**Описание программного обеспечения**

**«Генератор сигналов»**

**Автор: ведущий инженер отдела РЭУ Свиридов С. А.**

**ОАО НПП «АМЭ»**

**2017**

Оглавление

[Системные требования. 3](#_Toc473541471)

[Установка и запуск 3](#_Toc473541472)

[Генератор 4](#_Toc473541473)

[Описание полей формы 4](#_Toc473541474)

[Типы сигналов. 6](#_Toc473541475)

[Наложение окна 12](#_Toc473541476)

[Тип окна 12](#_Toc473541477)

[Форма окна 13](#_Toc473541478)

[Место наложения 13](#_Toc473541479)

[Длительность окна 13](#_Toc473541480)

[Примеры наложения окон. 13](#_Toc473541481)

[Преобразование ШИМ 16](#_Toc473541482)

[Редактор формы спектра 16](#_Toc473541483)

[Фильтрация 19](#_Toc473541484)

[Выгрузка 19](#_Toc473541485)

[Выходные файлы 20](#_Toc473541486)

[Отрисовка 20](#_Toc473541487)

[Общее 20](#_Toc473541488)

[Описание кнопок 22](#_Toc473541489)

[Логирование 22](#_Toc473541490)

# Системные требования.

Операционная система: Windows, Linux

Рекомендуемые аппаратные требования: процессор с частотой не ниже 1 ГГц, ОЗУ не менее 1 Гб, сетевая карта.

Программные компоненты: Python версии не ниже 3.5, NumPy+MKL, SciPy, Matplotlib.

# Установка и запуск

Программа написана на языке программирования Python версии 3.5. Для работы программы тербуется установка Python с версией не ниже 3.5, а также устновка дополнительных компонентов: NumPy+MKL, SciPy и Matplotlib.

Порядок установки:

1. Устанавливаем Python 3.

Для Linux:

> sudo apt-get install python3

Для Windows скачиваем и устанавливаем версию не ниже 3.5:

<https://www.python.org/downloads/>

1. Устанавливаем дополнительные компоненты.

Для Linux:

> sudo apt-get install python3-numpy python3-scipy python3-matplotlib

Для Windows (установка через pip):

> python -m pip install -U pip

> pip install numpy scipy matplotlib

Внимание! pip автоматически устанавливается с Python начиная с версии 3.4. Если же pip по каким то причинам отсутствует, те необходимо установить его как описано здесь:

<https://pip.pypa.io/en/stable/installing/#do-i-need-to-install-pip>

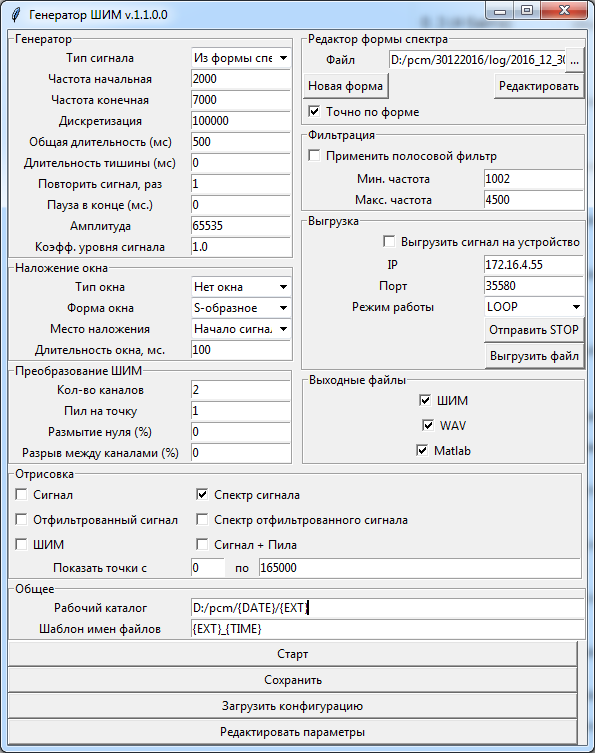
Запуск. Программа представляет сосбой набор скриптов с расширением py. Для запуска пограммы необходимо из командной строки запустить файл main.py.

Linux:

> python3 путь\_к\_файлу/main.py

Windows:

> python путь\_к\_файлу/main.py



Данное описание программы приведено для версии 1.1.0.х.

# Генератор

## Описание полей формы

Тип сигнала позволяет выбрать [тип генерируемого сигнала](#_Типы_сигналов.).

Частота начальная используется для генерации периодических сигналов типа синусоиды.

Частота конечная используется для генерации сигналов имеющих диапазон частот (например ЛЧМ или сигнал из спектра), а также для пакетов сигналов.

Дискретизация – количество отсчетов на 1 секунду в генерируемом сигнале.

Общая длительность – параметр, определяющий длину в миллисекундах одного сигнала в пачке. Общая длительность это длительность самого сигнала плюс тишина, которая необходима для отделения одного сигнала от другого. Используется при генерации пакетов сигналов или если необходимо повторить один и тот же сигнал несколько раз.

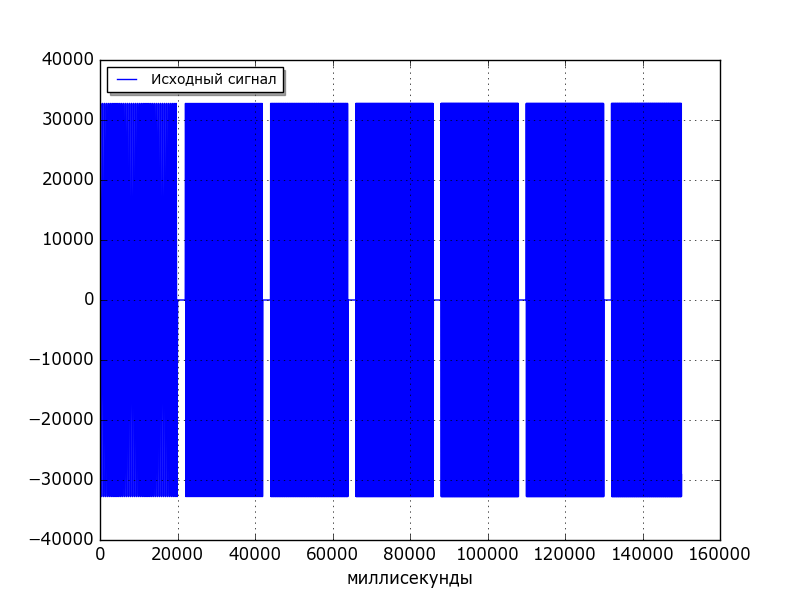
Длительность тишины – определяет длительность тишины в миллисекундах после каждого сигнала в пачке.

Например, если нам необходим сигнал длительностью 200 мс. и после него тишина 20 мс. то необходимо назначить параметрам следующие значения:

Общая длительность = 220

Длительность тишины = 20

Пример пакетного сигнала с данными параметрами:



Повторить сигнал раз – параметр, который позволяет повторить один и тот же сигнал заданное число раз. То есть сигнал генерируется с заданными параметрами ОДИН раз, а затем копируется заданное количество раз. Это позволяет излучить в точности один и тот же сигнал несколько раз.

Пауза в конце – параметр, который определяет длительность паузы после всех повторов сигнала в миллисекундах. Позволяет вставлять разрыв между пакетами сигналов, для удобства анализа.

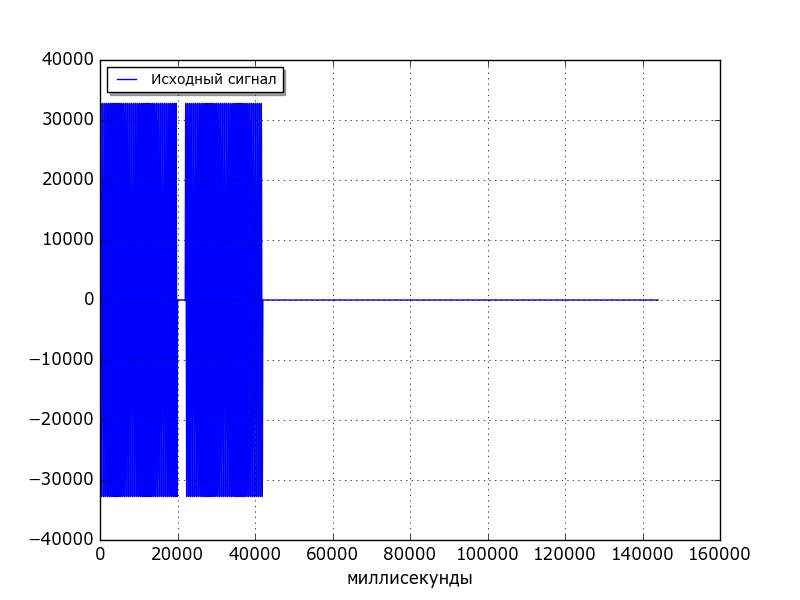
Например, при следующих параметрах

Общая длительность = 220

Длительность тишины = 20

Повторить сигнал раз = 2

Пауза в конце = 1000, получим примерно следующий сигнал:



Общая длительность сигнала вычисляется как:

Общая длительность × Количество повторов + Пауза в конце

Амплитуда – задает максимальные и минимальные значения генерируемого сигнала. Этот параметр носит условный характер. Влияет на точность вычисления уровня каждого отсчета сигнала. Максимальное значение 65535 (0хFFFF).

Коэфф. уровня сигнала – параметр, с помощью которого можно регулировать уровни отсчетов сигнала (изменять громкость). Например, при значении параметра равном 0,5 все уровни сигнала будут уменьшены на 50%, то есть понизится громкость. При значении 100 все уровни будут увеличены в 100 раз. Как показывает практика, сигнал со средним уровнем ниже 1 000 не слышен и его необходимо поднимать.

## Типы сигналов.

#### Шум. Амплитуда каждого отсчета определяется с помощью генератора случайных чисел с нормальным законом распределения.

#### Синус. Тоновый сигнал с частотой, задаваемой параметром «Частота начальная».

#### Меандр (Импульсный сигнал)

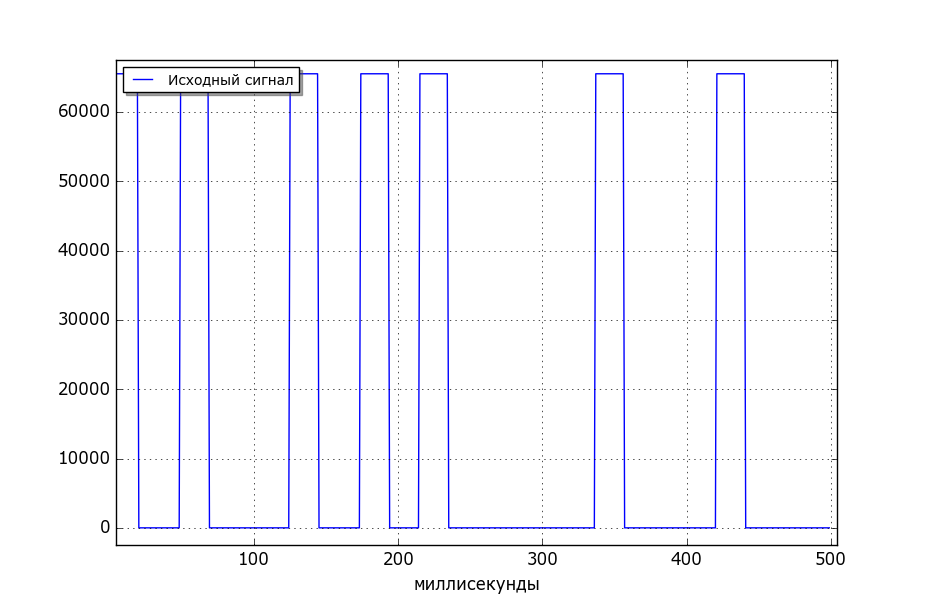
Значения сигнала имеют два значения: либо 0, либо ±Амплитуда.

Параметры меандра не вынесены на форму программы и определяется параметрами файла main.config. Если необходимо изменить параметры меандра, необходимо открыть файл main.config, отредактировать нужный параметр и сохранить файл. После этого сгенерировать сигнал заново с новыми параметрами. Программу перезапускать не нужно!

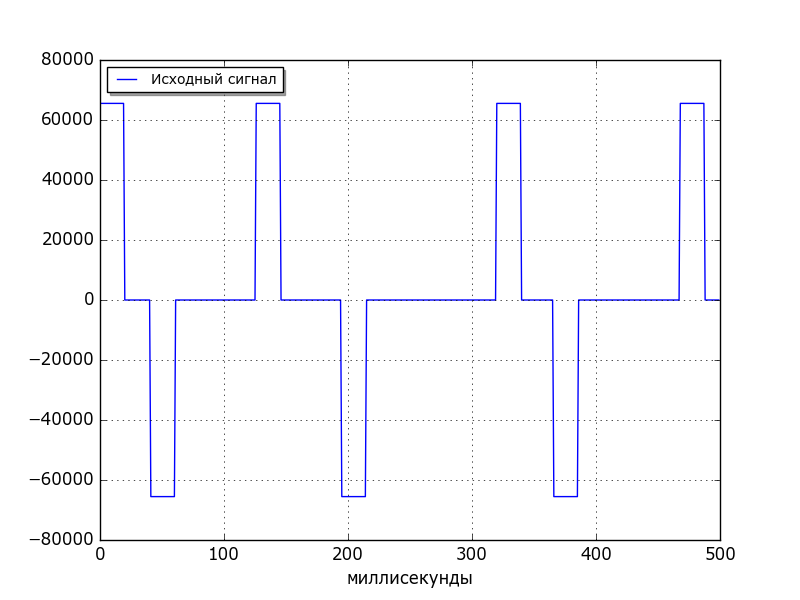
Внимание! Все временные параметры меандра измеряются в микросекундах.

**meandr\_type** – тип меандра (0, 1, 2), где:

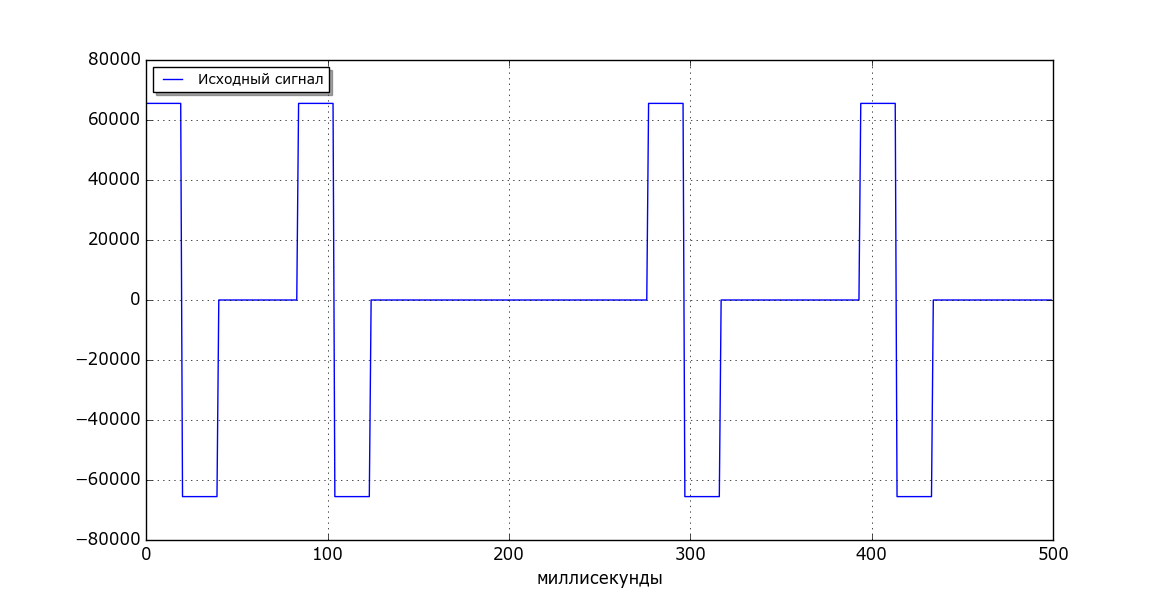
0 – однополярный, все значения сигнала находятся в положительной области. Пример:



1 – двуполярный чередующийся, положительные и отрицательные значения сигнала чередуются, причем между ними находится интервал. Пример:



2 – двуполярый би-импульс. Положительный и отрицатеьлный импульсы следуют друг за другом, а затем идет интервал. Пример:



**meandr\_pulse\_width** – ширина импульса, в микросекундах;

**meandr\_interval\_width** и **meandr\_random\_interval** – определяют ширину интервалов в микросекундах. Ширина интервала может быть постоянная или случайная. При meandr\_random\_interval = 0, ширина интервала постоянна, и равна значению meandr\_interval\_width.

При meandr\_random\_interval НЕ равном 0, meandr\_interval\_width определяет нижнюю границу ширины интервала, а meandr\_random\_interval – верхнюю границу ширины интервала. То есть, при формировании сигнала, ширина каждого интервала будет выбираться случайным образом в интервале значений [meandr\_interval\_width, meandr\_random\_interval]. В данный момент, используется треугольный закон распределения, со смещением вершины в сторону меньшего значения. В результате увеличивается суммарный вес высоких частот в спектральной области.

Примерное распределение длин интервалов при:

meandr\_interval\_width = 200 мкс.

meandr\_random\_interval = 2000 мкс.

#### Пакет синусов

Формируется пакет синусоид с повышающейся частотой. Длительность каждой синусоиды определяется параметром «Общая длительность». Частота начальной синусоиды определяется параметром «Частота начальная». Частота последующих синусоид определяется параметром **sinus\_pack\_step** в файле main.config. Параметр задает приращение частоты в герцах. Синусоиды формируются до тех пор не будет достигнута «Частота конечная». Например при следующих параметрах:

Частота начальная = 100

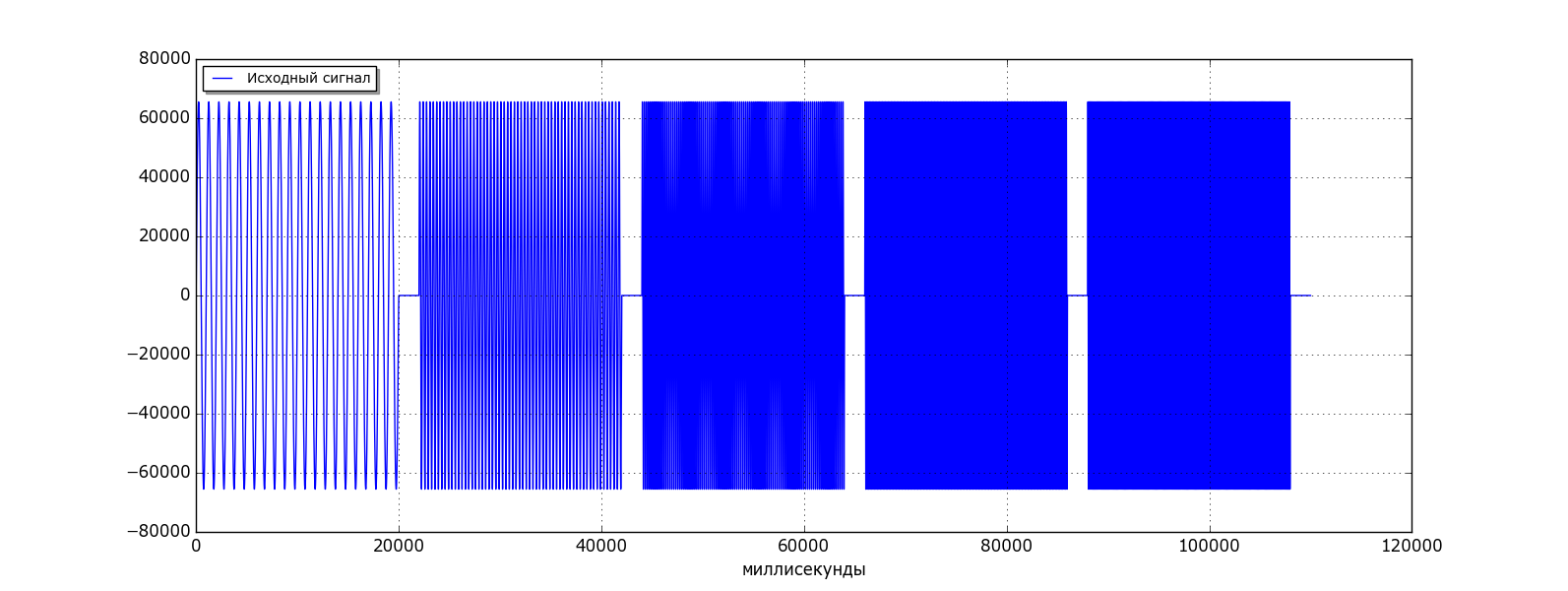
Частота конечная = 1000

sinus\_pack\_step = 200

Общая длительность = 220

Длительность тишины = 20

получим набор синусоид с частотами 100, 300, 500, 700 и 900 Гц.



#### Синус + Синус + Шум

Одна синусоида со сдвигом по фазе и амплитуде наложенная на другую и с наложенным шумом. Сигнал для тестирования и отладки программы.

#### ЛЧМ

Сигнал с линейно-частотной модуляцией. Начальная частота определяется параметром «Частота начальная». Далее частота сигнала увеличивается с каждым отсчетом, таким образом, что за время Общая длительность – Длительность тишины, достигает конечной частоты. Формула для вычисления значения уровня каждого отсчета имеет вид:

, где:

A – амплитуда

*f0*– частота начальная

*f1*– частота конечная

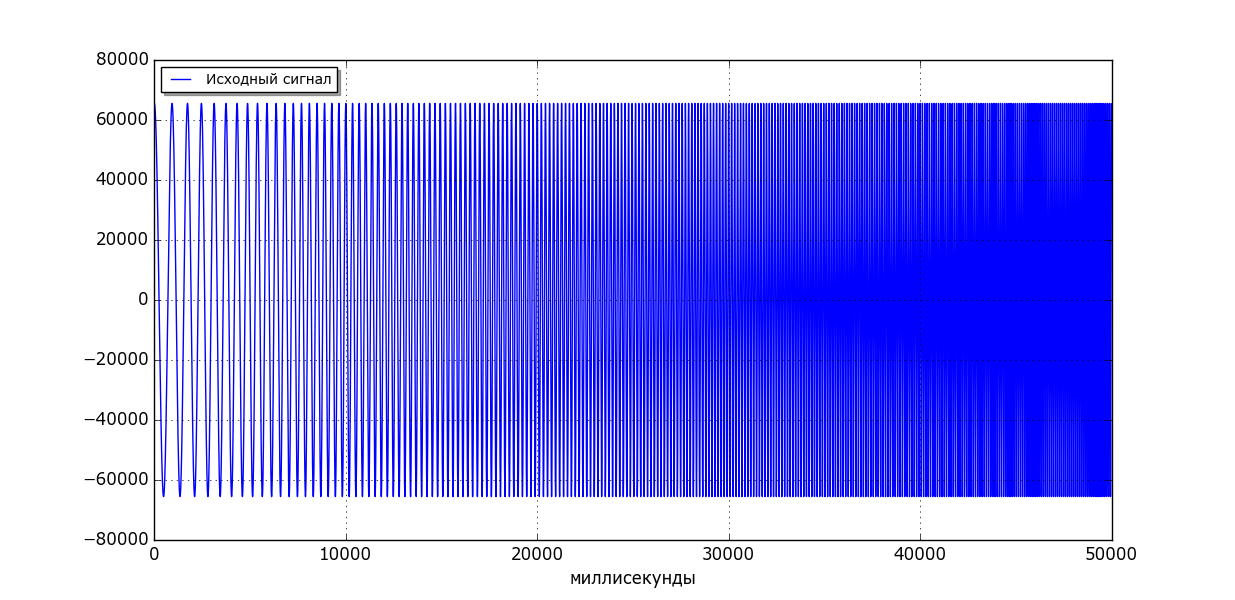
*fd*– частота дискретизации

*d* - шаг приращения

*d* = 1 / (*fd* \* *T*), где *T* – время чистого сигнала без тишины в секундах

*T* = (Общая длительность – Длительность тишины) / 1000

n – номер отсчета



#### Пакет меандров

Формируется пакет меандров с повышающейся длиной интервала. Длительность каждого меандра определяется параметром «[Общая длительность](#_Описание_полей_формы)». Начальная длительность интервала определяется параметром **meandr\_interval\_width** в файле main.config. Длительность последующих интервалов определяется параметром **meandr\_pack\_step** в файле main.config. Параметр определяет шаг приращения длительности в микросекундах. Меандры формируются до тех пор пока не будет достигнуто значение параметра **meandr\_random\_interval**. Например при следующих параметрах:

meandr\_interval\_width = 200

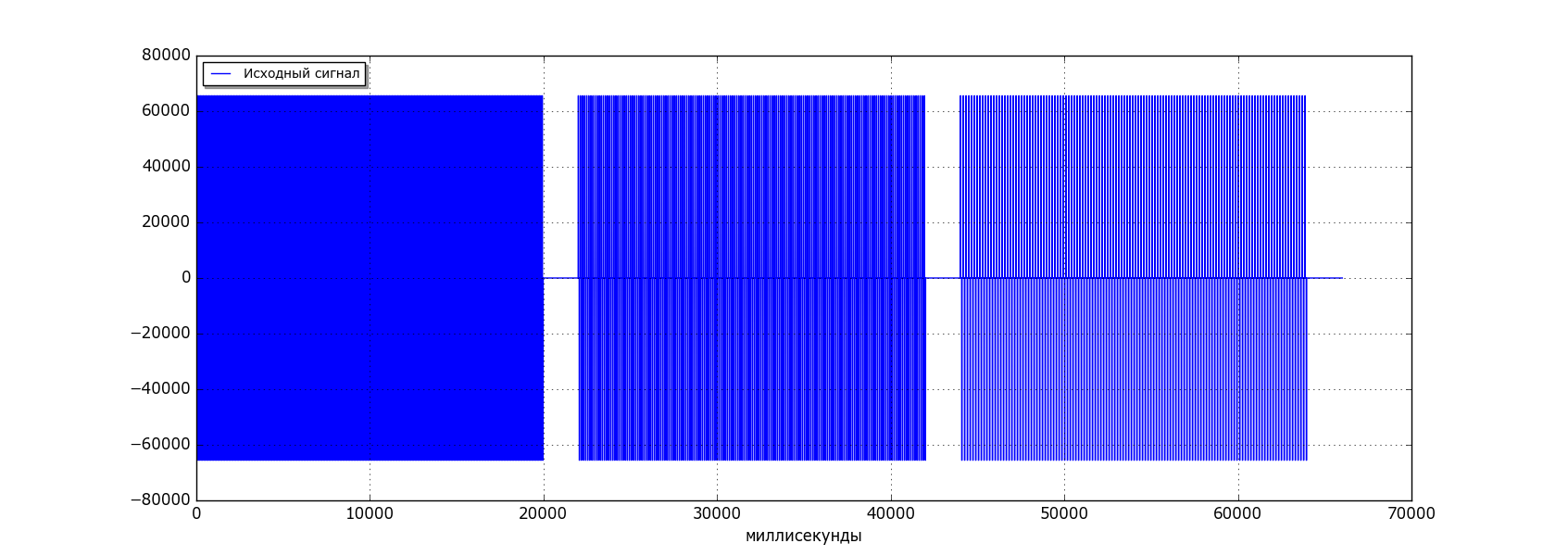
meandr\_random\_interval = 700

meandr\_pack\_step = 200

Общая длительность = 220

Длительность тишины = 20

получим набор меандров с длительностями интервалов 200, 400 и 600 микросекунд.



#### Из спектра

Не для практического применения. Данный сигнал был добавлен для отладки алгоритмов работы со спектрами.

Для формирования брался ранее полученный сигнал, с определенной АЧХ. Через быстрое преобразование Фурье получаем спектр данного сигнала. Далее по форме этого спектра формируется новый сигнал.

Данный способ не удобен, так как не позволяет изменять форму спектра и требует дополнительных параметров, поэтому для формирования сигналов с заданной формой добавлены два сигнала: «[Ровный спектр](#_Ровный_спектр)» и «[Из формы спектра](#_Из_формы_спектра)».

#### Ровный спектр

Алгоритм формирования сигнала следующий:

1. сначала формируется обычный [шумовой сигнал](#_Шум.). Данный сигнал имеет равномерный спектр.
2. Производим БПФ этого сигнала и получаем массив комплексных чисел, описывающих спектр данного сигнала.
3. В диапазоне частот, которые заданы нижней и верхней границами, проходим по этому массиву и выбираем значение фазы. Вычисляем новое комплексное число по формуле:

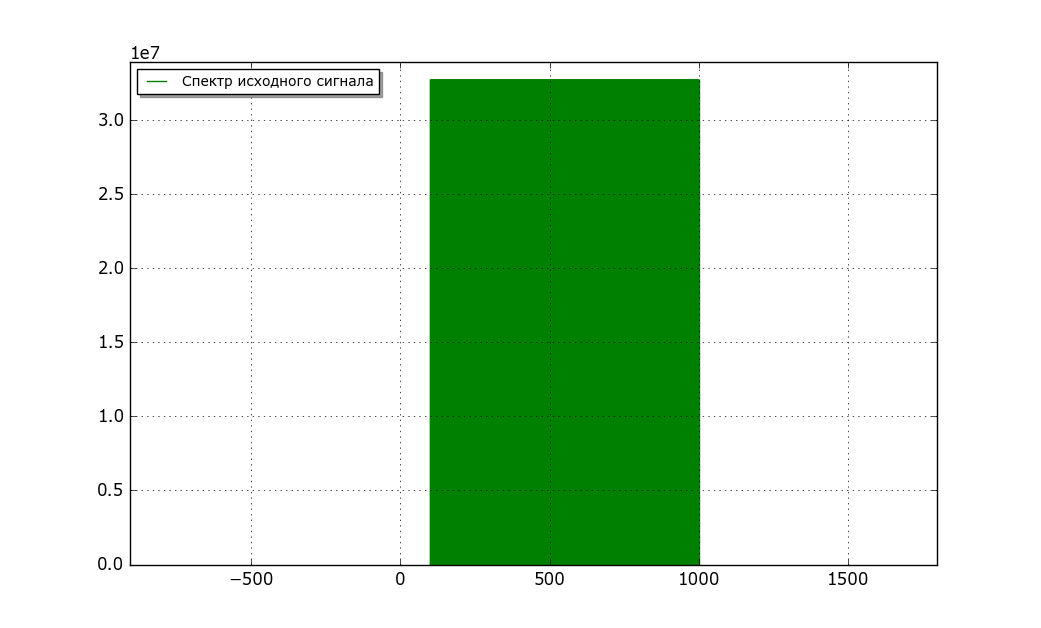
Im = A ∙ cos(ϕ) – мнимая часть

Re = A ∙ sin(ϕ) – реальная часть, где

A – значение [амплитуды](#_Описание_полей_формы), умноженное на [коэффициент уровня сигнала](#_Описание_полей_формы)

ϕ – значение фазы, полученное из шумового сигнала

1. После этого производим обратное преобразование Фурье и получаем шумовой сигнал с ровной АЧХ. Пример спектра сигнала полученного таким образом:



Данный сигнал может применяться для изучения искажения сигнала в среде при передаче.

#### Из формы спектра

Для формирования сигнала с заданной АЧХ можно вручную нарисовать необходимую форму спектра, по которой потом будет сформирован сигнал. В разделе «Редактор форм спектра» описана работа с файлами форм спектров.

После редактирования формы спектра, сигнал формируется также как описано для сигнала «Ровный спектр», за исключением вычисления амплитуды нового сигнала. Амплитуда вычисляется следующим образом.

1. Из файла формы спектра считываются значения регуляторов. Каждый регулятор имеет два параметра – частота, к которой он привязан и уровень в процентах.
2. Вычисляются уровни точек, лежащих на прямой между двумя соседними регуляторами.
3. Значение уровня каждой точки формы спектра является коэффициентом, на который умножается значение амплитуды из спектра начального шумового сигнала.

A = kn ∙ Aисх. ∙ koeff

kn = ((n - x1) / (x2 - x1) ∙ (y2 - y1) + y1) / 100, где

kn – коэффициент для n-го отсчета в частотной области;

n – номер отсчета в частотной области, лежащий между регуляторами (x1, y1) и (x2, y2);

x1 y1 x2 y2 – параметры двух соседних регуляторов: частота и уровень в процентах.

koeff – [коэффициент уровня сигнала](#_Описание_полей_формы)

1. Вычисляем новое комплексное число по формуле:

Im = A ∙ cos(ϕ) – мнимая часть

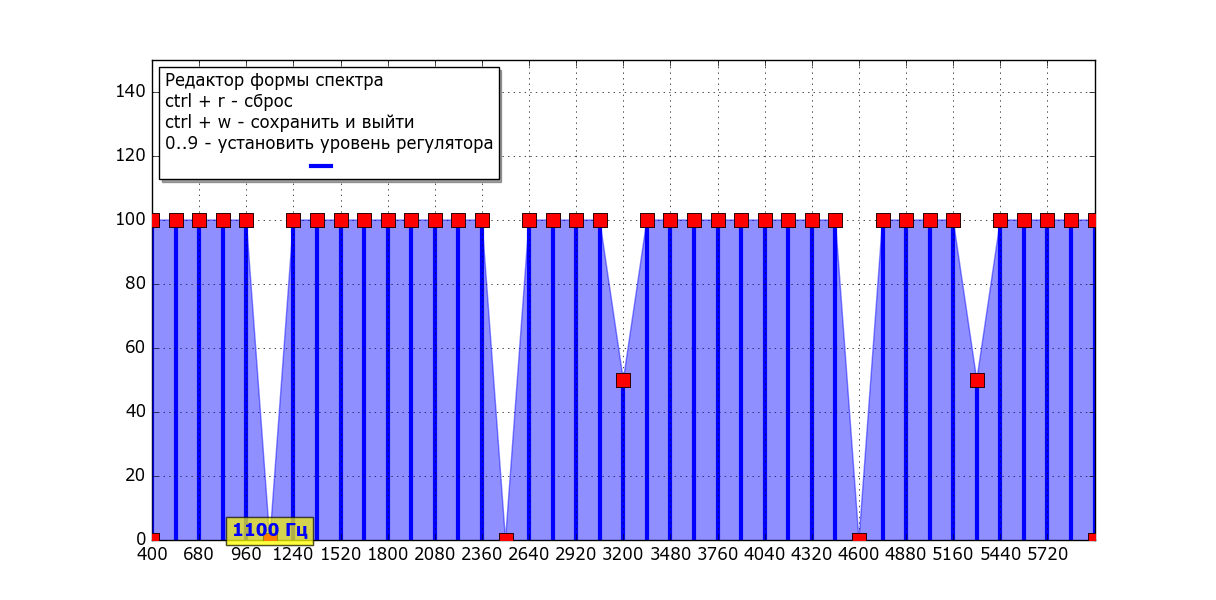
Re = A ∙ sin(ϕ) – реальная часть, где

A – значение [амплитуды](#_Описание_полей_формы), вычисленное ранее

ϕ – значение фазы, полученное из шумового сигнала

Если установлена галочка «Точно по форме», то в качестве Aисх. берётся значение амплитуды. В этом случае спектр конечного сигнала полностью соответствует заданной форме.

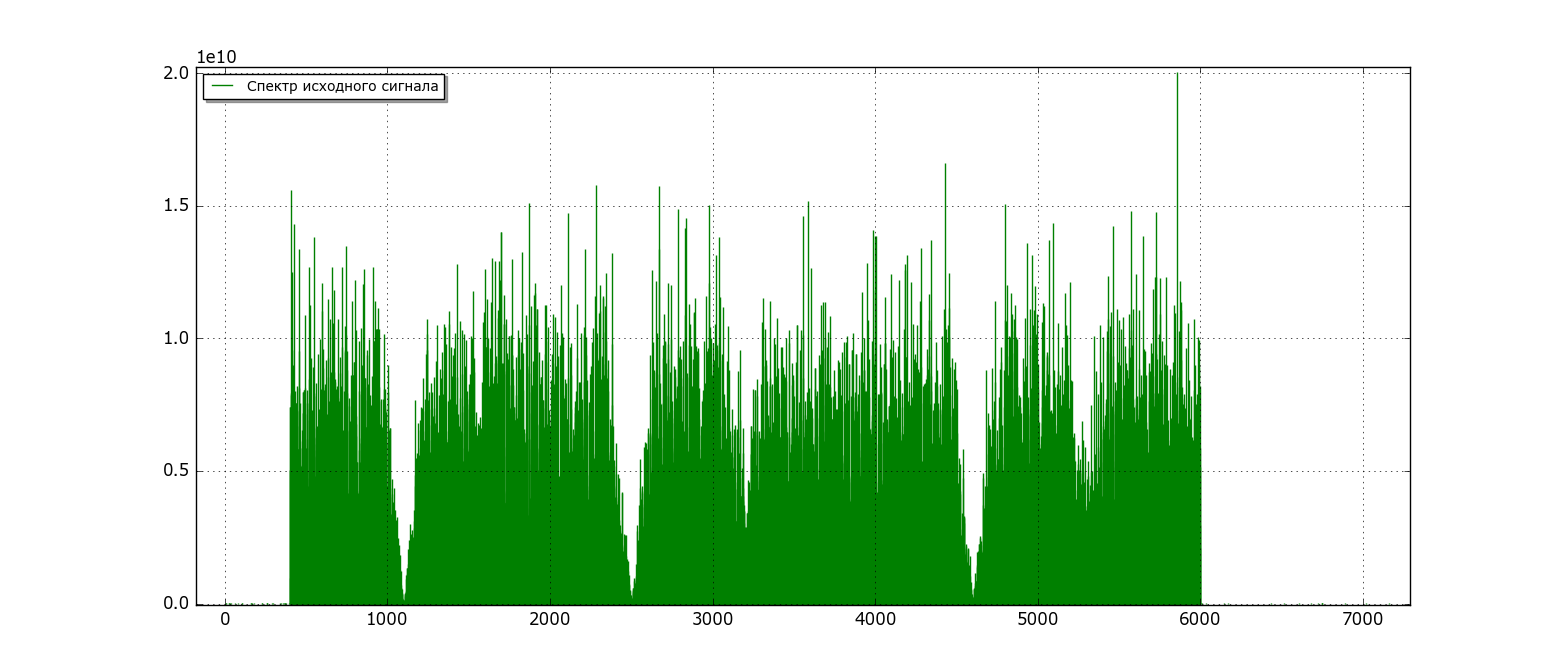
Для примера возьмем следующую форму спектра:



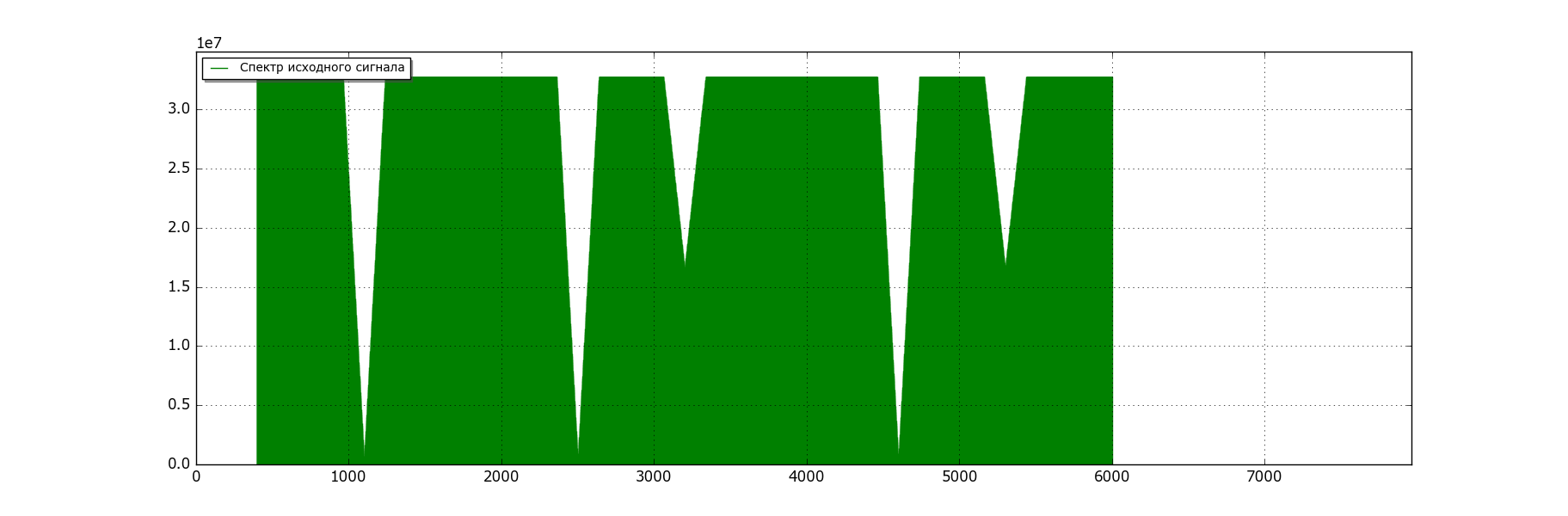
Частота начальная = 400 Гц

Частота конечная = 6 кГц

Тогда при снятой галочке «Точно по форме» получим сигнал со следующим спектром:



При установленной галочке «Точно по форме» получим сигнал со следующим спектром



# Наложение окна

Данные параметры служат для изменения формы начала и/или конца сигнала.

## Тип окна

Нет окна – не производится никаких модификаций сигнала.

Поверх сигнала – данные сигнала изменяются согласно заданной длительности и форме окна. Длительность сигнала при этом не изменяется

Добавить шум – к сигналу добавляется шумовой сигнал заданной длительности, и на него накладывается окно. Таким образом исходный сигнал не искажается, но становится длиннее.

Обрезать – от сигнала отрезается часть, заданной длительности.

## Форма окна

Определяет форму накладываемого окна – с-образное, cos или трапеция. См. примеры.

## Место наложения

Определяет место наложения окна. Возможны различные варианты:

В начало и конец каждой пачки, только в начало или только в конец каждой пачки, в начало и конец всего сигнала, только в начало или только в конец сигнала. См. примеры.

## Длительность окна

Длительность задает длительность в миллисекундах. Внимание! Можно указывать только целое количество. То есть минимальная длительность окна равна 1 мс.

## Примеры наложения окон.

#### Общие параметры

Тип сигнала – синус

Частота начальная = 4000 Гц

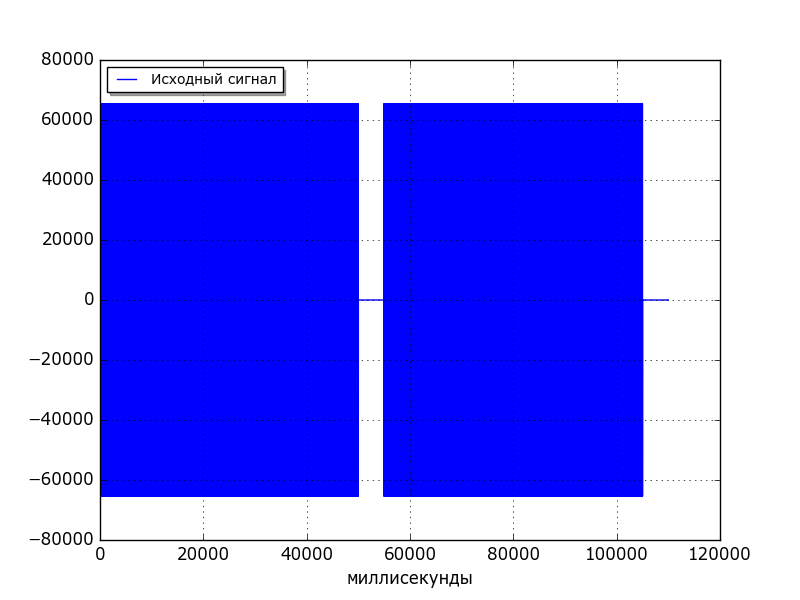
Общая длительность = 550 мс.

Длительность тишины = 50 мс.

Количество повторов = 2

Длительность окна = 100 мс.

#### Без окна

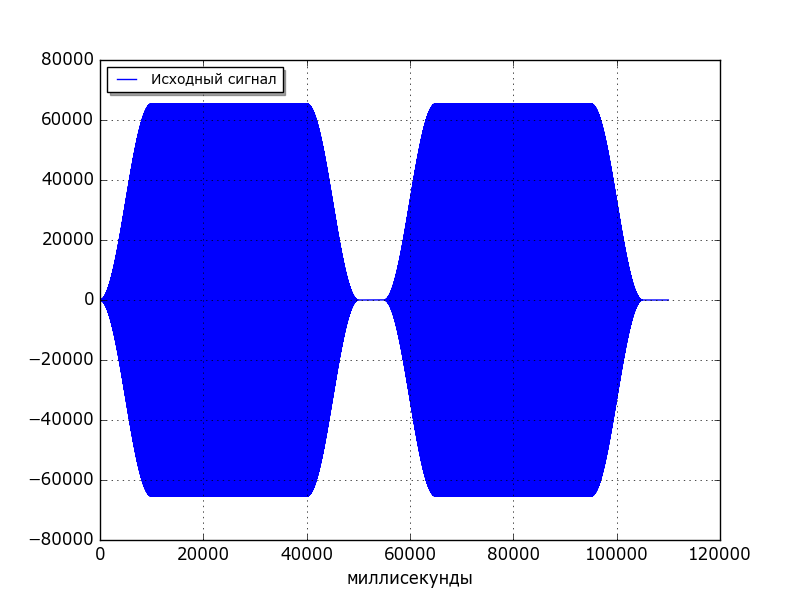


#### Пример 1.

Тип окна – Поверх сигнала

Форма – S-образное

Место наложения – начало и конец каждой пачки

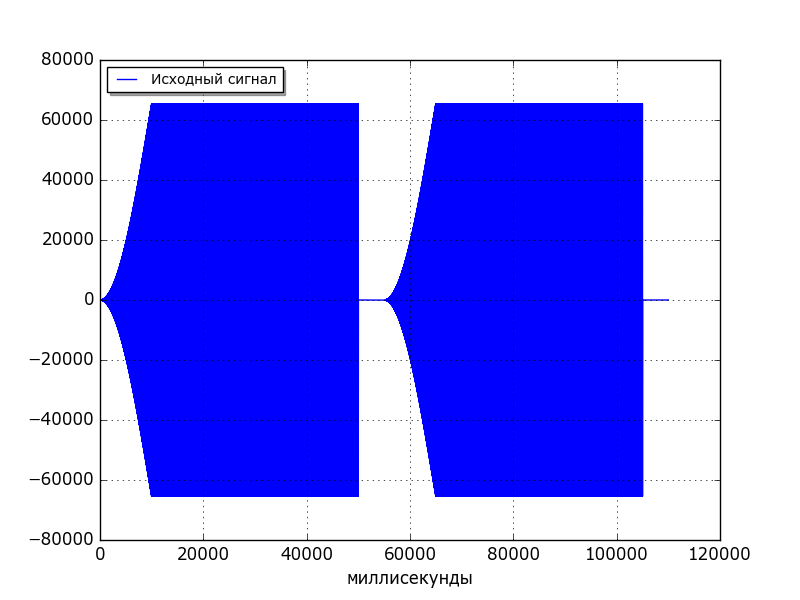


#### Пример 2.

Тип окна – Поверх сигнала

Форма – cos

Место наложения – начало каждой пачки

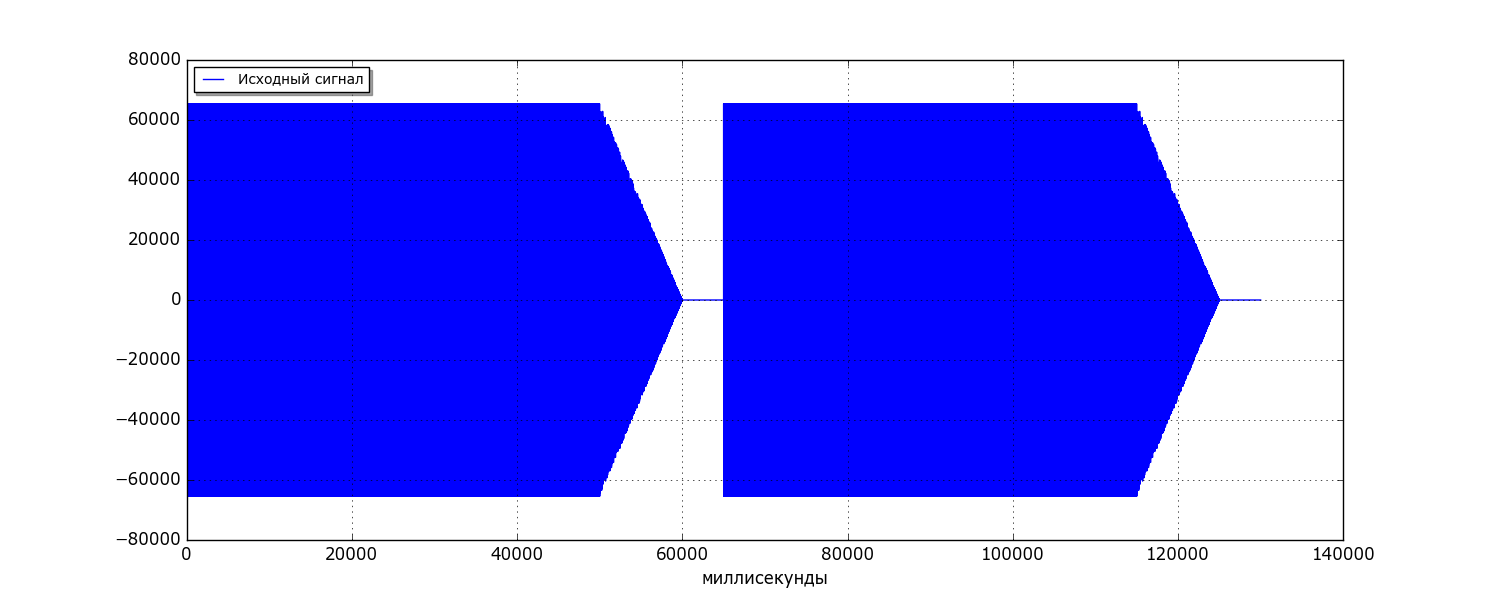


#### Пример 3.

Тип окна – Добавить шум

Форма – Трапеция

Место наложения –конец каждой пачки



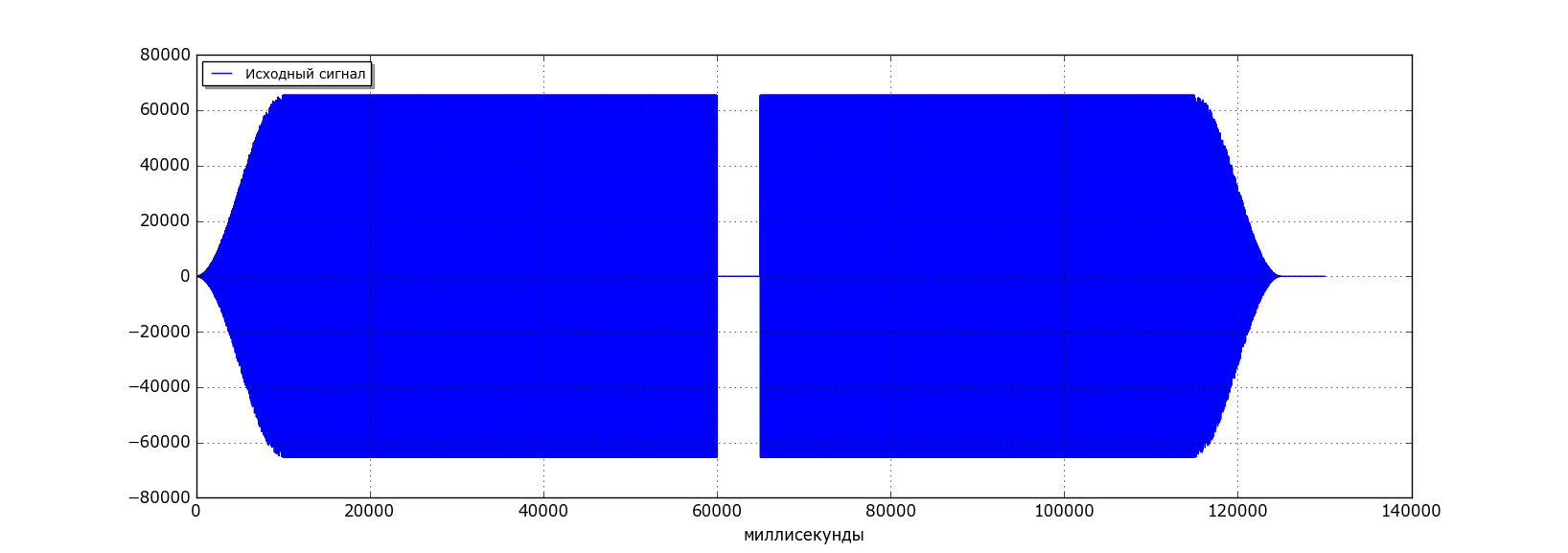
Длительность сигнала увеличивается на 100 × 2 = 200 мс. (длительность окна 100 мс. умножаем на количество повторов 2). Если наложить окно в начало и конец каждой пачки, то соответственно длительность увеличится на 100 × 2 × 2 = 400 мс.

#### Пример 4.

Тип окна – Добавить шум

Форма – S-образное

Место наложения –начало и конец сигнала

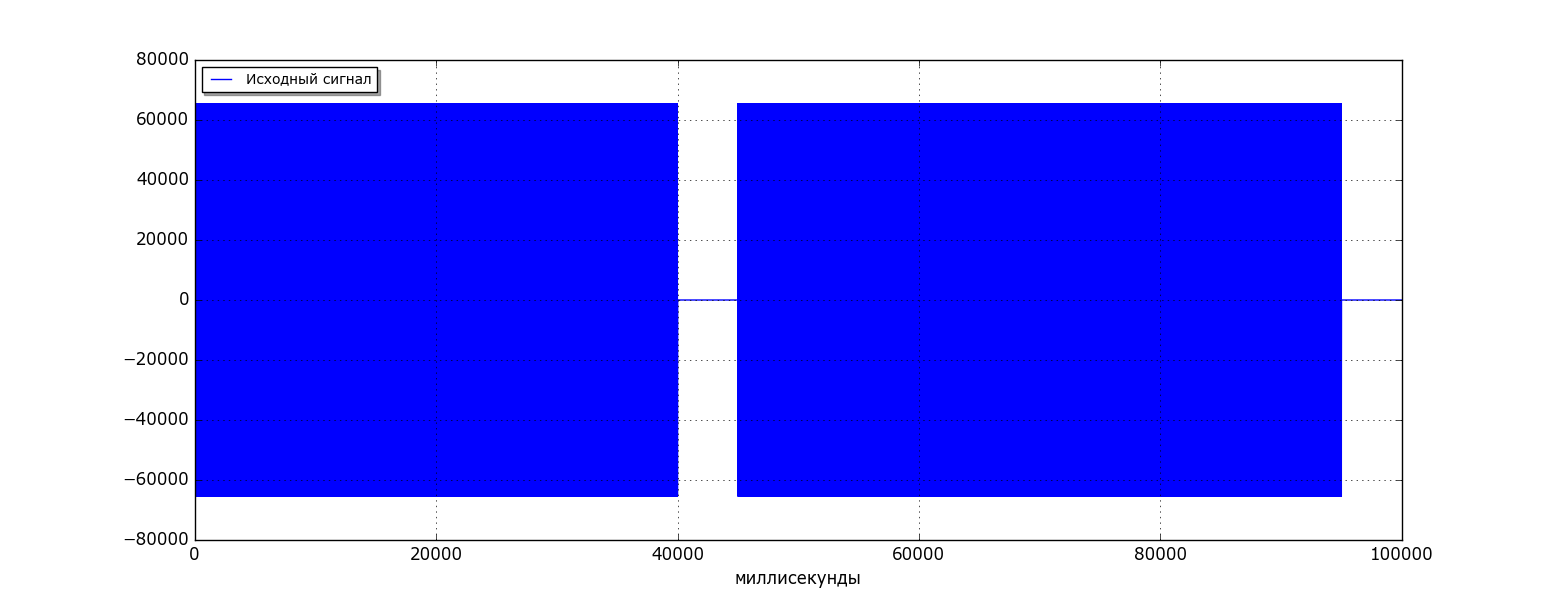


Длительность сигнала увеличивается на 100 × 2(начало и конец сигнала) = 200 мс.

#### Пример 5.

Тип окна – Обрезать

Место наложения –начало сигнала



В начале сигнала обрезано 100 мс.

# Преобразование ШИМ

Параметры этого раздела служат для отладки преобразования ШИМ и их изменять не нужно. Значения по умолчанию:

Кол-во каналов = 2

Пил на точку = 1

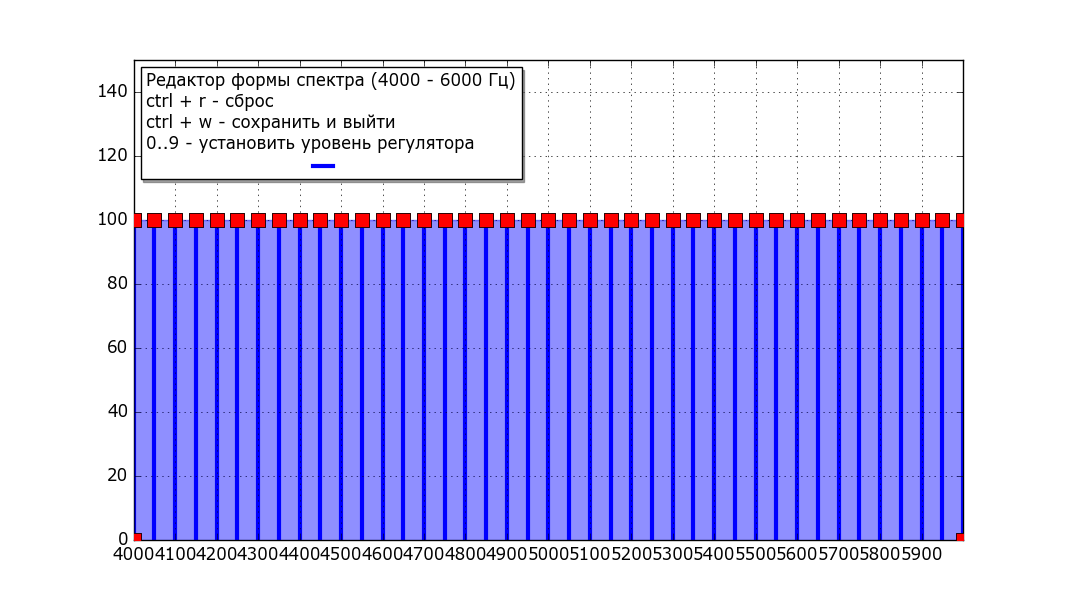
Размытие нуля = 0

Разрыв между каналами = 0

# Редактор формы спектра

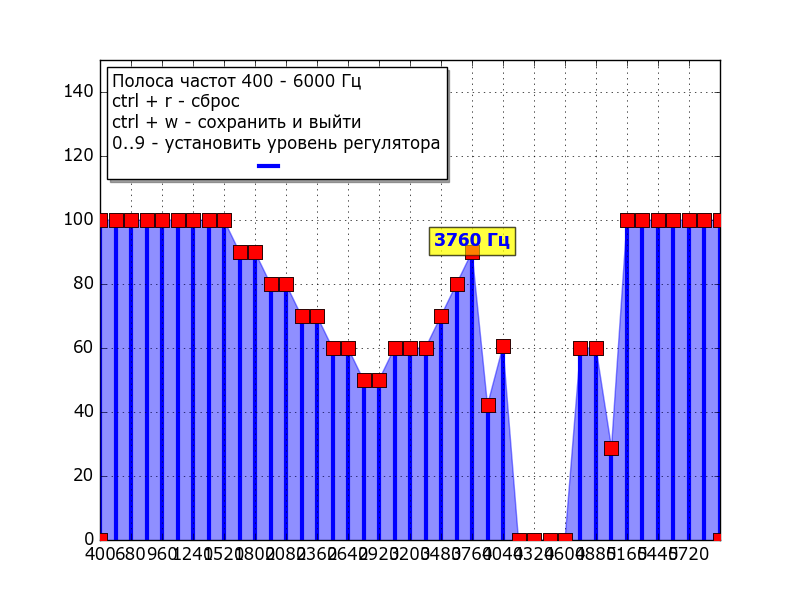
Позволяет редактировать форму спектра для генератора сигнала «[Из формы спектра](#_Из_формы_спектра)». Для работы с формой спектра, необходимо вначале указать имя и путь к файлу в поле «Файл». После указания имени файла можно создать новый файл или, если указан существующий файл, начать его редактирование. Форма спектра редактируется в полосе частот заданных полями «Частота начальная» и Частота конечная». Все частоты не входящие в данную полосу, подавляются (равны нулю).

При создании нового файла формы спектра редактор имеет вид:

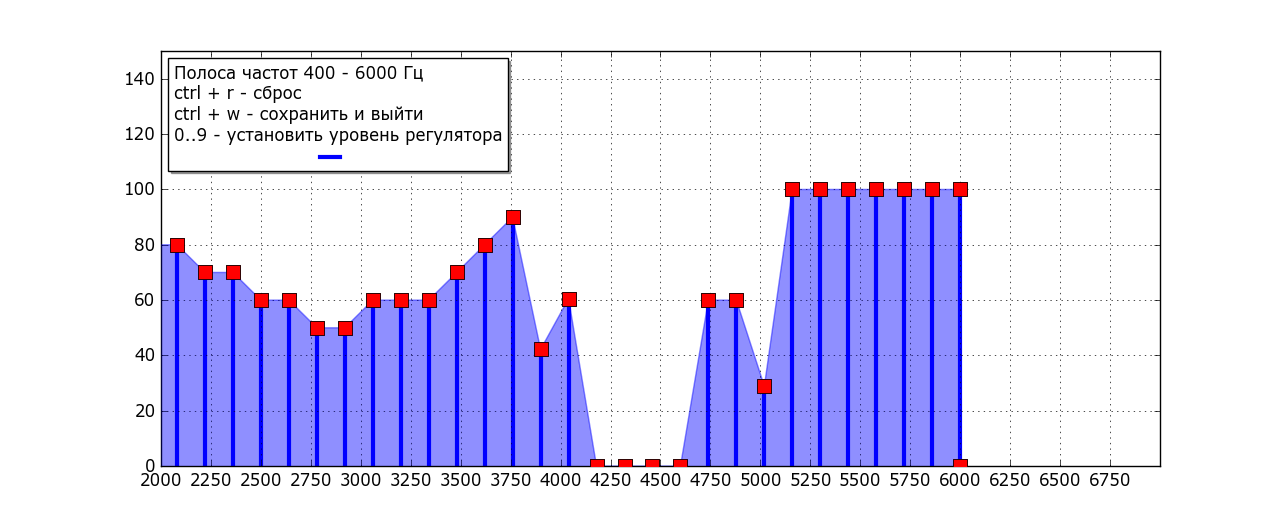


Красные квадраты это регуляторы, позволяющие задать уровень для соответствующей частоты в процентах. Частота, которой соответствует регулятор, высвечивается при наведении на него курсора мыши. По умолчанию количество регуляторов равно 41, независимо от ширины полосы частот. Если необходимо задать более точную форму спектра, то количество регуляторов можно изменить. Для этого в файле main.config необходимо изменить параметр **spec\_form\_edit\_control\_count**, задав ему положительное целое число больше 1. Количество регуляторов фиксируется при создании файла формы, поэтому изменение количества регуляторов влияет только на вновь создаваемые файлы кнопкой «Новая форма». При редактировании формы используется то количество регуляторов, которое записано в файле формы.

Вначале, при создании формы, уровень всех частот находится на одном уровне. Двигая регуляторы вверх-вниз можно задать необходимую форму АЧХ. После сохранения файла формы формируется спектр сигнала в соответствии с заданной формой. Затем производится обратное преобразование Фурье и мы получаем сигнал с заданной АЧХ. Если не изменять форму после создания, то будет сгенерирован шумовой сигнал с равномерной АЧХ.



Полоса частот, для которой создана данная форма, также записывается в файл при сохранении. На приведенном примере видно, что при создании файла начальная частота была равна 400 Гц, конечная 6000 Гц. Если в дальнейшем изменить значения частот и открыть файл формы для редактирования, то редактор все равно будет работать в той полосе частот, которая была задана при создании файла. Например, если поставить начальную частоту 2000, а конечную 7000, то мы увидим следующую картину:

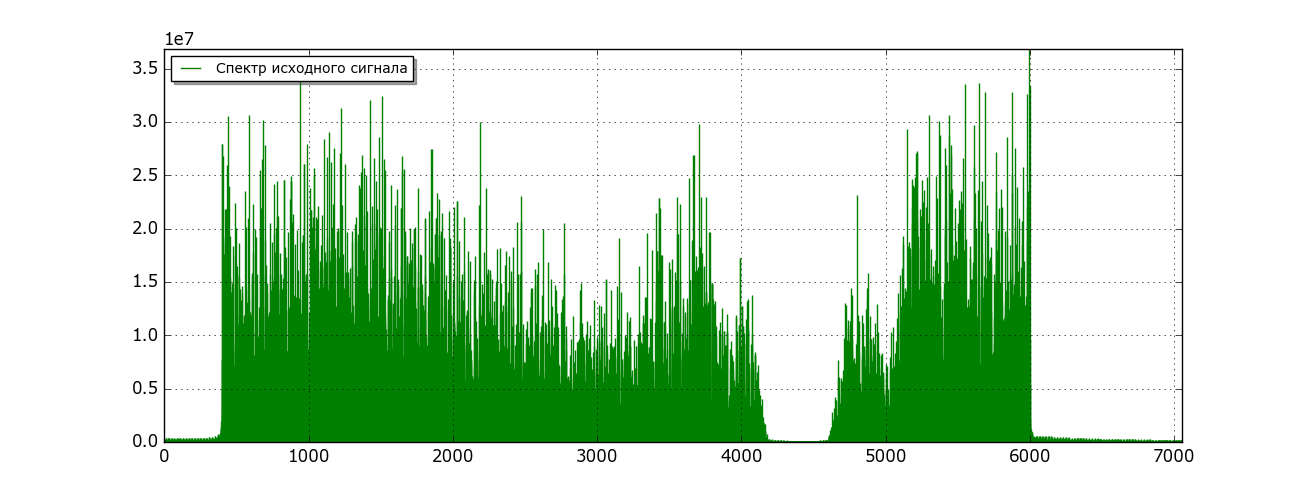


То есть область редактирования просто сдвинулась вправо. И используя инструменты редактора можно двигать область редактирования в нужное место. При формировании сигнала будет использоваться полоса частот, записанная в файле.

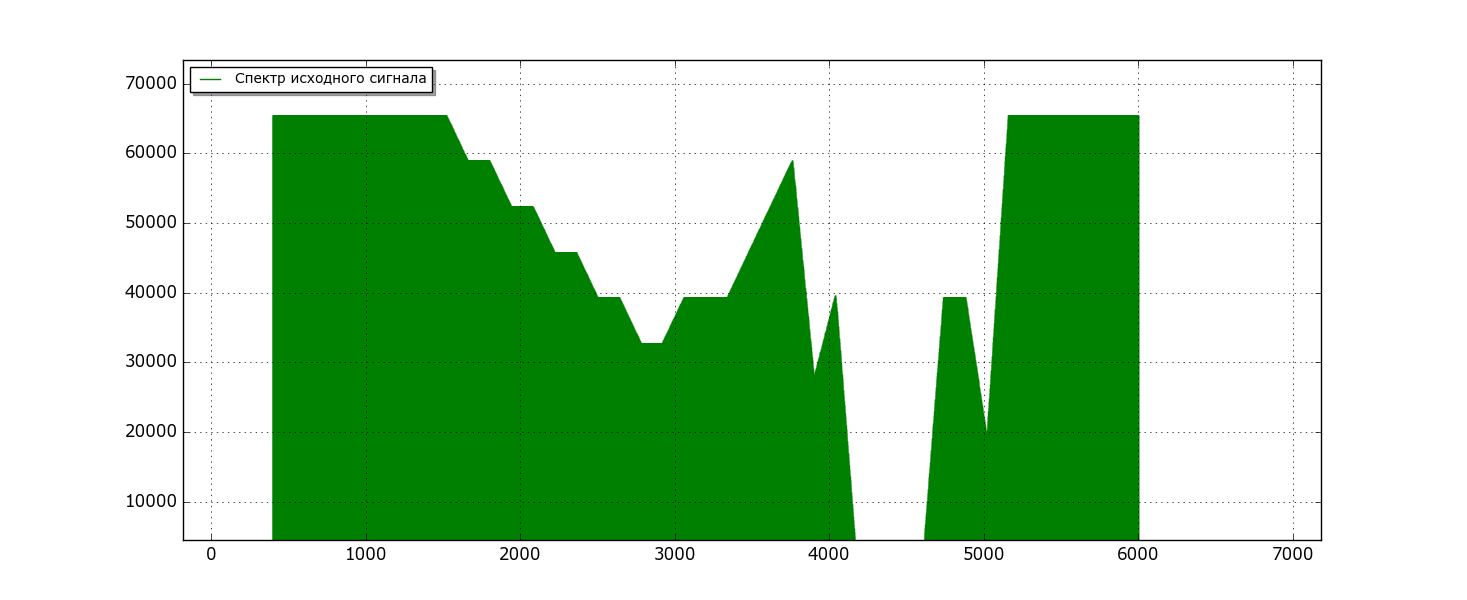
Исходя из состояния галочки «Точно по форме», по-разному формируется спектр конечного сигнала. Для понимания различия, необходимо знать, что для формирования спектра конечного сигнала используется спектр случайного шумового сигнала, из которого выбираются значения фаз. Если галочка «Точно по форме» НЕ установлена, то из спектра случайного шумового сигнала выбираются также значения амплитуд, которые умножаются на соответствующий уровень нашей формы спектра.

Если галочка «Точно по форме» установлена, то в качестве значения амплитуды используется значение параметра «Амплитуда», которое также умножаются на соответствующий уровень нашей формы спектра. Вычисления описаны в описании сигнала «[Из формы спектра](#_Из_формы_спектра)».

Пример, иллюстрирующий спектр полученного сигнала при снятой галочке «Точно по форме»:



И при установленной:



# Фильтрация

Позволяет применить полосовой фильтр к шумовому сигналу. Генерируется [шумовой сигнал](#_Шум.), затем этот сигнал с помощью БПФ преобразуется в спектр, в котором удаляются все частоты не входящие в заданную полосу частот. Затем выполняется обратное преобразование Фурье.

Параметры «Мин. частота» и «Макс. частота» задают полосу частот для фильтра.

Внимание! Галочка «Применить полосовой фильтр» влияет только на сигнал типа «Шум».

# Выгрузка

В этом разделе находятся параметры для управления выгрузкой сгенерированного сигнала на приемное устройство.

Выгрузить на устройство – включает/отключает выгрузку.

IP и порт – адрес и порт приемного устройства.

Режим работы – ONE или LOOP, соответственно однократное воспроизведение сигнала или циклическое.

Отправить STOP – отправляет приемному устройству команду STOP, после которой воспроизведение прекращается.

Выгрузить файл – позволяет выгрузить на приемное устройство ранее сохраненный файл .shim

# Выходные файлы

Определяет какие типы файлов будут сформированы после генерации сигнала.

ШИМ – файл с расширением shim, файл, выгружаемый на приемное устройство. Если данная галочка не устоит, то выгрузка не производится, даже если установлена галочка «Выгрузить сигнал на устройство».

WAV – формируется звуковой файл с расширением wav, со следующими параметрами: формат PCM, каналов 1 (моно), заданная частота дискретизации, 4 байта на один сэмпл (int).

Matlab – формируется файл матлаб, для последующего анализа сгенерированного и принятого сигналов.

# Отрисовка

Определяет какие графические данные выводить на экран.

Рекомендуется выводить на экран только ту информацию, которая нужна в данный момент, так как большой объем графики может тормозить систему.

# Общее

Задаются пути к рабочему каталогу и шаблон имен файлов.

Рабочий каталог – указывает путь в который будут помещаться все сформированные файлы.

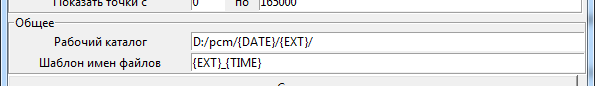
Шаблон имени файла – задает общий шаблон имени для всех формируемых файлов. Формируемые файлы будут иметь одинаковые имена, но разные расширения.

При указании рабочего каталога и имени файлов возможно использовать следующие регулярные выражения: {DATE}, {TIME}, {EXT}.

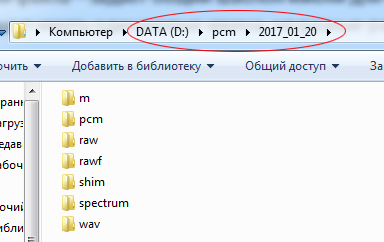
{DATE}, {TIME} подставляют соответственно дату и время в нужное место. Формат даты yyyy\_mm\_dd, формат времени hh\_mm\_ss.

{EXT} заменяется на расширение файла. То есть при формировании исходного сигнала {EXT} будет заменено на raw, при формировании ШИМ файл на shim, при формировании wav файла на wav и т.д.

Пример 1. При следующих параметрах:



После генерации сигнала мы получим на диске:



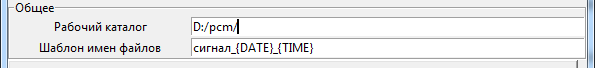
Все файлы будут помещены каждый в соответствующую папку с именем, содержащим время создания.

raw\_13\_48\_53.raw

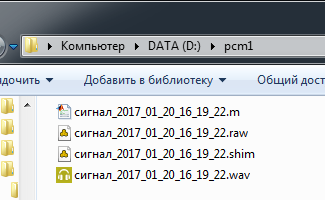
wav\_13\_48\_53.wav

shim\_13\_48\_53.shim

Пример 2. При следующих параметрах:



Получим на диске:



# Описание кнопок

Старт – начинает генерацию сигнала и выполнение прочих операций. Последовательность действий следующая:

1. сохранение текущей конфигурации в log файл,
2. генерация сигнала,
3. формирование файла шим,
4. выгрузка его на устройство,
5. формирование файлов wav и matlab,
6. вывод графики на экран.

Результат выполнения всех этапов выводится на экран.

Сохранить – сохраняет текущую конфигурацию без генерации сигнала.

Загрузить конфигурацию – позволяет загрузить конфигурацию из log файла или config файла, сохраненных ранее.

# Логирование

При запуске генерации файла текущая конфигурация сохраняется в файл log. Файл log сохраняется с теми же правилами, что и прочие файлы – в рабочий каталог и с именем, заданным шаблоном.

В файл log записывается текущая конфигурация, то есть значения всех параметров на момент запуска. В дальнейшем, при необходимости можно в точности восстановить данную конфигурацию нажав кнопку «Загрузить конфигурацию».

Также, если генерируется сигнал типа «Из формы спектра», то создается резервная копия текущего файла формы спектра с расширением spectrum. Который также можно использовать в дальнейшем.

Можно отключить логирование. Для этого необходимо назначить параметру save\_log из файла main.config значение False. Соответственно значение True включает логирование.