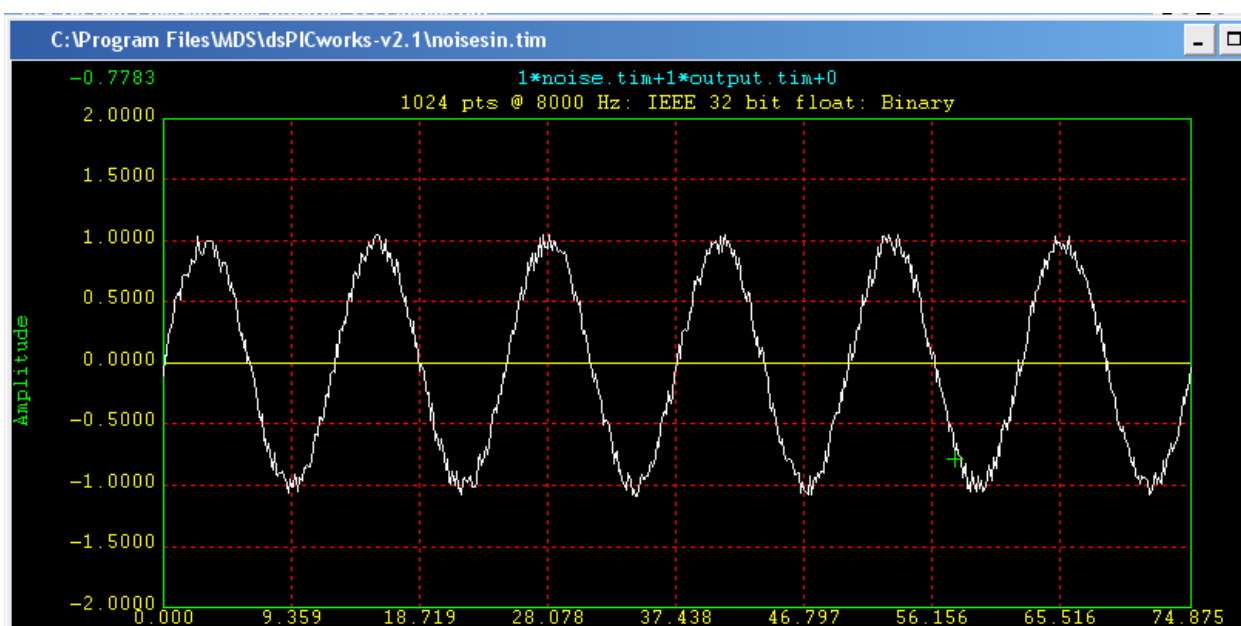


Рисунок 5.2.4

Найдем спектр полученного сигнала с помощью быстрого преобразования Фурье «DSP ->FastFourierTransform»

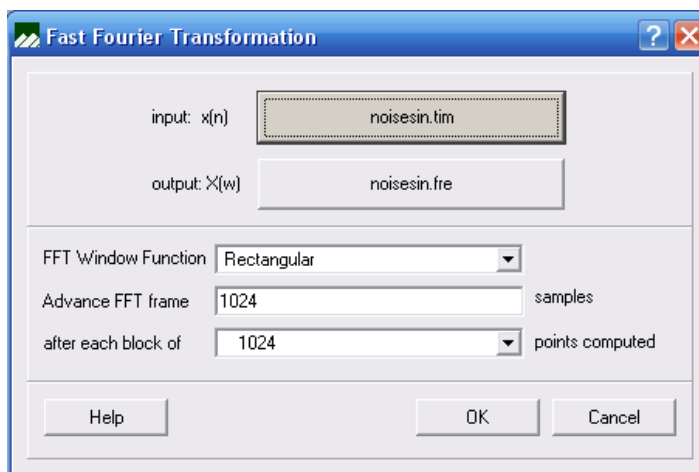


Рисунок 5.2.5

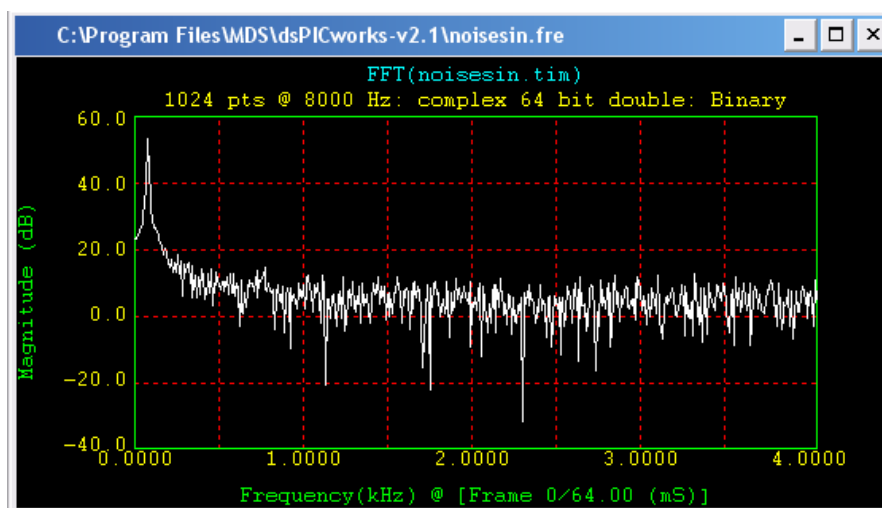


Рисунок 5.2.6

3) В завершение очистим сигнал от шумовой составляющей. Готовых фильтров в программе нет, да и гораздо удобнее создать свой фильтр с конкретными параметрами. Для этого воспользуемся программным приложением dsPICFDLite. Не будем рассматривать это программное приложение столь подробно как dsPICWorks, но основные положения для ускорения процесса освоения данного продукта предложим ниже. Для этого рассмотрим вкладки и их назначения.

File	Файл
Load Specifications ...	Загрузить спецификацию
Print ...	Печать ...
System Settings ...	Системные настройки
User Information ...	Пользовательская информация
Exit ...	Выход
About dsPIC FD Lite ...	O dsPIC FD Lite ...

Рисунок 5.3.1

View	Вид
<input checked="" type="checkbox"/> Toolbar	<input checked="" type="checkbox"/> Инструменты
<input checked="" type="checkbox"/> Status Bar	<input checked="" type="checkbox"/> Строка состояния
View Filter File ...	Просмотреть файл

Рисунок 5.3.2

Design	Разработка
<input checked="" type="checkbox"/> FIR Design ...	<input checked="" type="checkbox"/> БИХ проект
FIR Window Design ...	КИХ проект
Equiripple Design ...	Эллиптический проект

Рисунок 5.3.3

Filter	Фильтр
<input checked="" type="checkbox"/> Lowpass ...	<input checked="" type="checkbox"/> НЧ фильтр
Highpass ...	ВЧ фильтр
Bandpass ...	Полосовой фильтр
Bandstop ...	Режекторный ...
Start Design ...	Старт проекта

Рисунок 5.3.4

Output	Выход
Plot Control ...	Участок контроля
Snap to Grid	Привязка к сетке
Quantization ...	Квантизация
Create Specification File ...	Создать спецификационный файл
Create Filter Coefficient File ...	Создать файл фильтра коэф ...
Create Matlab File ...	Создать файл Matlab
Create Plot Data Files ...	Создать файл участка данных

Рисунок 5.3.5

Codegen	Codegen
C Code ...	Код ...
Microchip ▶ dsPIC30	Microchip ▶ dsPIC30

Рисунок 5.3.6

Window	Окно
Cascade	Каскад
Tile	Tile
Select Plots ...	Выбрать участок
Display Control ...	Отобразить контроль
Magnitude	Величина
Log Magnitude(dB)	Log величина (dB)
Phase	Фаза
Group Delay	Групповая задержка
Impulse Response	Импульсная хар-ка
Step Response	Переходная хар-ка
✓ Pole/Zero Plot	✓ Полюсы/нули

Рисунок 5.3.7

На панели задач имеются соответствующие пиктограммы:

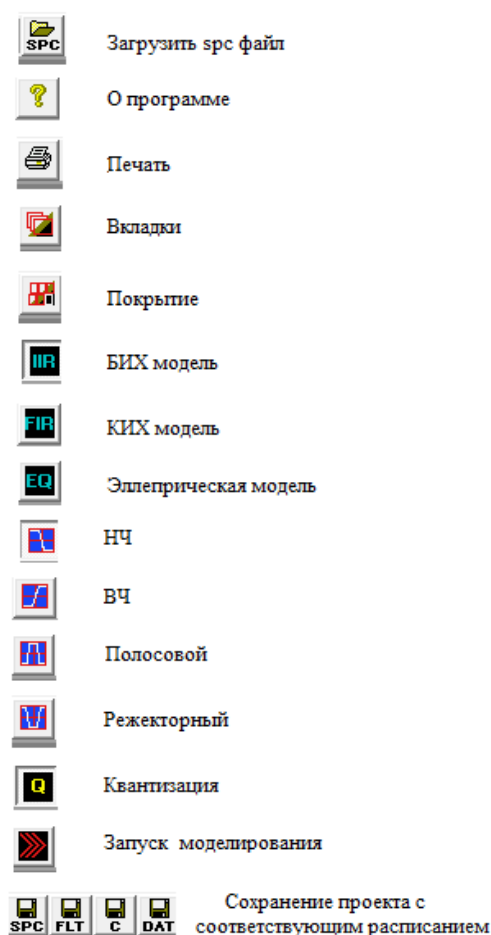


Рисунок 5.3.8

ВовкладкеFilter ->LowpassFilter

Какой фильтр?

На примере спроектируем НЧ**БИХ**фильтр, который работает при частоте дискретизации 44,1 кГц. Фильтр определен согласно рис. 6.22. Граничная частота полосы пропускания составляет 18 кГц. Полоса задержки начинается при 21 кГц, неравномерность полосы пропускания равна 0,01 дБ, а неравномерность полосы задержки (ослабление) — 96 дБ. Мы также должны определить длину слова (разрядность) коэффициентов, которая в данном случае составляет 16 разрядов, принимая во внимание, что используется 16-разрядный **процессор DSP** с фиксированной точкой

Задаем параметры для фильтра согласно рис. 5.3.9

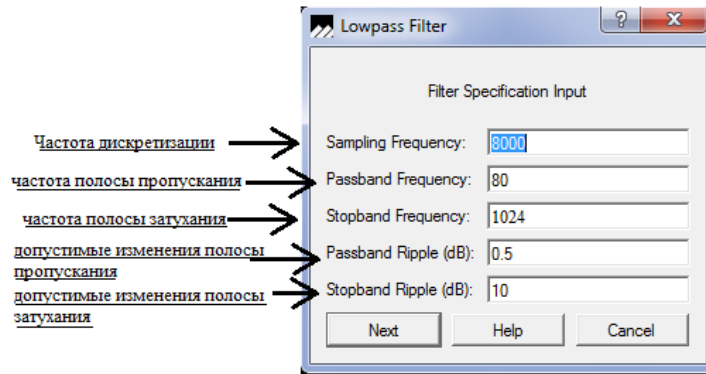


Рисунок 5.3.9

Где допустимые изменения полосы пропускания и полосы заграждения это величины, изображенные на рисунке Рисунок 5.3.10

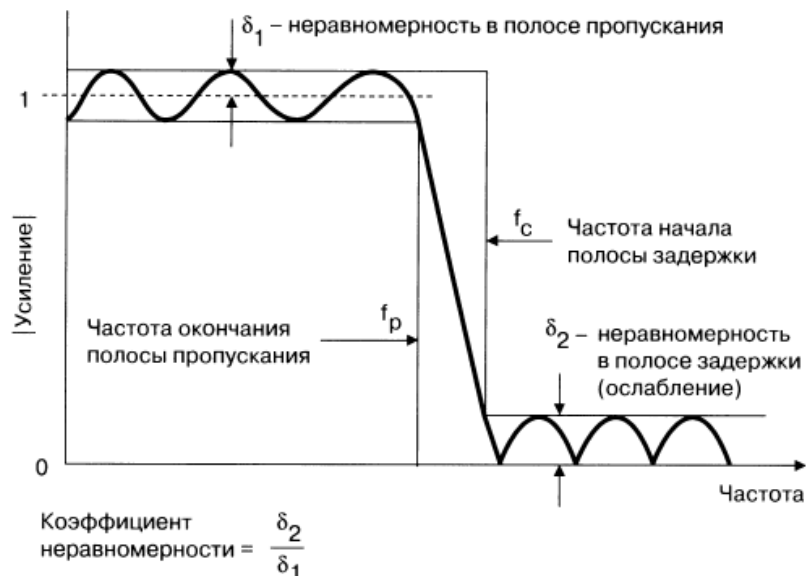


Рисунок 5.3.10

БИХ-фильтры которые имеют аналоговые эквиваленты (фильтры Баттерворта, Чебышева, эллиптический и Бесселя). Выберем, эллиптический(Рисунок 5.3.11).

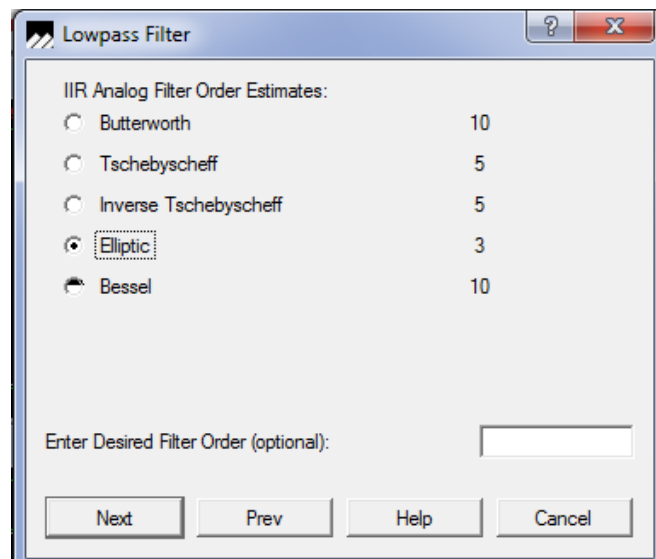


Рисунок 5.3.11

При этом характеристики формируемого фильтра отображаются на графиках различного вида:

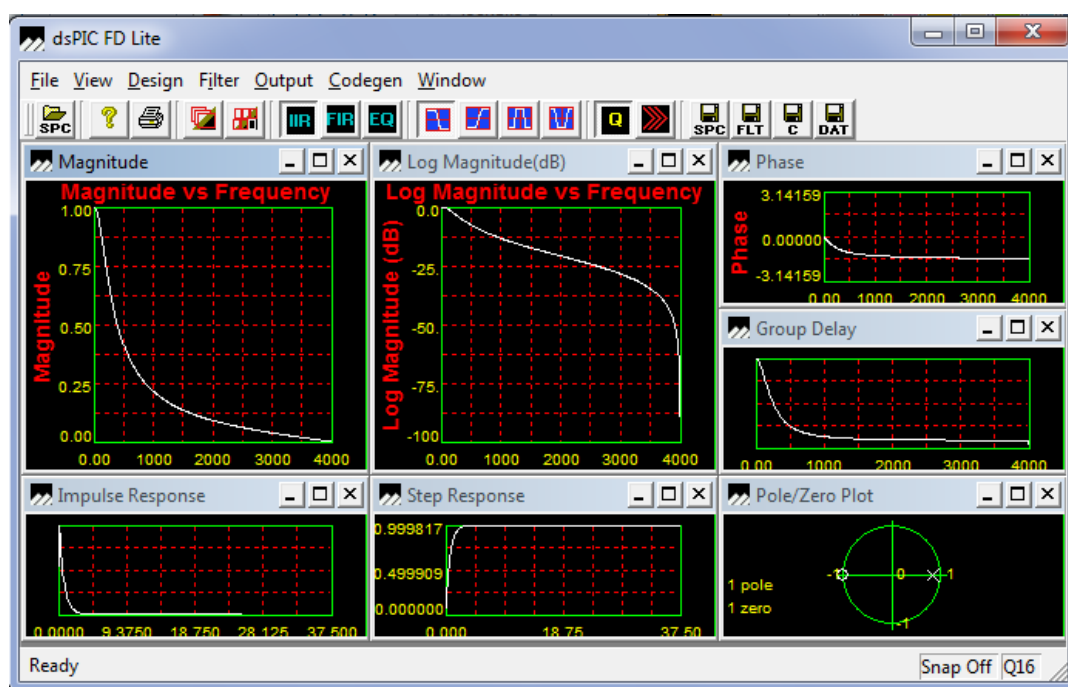



Рисунок 5.3.12

Сохраняем полученный фильтр в формате (*.flt), нажав соответствующую иконку на панели задач .

Возвращаемся в dsPICWorks. Вкладка DSP ->SignalFiltering

Применим полученный фильтр к сигналу

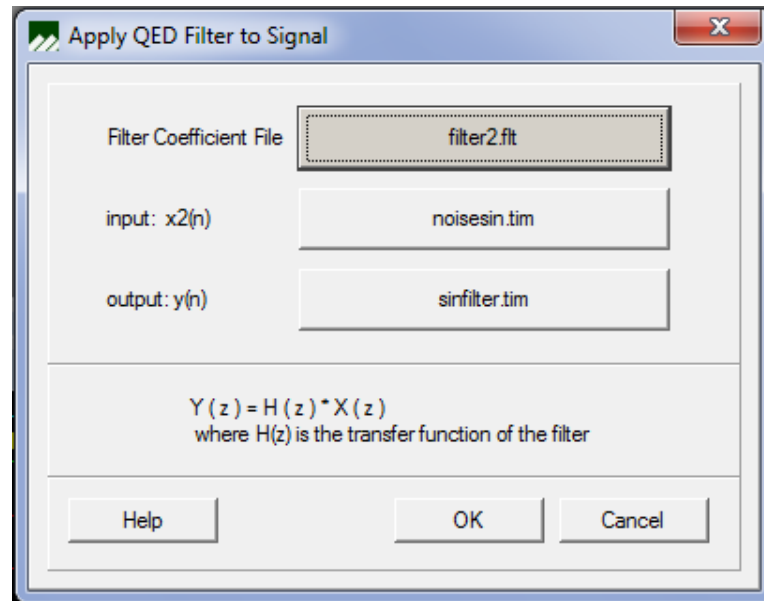


Рисунок 5.3.13

Получаем сигнал, по полюсу идентичный первоначально заданному.

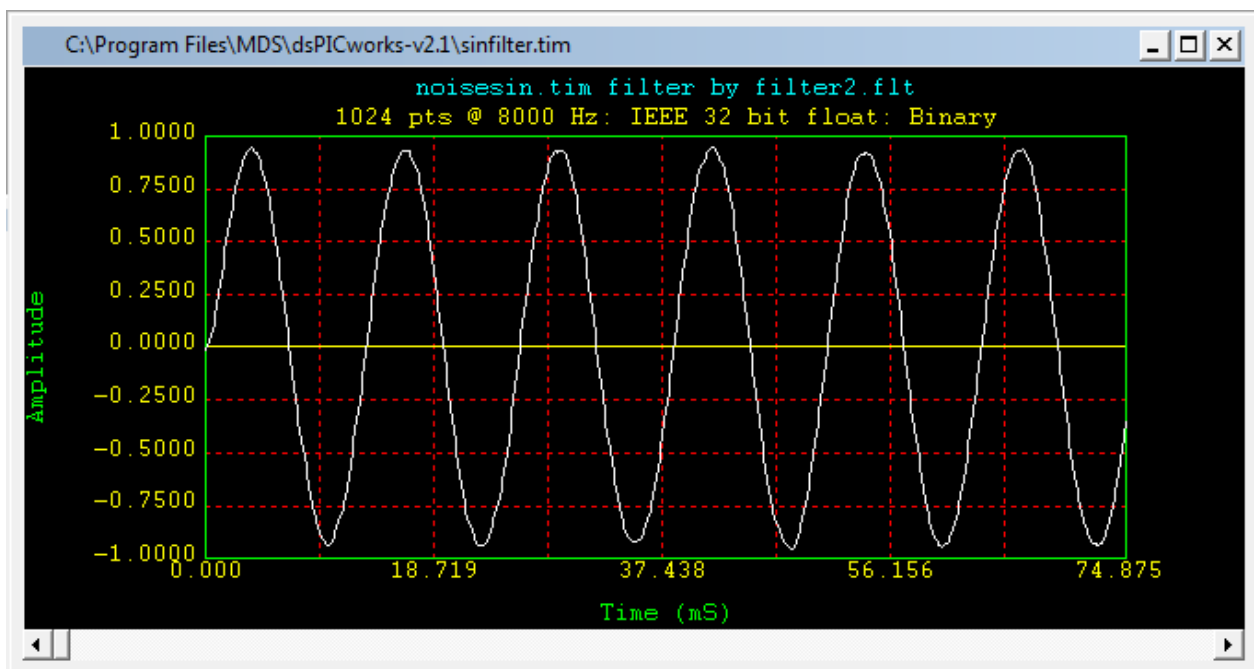


Рисунок 5.3.14

Важной особенностью программы является то, что все произведенные действия записываются в виде кода, для примера приведем кусок данного кода для нашего фильтра. Выбираем Codegen -> Ccode и

сохраняем файл в необходимой папке. При открытии данного файла получаем:

```

; .....
; File 2.s
; .....

.equ 2NumTaps, 64

; .....
; Allocate and initialize filter taps

.section .xdata
.align 128

2Taps:
.hword 0xFF7D, 0x001B, 0x0059, 0xFF37,
0x0127, 0xFE9E, 0x0174, 0xFEAD, 0x0102
.hword 0xFF7A, 0xFFEE, 0x00B9, 0xFEAA,
0x01D7, 0xFDDD, 0x022F, 0xFE14, 0x015E
.hword 0xFF79, 0xFF7E, 0x01AA, 0xFD35,
0x03C4, 0xFB98, 0x0494, 0xFBE6, 0x02D5
.hword 0xFF78, 0xFCD7, 0x093D, 0xEA99,
0x49DD, 0x49DD, 0xEA99, 0x093D, 0xFCD7
.hword 0xFF78, 0x02D5, 0xFBE6, 0x0494,
0xFB98, 0x03C4, 0xFD35, 0x01AA, 0xFF7E
.hword 0xFF79, 0x015E, 0xFE14, 0x022F,
0xFDDD, 0x01D7, 0xFEAA, 0x00B9, 0xFFEE
.hword 0xFF7A, 0x0102, 0xFEAD, 0x0174,
0xFE9E, 0x0127, 0xFF37, 0x0059, 0x001B
.hword 0xFF7D

; .....
; Allocate delay line in (uninitialized) Y data space

.section .ybss, "b"
.align 128

2Delay:

```

```

        .space 2NumTaps*2

; .....
; Allocate and initialize filter structure

        .section .data
        .global _2Filter

_2Filter:
        .hword 2NumTaps
        .hword 2Taps
        .hword 2Taps+2NumTaps*2-1
        .hword 0xff00
        .hword 2Delay
        .hword 2Delay+2NumTaps*2-1
        .hword 2Delay

; .....
; .....
; Sample assembly language calling program
; The following declarations can be cut and pasted as needed into a program
;
;         .extern _FIRFilterInit
;         .extern _BlockFIRFilter
;         .extern _2Filter
;
;         .section .bss
;
;         The input and output buffers can be made any desired size
;         the value 40 is just an example - however, one must ensure
;         that the output buffer is at least as long as the number of samples
;         to be filtered (parameter 4)
;input:         .space 40
;output:        .space 40
;         .text
;
;
; This code can be copied and pasted as needed into a program
;

```

```

;
; Set up pointers to access input samples, filter taps, delay line and
; output samples.
;          mov    #_2Filter, W0  ; Initialize W0 to filter structure
;          call   _FIRFilterInit ; call this function once
;
; The next 4 instructions are required prior to each subroutine call
; to _BlockFIRFilter
;          mov    #_2Filter, W0  ; Initialize W0 to filter structure
;          mov    #input, W1     ; Initialize W1 to input buffer
;          mov    #output, W2    ; Initialize W2 to output buffer
;          mov    #20, W3        ; Initialize W3 with number of required output
samples
;          call   _BlockFIRFilter ; call as many times as needed

```