

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**  
**НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Физический факультет**

**Кафедра радиофизики**

С.С. Середняков, А.С. Стюф

**ЦИФРОВЫЕ МЕТОДЫ ГЕНЕРИРОВАНИЯ, РЕГИСТРАЦИИ И  
ОБРАБОТКИ СИГНАЛОВ**

Лабораторная работа №6

(обучающий комплекс для выполнения  
новых лабораторных работ )

Новосибирск  
2012

До недавнего времени задача генерирования и регистрирования электрических импульсов различной формы и частоты решалась с помощью специализированных приборов – генераторов импульсов и быстрых осциллографов, управляемых при помощи переключения ручек на передней панели. С развитием цифровой техники появились устройства аналогичные по функциональности, но совместимые со стандартом магистрально-модульных систем и управляемые от персонального компьютера. Последняя возможность сделала возможность разрабатывать специализированные программы, осуществляющие как генерацию электрического сигнала нужной формы и частоты, так и анализ, и обработку измеряемого сигнала.

Данный обучающий комплекс ставит своей целью приобретение студентами навыков работы с данными устройствами, управления ими с помощью прикладного программного обеспечения и применение этих навыков на практике.

Обучающий комплекс включает в себя аппаратное и программное обеспечение, необходимые для выполнения данной работы, а также текст методического пособия.

Работа предназначена для выполнения студентами 3 курса ФФ.

Авторы

кандидат тех. наук Середняков С.С., А.С. Стюф

Обучающий комплекс для выполнения новых лабораторных работ подготовлен в рамках реализации Программы развития НИУ-НГУ на 2009–2018 г. г.

## Введение

Весь реальный мир является миром величин аналоговых, принимающих бесконечное количество сколь угодно близких значений из непрерывного множества и описываемых непрерывной функцией времени. Аналоговые сигналы используются для представления непрерывно изменяющихся тока, температуры, давления, напряжения, угла и т. п. Из вводных работ практикума вам известно, что для перевода с «цифрового языка» компьютера на «аналоговый язык» периферийных устройств и наоборот, применяются цифроаналоговые и аналого-цифровые преобразователи. Это наиболее массовые технические средства автоматизации.

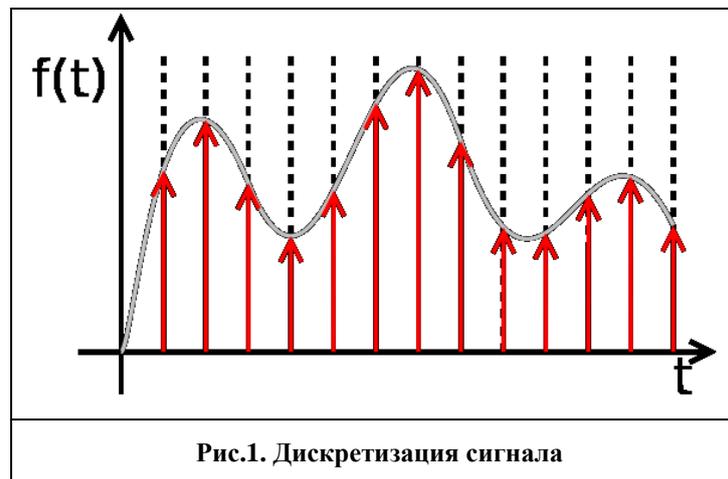
Цифровые методы генерирования и регистрации аналоговых сигналов в последние полтора десятилетия получили широкое распространение. Это вызвано с одной стороны широким внедрением компьютеров во все сферы деятельности людей, а с другой – бурным прогрессом в полупроводниковой электронике. Цифроаналоговые и аналого-цифровые преобразователи, работающие с электрическими сигналами, известны уже около ста лет. Тем не менее, создание первых цифровых генераторов сигналов и цифровых осциллографов стало возможным около 30 лет назад - с появлением быстродействующих преобразовательных устройств и микросхем памяти достаточно большого объёма.

В данной работе вам предстоит понять, как устроены эти приборы, и чем они характеризуются, а также научиться работать с ними и производить необходимую обработку данных.

## 1. Принципы преобразования сигналов

Рассмотрение начнём с цифровых методов регистрации аналоговых сигналов. В процедуре регистрации выделяют три фазы: дискретизация, квантование и запоминание массива полученных цифровых отсчётов.

*Дискретизацией* называется замена непрерывного сигнала  $f(t)$  на последовательность значений этого сигнала (выборок сигнала), взятых в моменты времени, отстоящие друг от друга на постоянную величину  $h$ , которая называется интервалом (шагом) дискретизации (Рис.1). Величина  $F_s = 1/h$ , называется частотой дискретизации. Интуитивно ясно, что чем меньше значение  $h$ , тем точнее  $f(t)$  может быть описана дискретными значениями.

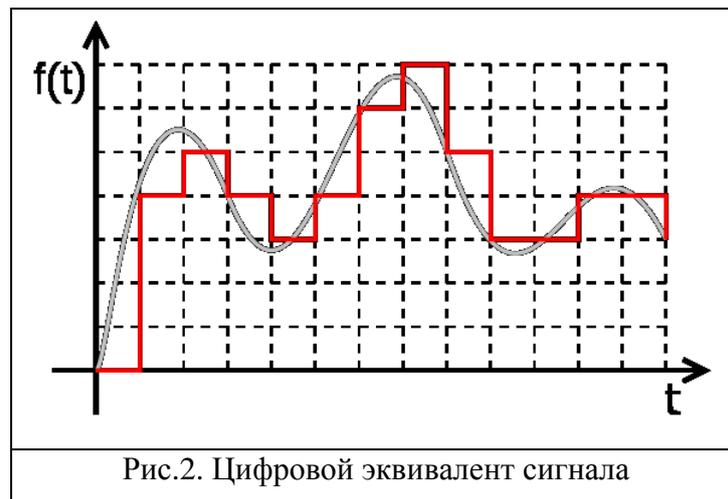


Предельное соотношение между максимальной частотой  $F_M$  в спектре сигнала и интервала  $h$  устанавливает теорема отсчётов, называемая в отечественной литературе как теорема Котельникова [1]. Теорема утверждает, что если непрерывная функция времени  $f(t)$  имеет спектр, ограниченный частотой  $F_M$ , то эта функция может быть полностью восстановлена по дискретным выборкам функции  $f(h \cdot n)$ , где  $n=1, 2, 3, \dots$ , а  $h = \frac{1}{2 \cdot F_M}$ .

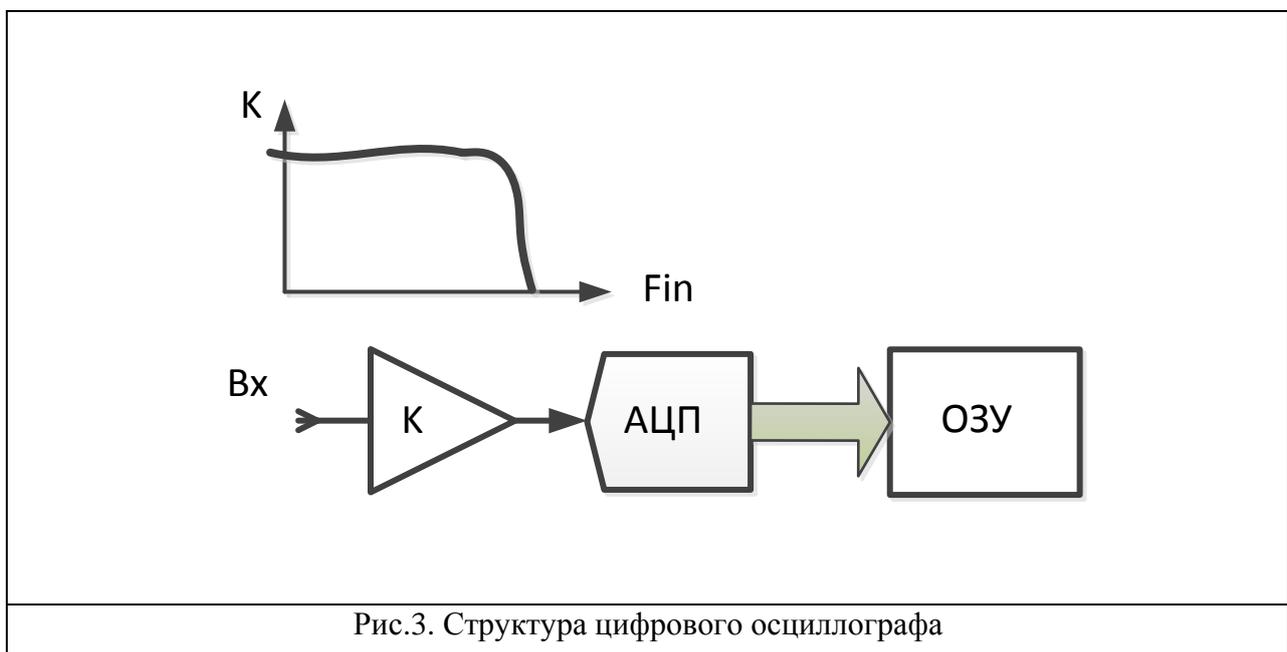
Непосредственное применение теоремы Котельникова на практике встречает ряд трудностей, т.к. большинство сигналов имеет слабо ограниченный спектр. Поэтому для определения интервала  $h$  применяют различные упрощённые оценки, сделанные для линейной интерполяции при восстановлении и использующие в качестве параметра длительность или характерное время изменения сигнала. Так, например  $h$  можно определять из выражения:  $h = \tau \cdot 2 \cdot \sqrt{\delta}$ , где  $\tau$  – постоянная времени изменения сигнала, а  $\delta$  – погрешность линейной аппроксимации.

После дискретизации, каждая полученная выборка подвергается процедуре *квантования*. Эта процедура заключается в том, что непрерывная по амплитуде выборка сравнивается с набором дискретных, равноотстоящих с шагом  $q$  (квант преобразования), значений, после чего каждому отсчету присваивается номер наиболее близкого значения (число). Понятно, что чем меньше квант  $q$ , тем точнее получаемые отсчёты описывают реальный сигнал.

В результате выполнения дискретизации и квантования непрерывный аналоговый сигнал заменяется на массив дискретных цифровых значений (Рис.2), то есть цифровой эквивалент сигнала, который затем подвергается обработке.



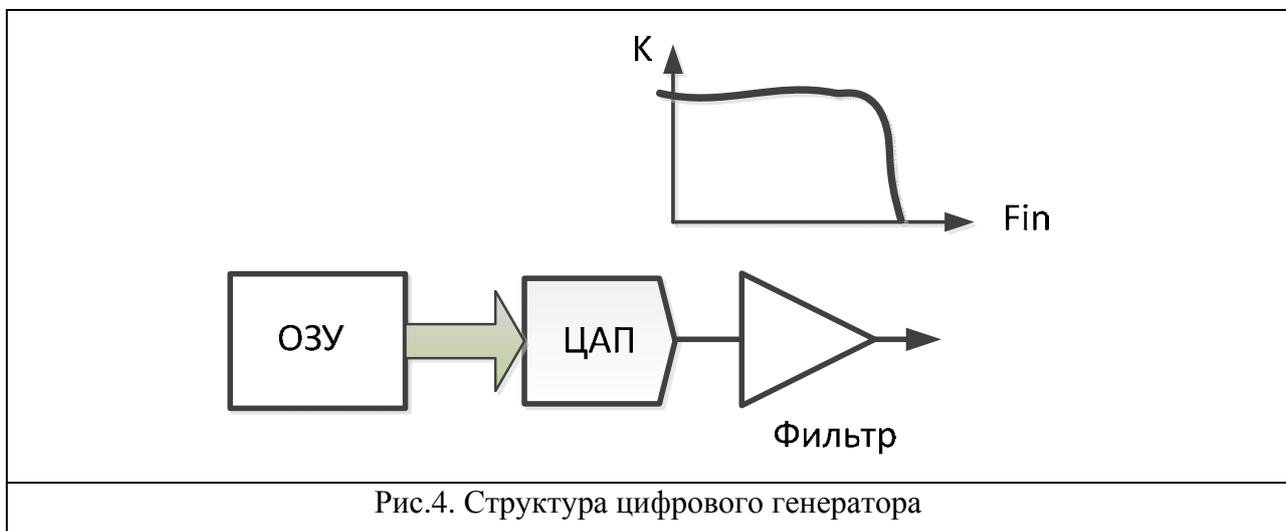
И, наконец, последний этап – запись и сохранение полученных цифровых отсчётов в запоминающем устройстве. Эта процедура необходима в связи с тем, что быстродействие магистрали передачи данных ограничено как аппаратными, так и программными факторами, и оказывается недостаточным. Например, если при восстановлении колоколообразного сигнала с длительностью 1 мкс, чему соответствует  $\tau$ , равное примерно 0.25 мкс, принять погрешность интерполяции 1%, то получим  $h=50$  нс, т.е. частота выдачи данных составит 20МГц.



В цифровых осциллографах процедуры дискретизации и квантования выполняются в одном и том же устройстве – аналого-цифровом преобразователе. Буферная память представляет, как правило, отдельный узел. Есть ещё и третье устройство – фильтр, ограничивающий полосу частот, что минимизирует ошибки, вызываемые невыполнением теоремы Котельникова. Обычно, частотный фильтр совмещён с входным усилителем. Таким

образом, три элемента являются обязательными для всех цифровых осциллографов (Рис.3): входной усилитель, ограничивающий сигнальную полосу, аналого-цифровой преобразователь и оперативное запоминающее устройство.

Обратимся теперь к цифровым генераторам сигналов. Основным элементом цифрового генератора сигналов является быстродействующий цифроаналоговый преобразователь. Вспомнив, что назначение этого устройства – преобразование входного кода в выходной аналоговый сигнал, нетрудно догадаться, что если данные на входе описывают какой-либо сигнал, то на выходе ЦАП'а и будет получен этот сигнал, хотя и с некоторыми погрешностями. Принципиальная погрешность так же, как и в цифровых осциллографах, вызвана двумя факторами: интервалом между кодовыми посылками и величиной кванта ЦАП'а. Влияние и того и другого фактора проявляется как слишком заметные «ступеньки» в выходном сигнале. По амплитуде уменьшение погрешности устраняется увеличением разрядности ЦАП'а, а по оси времени – повышением частоты дискретизации и установкой на выходе ЦАП'а фильтра, сглаживающего ступеньки. Для того, чтобы поток данных на входе не зависел от быстродействия системной магистрали, цифровой эквивалент сигнала записывают в буферное ОЗУ, откуда коды с частотой дискретизации поступают на вход ЦАП'а. Таким образом, структура цифрового генератора сигналов (Рис.4) похожа на структуру «обращённого» цифрового осциллографа.



## 2.Оборудование, используемое в работе

В данной работе используется два модуля фирмы National Instruments: генератор сигналов NI PXI-5402 и высокоскоростной цифровой осциллограф NI PXI-5114. Оба

устройства являются модулями, совместимыми со стандартом PXIe, и установлены в кейт на вашем рабочем месте (Рис.5).



Рисунок 5. Кейт с модулями PXI, используемыми в работе

Каждый из упомянутых выше модулей имеет на своей передней панели несколько разъемов (Рис 6.). Кроме разъемов, предназначенных для подключения входов и выходов сигналов (разъемы “CH0”, “CH1”), на передней панели представлены разъемы для внешней синхронизации (“SYNC OUT / PFI 0”, “PFI 1”), а также разъем для внешнего триггера (“TRIG”). Для выполнения данной работы необходимо использовать нулевые входные и выходные каналы обоих модулей или разъемы “CH0”. Для выполнения заданий 1-4 необходимо, чтобы эти разъемы были соединены между собой. При выполнении задания 5 между данными разъемами нужно подключить испытываемую схему.



Рисунок 6. Передние панели модулей NI PXI-5402 и NI PXI-5114

## 2.1. Генератор сигналов NI PXI-5402.

Данный модуль предназначен для генерации сигналов различной формы. Модуль может выдавать следующие стандартные сигналы с неизменной частотой:

- синусоидальный;
- прямоугольный;
- треугольный;
- пилообразный.

Предельная частота следования для синусоидальных и прямоугольных напряжений составляет 20 МГц, для остальных типов – 1 МГц.

Кроме того, модуль может генерировать сигналы стандартной формы с изменяющейся частотой. Максимальная частота - 20 МГц и 1 МГц. При подготовке последовательности указывается минимальная и максимальная частота, число шагов по частоте и время исполнения одного шага.

Также, генератор может выдавать белый шум, т.е. случайный сигнал, спектральная плотность которого равномерна в рабочей полосе частот.

Внутреннее устройство NI PXI 5402 изображено на рисунке 5:

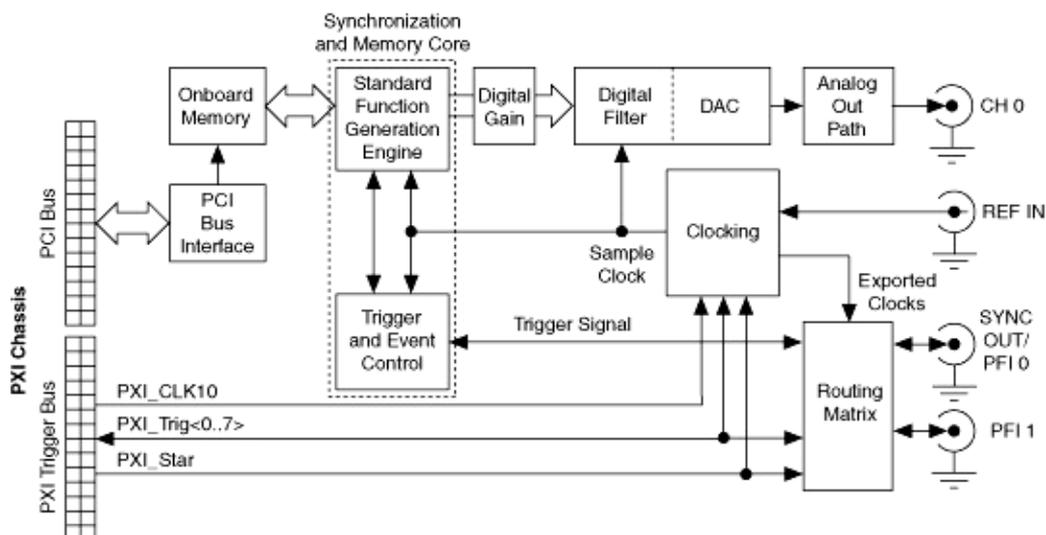


Рис.5. Блок-схема модуля NI PXI 5402

Основные узлы модуля следующие:

- внутренняя память (Onboard Memory). Используется для хранения таблиц сигналов произвольной формы, которые готовит и загружает пользователь;

- блок-генератор стандартных функций (Standard Function generation engine). Выдает последовательность 16-разрядных слов, соответствующих выбранному стандартному сигналу. Для подготовки данных, описывающих стандартные сигналы, используется так называемый непосредственный (или прямой) цифровой синтез –DDS (Direct Digital Synthesis). Этот способ использует память-таблицу (16 K look-up table) для подготовки данных. Подробности о DDS можно прочитать в [<http://www.analog.com/library/analogDialogue/archives/38-08/dds.pdf>]
- Тактовый генератор (Clocking). Основное назначение - тактирование блок-генератора функций стабильной частотой 100 МГц.
- Цифро-аналоговый преобразователь (DAC);
- Выходной усилитель с фильтром (Analog out Path).

Центральным устройством модуля является генератор стандартных функций, который генерирует последовательность цифровых кодов с тактовой частотой в зависимости от необходимой частоты и типа генерируемого сигнала. После прохождения блоков усиления (Digital Gain) и фильтрации (Digital Filter), цифровые коды подаются на ЦАП, который и генерирует аналоговый сигнал. Полученный сигнал перед подачей на выход также проходит процедуры усиления и фильтрации (Analog Out Path). В режиме генерации частотной последовательности генератор стандартных функций считывает параметры импульсов из встроенной памяти, и в соответствии с ними генерирует цифровой код.

## **2.2. Цифровой осциллограф NI PXI-5114**

Этот модуль используется для регистрации формы (цифровой осциллограммы) сигнала, поданного на вход. Если говорить вышеупомянутыми терминами, модуль производит дискретизацию, квантование и запоминание цифровых отсчётов. Основные параметры модуля следующие:

1. Аналоговых входов – 2.
2. Разрешение – 8 бит.
3. Макс. частота дискретизации – 250 Msamples/sec
4. Объем памяти – 256 Мб на один канал.

Блок-схема модуля приведена на рисунке 5:



Идеология работы с данными приборами аналогична работе с любым другим устройством. Сначала нужно установить соединение с модулем, для чего нужно использовать функции *Fgen\_Connect(PXIPos)* и *Scope\_Connect(PXIPos)* для модулей PXI-5402 и PXI-5114 соответственно. Обе функции принимают переменную – номер позиции модуля в крейте и возвращают результат операции – *VI\_TRUE* в случае успешного соединения и *VI\_FALSE* - если модуль не найден.

После успешного соединения с модулем можно начинать управлять модулем. Например, генератор функций можно сразу включить в режим генерации импульсов функцией *Fgen\_StartStandartWaveForm(,,)*. После этого можно изменить основные параметры генерируемых импульсов – амплитуду и частоту при помощи функций *Fgen\_SetWaveFormAmplitude()* и *Fgen\_SetWaveFormFrequency()*. Для запуска генерации последовательности импульсов с переменной частотой, нужно сначала сгенерировать список частот функцией *Fgen\_GenerateFreqList(,,)*, после чего сгенерированные массивы передать на выполнение функции *Fgen\_StartFreqListGenerate(,,)*, которая и запустит процесс генерации данной последовательности. Перед окончанием работы с модулем нужно остановить генерацию импульсов функцией *Fgen\_Stop()*.

Для успешной работы модуля PXI-5114 нужно, предварительно установить соединение с ним (см.выше), и включить модуль PXI-5402 в режим генерации (чтобы было что оцифровывать). Перед процессом оцифровки нужно задать основные параметры – частоту дискретизации и число измерений, которое произведет модуль, при помощи функции *Scope\_SetHorizontalParameters(,)*. После этого можно запустить функцию *Scope\_StartMeasureWaveForm(,,)*, которая запускает процесс оцифровки, ждет указанное количество времени, считывает массив оцифрованных значений из памяти модуля и выдает их в программу через аргумент функции. Также в процессе работы с модулем может быть полезна функция *Scope\_GetHorizontalParameters(,)*, которая считывает параметры оцифровки, которые заданы в модуле в данный момент, и выдает их также через аргументы функции.

После окончания работы с модулями нужно от них отсоединиться при помощи функций *Fgen\_Disconnect()* и *Scope\_Disconnect()*.

Точный синтаксис функций библиотеки PXIDev приведен в приложении 1. Для использования данной библиотеки в программе, нужно подключить её, а также файл PXIDev.h, к проекту. Для этого нужно выполнить команды меню Edit->Add Files To Project-> Include(\*.h)... , ->Library(\*.lib)... .

## Практические задания

1. Напишите программу для управления генератором функций (NI PXI-5402). Программа должна создать панель управления генератором, элементы управления которой позволят установить форму сигнала, его частоту и амплитуду.
2. Добавьте в программу функционал, необходимый для управления цифровым осциллографом (NI PXI-5114). Программа должна запускать прибор в режим непрерывных измерений. Основные параметры измерений – частоту дискретизации и количество измерений программа должна считывать из соответствующих элементов управления. Массив считанных данных должен выводиться на график. Для этого лучше воспользоваться стандартной функцией библиотеки *NI* - ***PlotWaveform()***.

Для лучшей организации программы и её последующей модификации, желательно, чтобы программа содержала следующие функции:

***InitDevices()*** – подключение к обоим приборам. В случае если одно из устройств не найдено, программа должна выдавать сообщение об ошибке.

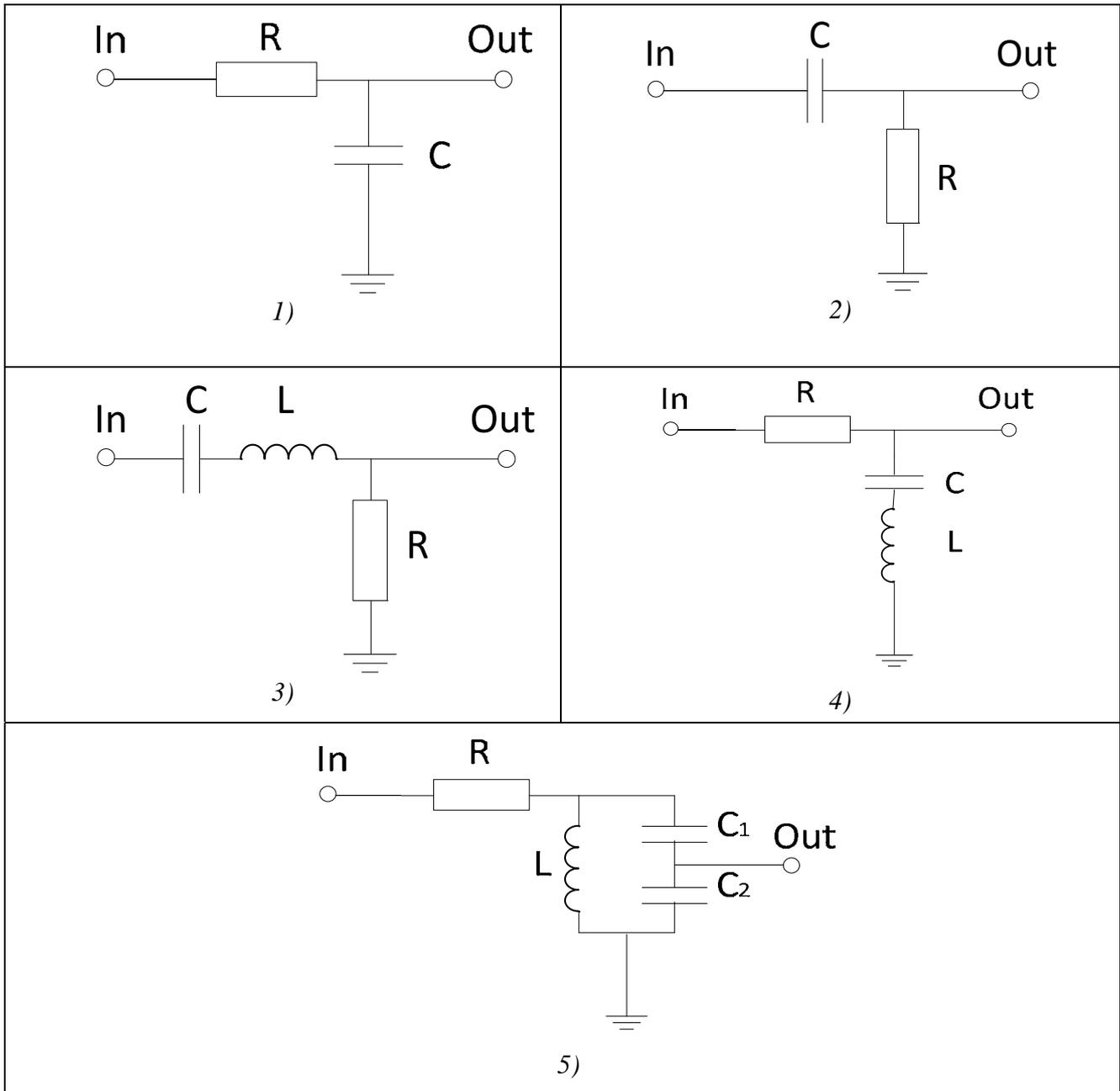
***StartGeneration()*** – считывание с элементов управления окна программы параметров генерации сигнала, конфигурирование прибора и старт генерации импульсов.

***ReadPlotWaveForm()*** – Запуск осциллографа на измерение, ожидание, чтение оцифрованного сигнала из его памяти и вывод графика на экран.

***StopGeneration()*** – остановка процесса генерации импульсов.

***CloseDevices()*** – отключение от обоих устройств.

3. Добавить в программу вычисление спектра сигнала при помощи быстрого преобразования Фурье (аналогично заданию из работы 1), вывести вычисленный спектр на график. Максимальная частота на графике (масштаб) должна задаваться из программы.
4. Добавить в программу возможность включения генерации импульсов переменной частоты (частотного списка). Параметры списка – минимальная, максимальная частота, число шагов, продолжительность шага программа должна считывать из текстовых полей.
5. С помощью подготовленного вами виртуального генератора и виртуального осциллографа проведите измерения характеристик «чёрного ящика», по которым попробуйте определить, какая из пяти показанных ниже схем собрана в «чёрном ящике».



## Приложение 1. Библиотека PXIDev.

### 1.a Функции для работы с модулем NI PXI-5402

#### ***Fgen\_Connect***

Назначение:

Производит подключение к модулю, установленному в определенную позицию в PXI-крейте.

**Синтаксис:**

*ViBoolean Fgen\_Connect(int FgenPos);*

**Аргументы функции:**

Имя	Тип	Описание
<i>FgenPos</i>	<i>int</i>	Число, указывающее на позицию модуля в крейте. Может принимать значения – 2,6,7,8

**Возвращаемое значение:**

**VI\_TRUE** - в случае успешного соединения с модулем.

**VI\_FALSE** – В данной позиции модуль не найден.

### ***Fgen\_Disconnect***

**Назначение:**

Производит отключение от модуля и освобождает все занятые им системные ресурсы.

**Синтаксис:**

*void Fgen\_Disconnect(void);*

**Аргументы функции:**

*void*

**Возвращаемое значение:**

*void*

### ***Fgen\_StartStandartWaveForm***

**Назначение:**

Включает модуль в режим генерации импульсов стандартной формы с заданной амплитудой и частотой. При вызове функции также указывается тип импульсов.

**Синтаксис:**

*ViBoolean Fgen\_StartStandartWaveForm(float Amplitude,float Frequency, ViInt32 WaveFormType);*

**Аргументы функции:**

Имя	Тип	Описание
<i>Amplitude</i>	<i>float</i>	Амплитуда генерируемых импульсов (в вольтах)

<i>Frequency</i>	<i>float</i>	Частота импульсов (в герцах)
<i>WaveFormType</i>	<i>Vilnt32</i>	Число задающее форму импульсов: 0 – синусоидальная 1 – прямоугольная 2 – треугольная 3 – Пилообразная( подъем напряжения) 4 - Пилообразная( спад напряжения) 5 – постоянное напряжение 6 – Белый шум.

**Возвращаемое значение:**

**VI\_TRUE** - в случае успешного выполнения.

**VI\_FALSE** – в случае ошибки.

### ***Fgen\_SetWaveFormAmplitude***

**Назначение:**

В режиме генерации производит изменение амплитуды генерируемых импульсов.

**Синтаксис:**

*void Fgen\_SetWaveFormAmplitude(float NewAmplitude);*

**Аргументы функции:**

Имя	Тип	Описание
<i>NewAmplitude</i>	<i>float</i>	Задаваемое значение амплитуды импульсов (в вольтах)

**Возвращаемое значение:**

*void*

### ***Fgen\_SetWaveFormFrequency***

**Назначение:**

В режиме генерации производит изменение частоты генерируемых импульсов.

**Синтаксис:**

*void Fgen\_SetWaveFormFrequency(float NewFrequency);*

**Аргументы функции:**

Имя	Тип	Описание
<i>NewFrequency</i>	<i>float</i>	Задаваемое значение частоты импульсов (в герцах)

**Возвращаемое значение:**

*void*

### ***Fgen\_GenerateFreqList***

**Назначение:**

Генерирует «частотный массив» и массив длительностей импульсов каждой частоты в заданном диапазоне частот и с заданными шагом и длительностью.

**Синтаксис:**

```
void Fgen_GenerateFreqList(int FreqSteps, float FreqMin, float FreqMax, float StepDuration, ViReal64 *pFreqList, ViReal64 *pTimeIntervallList);
```

**Аргументы функции:**

Имя	Тип	Описание
<i>FreqSteps</i>	<i>int</i>	Количество частотных интервалов в последовательности
<i>FreqMin</i>	<i>float</i>	Начальная частота импульсов в последовательности (в герцах)
<i>FreqMax</i>	<i>float</i>	Конечная частота импульсов в последовательности (в герцах)
<i>StepDuration</i>	<i>float</i>	Продолжительность одного частотного интервала (в секундах)
<i>pFreqList</i>	<i>ViReal64 *</i>	Указатель на массив, в который будет передана сгенерированная частотная последовательность
<i>pTimeIntervallList</i>	<i>ViReal64 *</i>	Указатель на массив, в который будет передана последовательность длительностей импульсов

**Возвращаемое значение:**

*void*

### ***Fgen\_StartFreqListGenerate***

**Назначение:**

Запускает в режим генерации частотную последовательность импульсов в соответствии с переданными параметрами – массив частот, амплитуда, форма импульса.

**Синтаксис:**

```
void Fgen_StartFreqListGenerate(ViInt32 WaveFormType, ViReal64 Amplitude, int ListSize, ViReal64 *pFreqList, ViReal64 *pTimeIntervallList);
```

**Аргументы функции:**

Имя	Тип	Описание
<i>WaveFormType</i>	<i>ViInt32</i>	Форма генерируемых импульсов. В текущей версии поддерживаются только синусоидальные импульсы. Может принимать любое целое значение.
<i>Amplitude</i>	<i>ViReal64</i>	Амплитуда импульсов генерируемой последовательности (в вольтах)
<i>ListSize</i>	<i>int</i>	Количество частотных интервалов в последовательности (размер передаваемых массивов)
<i>pFreqList</i>	<i>ViReal64 *</i>	Указатель на массив, в котором содержится генерируемая последовательность частот
<i>pTimeIntervalList</i>	<i>ViReal64 *</i>	Указатель на массив, в котором содержатся значения временных интервалов, соответствующих разным значениям частот в генерируемой последовательности

**Возвращаемое значение:***void****Fgen\_Stop*****Назначение:**

Останавливает текущую генерацию импульсов и переводит модуль в режим ожидания.

**Синтаксис:**

```
void Fgen_Stop(void);
```

**Аргументы функции:***void***Возвращаемое значение:***void***1.6. Функции для работы с модулем NI PXI-5114*****Scope\_Connect*****Назначение:**

Производит подключение к модулю, установленному в определенную позицию в PXI-крейте.

**Синтаксис:**

*ViBoolean Scope\_Connect(int ScopePos);*

**Аргументы функции:**

Имя	Тип	Описание
<i>ScopePos</i>	<i>int</i>	Число, указывающее на позицию модуля в крейте. Может принимать значения – 2,6,7,8

**Возвращаемое значение:**

**VI\_TRUE** - в случае успешного соединения с модулем.

**VI\_FALSE** – В данной позиции модуль не найден.

**Scope\_Disconnect****Назначение:**

Производит отключение от модуля и освобождает все занятые им системные ресурсы.

**Синтаксис:**

*void Scope\_Disconnect(void);*

**Аргументы функции:**

*void*

**Возвращаемое значение:**

*void*

**Scope\_SetHorizontalParameters****Назначение:**

Устанавливает непосредственно в модуль временные параметры оцифровки сигнала – частоту и количество измерений.

**Синтаксис:**

*void Scope\_SetHorizontalParameters(int SampleRate,int RecordLength);*

**Аргументы функции:**

Имя	Тип	Описание
<i>SampleRate</i>	<i>int</i>	Задаваемая скорость оцифровки – количество измерений в секунду
<i>RecordLength</i>	<i>int</i>	Количество измерений, запоминаемых в память, по достижении которого весь массив измерений может быть прочитан.

**Возвращаемое значение:***void***Scope\_GetHorizontalParameters****Назначение:**

Считывает из модуля установленные в нем значения оцифровки сигнала - частоту и количество измерений.

**Синтаксис:**

```
void Scope_GetHorizontalParameters(int *pSampleRate,int *pRecordLength);
```

**Аргументы функции:**

Имя	Тип	Описание
<i>pSampleRate</i>	<i>Int *</i>	Указатель на переменную, в которую будет передано считанное значение частоты измерений
<i>pRecordLength</i>	<i>Int *</i>	Указатель на переменную, в которую будет передано считанное значение количества измерений, по достижении которого весь массив измерений может быть прочитан.

**Возвращаемое значение:***void***Scope\_StartMeasureWaveForm****Назначение:**

Запускает процесс оцифровки, ожидает его окончания, после чего считывает массив измеренных значений и передает в программу.

**Синтаксис:**

```
void Scope_StartMeasureWaveForm(float AcqTimeOut,int MeasArrSize,ViReal64 *pMeasArr,int *pActualSamples,ViReal64 *XStepValue);
```

**Аргументы функции:**

Имя	Тип	Описание
<i>AcqTimeOut</i>	<i>float</i>	Время ожидания (таймаут), которое функция будет ждать окончания измерений
<i>MeasArrSize</i>	<i>int</i>	Предполагаемое количество измерений (размер массива).
<i>pMeasArr</i>	<i>ViReal64 *</i>	Указатель на массив, в который будут переданы измерения.
<i>pActualSamples</i>	<i>Int *</i>	Указатель на переменную, в которую будет

		передано количество измерений, которое удалось прочесть
<i>XStepValue</i>	<i>ViReal64 *</i>	Указатель на переменную, в которую будет передано фактическое значение, соответствующее времени между соседними измерениями

**Возвращаемое значение:**

*void*

### **1.с. Некоторые типы переменных библиотеки VISA, используемые при вызовах функций библиотеки PXIDev**

Название типа	Предназначение, описание	Эквивалентные C-типы
ViSession	Идентификатор(хэндл) подключения к модулю	-----
ViChar	Символ	char
ViBoolean	Булева переменная. Принимает значения <i>VI_TRUE</i> и <i>VI_FALSE</i>	Bool
ViInt32	32-разрядное целое	int
ViReal64	Число с плавающей точкой двойной точности	double

## **Литература**

1. Котельников В.А. О пропускной способности эфира и проволоки в электросвязи. // Всесоюзный энергетический комитет. Материалы к I Всесоюзному съезду по вопросам технической реконструкции дела связи и развития слаботочной промышленности, 1933г. Переиздана в: УФН, т.176, №7, 2006, стр.762-769
2. National Instruments Signal Generators Help  
<http://www.ni.com/pdf/manuals/370524p.zip>
3. National Instruments High-Speed Digitizers Help  
<http://www.ni.com/pdf/manuals/370592r.zip>