

НАЦИОНАЛЬНЫЙ АВИАЦИОННЫЙ УНИВЕРСИТЕТ.

КАФЕДРА АЭРОКОСМИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ



Воронов С.И.

РУКОВОДСТВО РАЗРАБОТЧИКА

ПРОГРАММНОЙ КОНСОЛИ

ДЛЯ ТЕХНИЧЕСКОГО ПЕРСОНАЛА

Версия 3.5

КИЕВ 2019г.

ОГЛАВЛЕНИЕ

[Введение 3](#_Toc28608982)

[Модульная структура приложения 3](#_Toc28608983)

[Файловая структура приложения 4](#_Toc28608984)

[Главная управляющая форма (Имитатор канала передачи данных) 4](#_Toc28608985)

[Имитаторы датчиков 5](#_Toc28608986)

[Консоль 6](#_Toc28608987)

[Основные R, L ,S - индикаторы консоли 6](#_Toc28608988)

[Основные панели и осциллограф сигнала 7](#_Toc28608989)

[GRID - Панель 8](#_Toc28608990)

[ALARM-Панель 9](#_Toc28608991)

[Цветовая индикация значений 10](#_Toc28608992)

[Пороговый контроль 10](#_Toc28608993)

[Допусковый контроль 10](#_Toc28608994)

[Спектральные измерения 11](#_Toc28608995)

[Вкладка спектрометр 11](#_Toc28608996)

[Осциллографы сигнала и истории выбранной гармоники 14](#_Toc28608997)

[Настройка приложения на конкретную техническую систему 15](#_Toc28608998)

[ФАЗА 1 15](#_Toc28608999)

[ФАЗА 2 15](#_Toc28609000)

[ФАЗА 3 16](#_Toc28609001)

[ФАЗА 4 20](#_Toc28609002)

# Введение

Данный документ описывает приложение, которое предназначено для визуализации большого количества входных оцифрованных сигналов в удобном для конечного пользователя виде. Такое приложение получило наименование «Консоль технического персонала» и может использоваться в составе современных информационно – измерительных, а также управляющих технических систем как мобильного, так и стационарного типа. В последнее время, архитектура построения таких систем все чаще реализуется в виде клиент – серверных решений, где измерение состояния объекта, а также управление объектом возлагается на сервер. В случае если техническая система состоит из нескольких подобных объектов, то возникает необходимость во втором уровне управления. Множество непосредственных или интегральных параметров в такой системе требует для разрешения проблемных ситуаций (ситуаций не разрешимых алгоритмами управления второго уровня) реализации вариантного и эргономичного отображения информации для оператора системы. В этом случае «Консоль технического персонала» в режиме клиента, который легко подключается к серверам или переключается между серверами, является хорошей альтернативой центра управления с множеством стендов и приборов. Другой сферой применения консоли в режиме клиент сервер могут являться процедуры отладки, настройки и регламентных проверок различных объектов и системы в целом.

## Модульная структура приложения

Приложение «Консоль технического персонала» включает в себя **16** прикладных модулей, структура связей между которыми показана на следующем рисунке:

### SensorGroup03

### ConsoleMAIN

## Common01

Sensor03

### MAIN ( главная форма приложения )

## Common01

## Common01

ConsoleDATA

ConsoleDATA

RingScope03

## CustomScope03

LineScope03

## CustomScope03

SwitchScope03

## CustomScope03

SpectrScopes03

Oscillograph03

DTF03

DnGridEd03

DnAlarm03

ImgLoadSave03

Рис. 1. Модульная структура приложения консоли технического персонала

Главным модулем, который получает управление при запуске приложения, является модуль **MAIN** (функции и экранная форма этого модуля описана в последующем разделе).

Под его управлением функционируют две программные подсистемы:

* **SensorGroup03** – Главный модуль подсистемы сбора первичной оцифрованной информации с имитаторов датчиков. Данная подсистема выполняет вспомогательную роль и может использоваться для отладки консоли, а при реализации в ней модели поведения сервера, трансформировать консоль в учебный или тренировочный комплекс;
* **ConsoleMAIN** – Главный модуль подсистемы «Консоль технического персонала».

## Файловая структура приложения

uses

Forms,

Common01 in 'Common01.pas',

**MAIN**  in 'MAIN.pas' {**Form1**},

ConsoleDATA in 'CONSOLE\ConsoleDATA.pas',

**ConsoleMAIN** in 'CONSOLE\ConsoleMAIN.pas' {**ConsoleForm**},

Sensor03 in 'SENSORs\Sensor03.pas',

**SensorGroup03** in 'SENSORs\SensorGroup03.pas' {**SensorGroupForm**},

Oscillograph03 in 'SERVICEs\Oscillograph03.pas',

CustomScope03 in 'SERVICEs\CustomScope03.pas',

ImgLoadSave03 in 'SERVICEs\ImgLoadSave03.pas',

LineScope03 in 'SERVICEs\LineScope03.pas',

RingScope03 in 'SERVICEs\RingScope03.pas',

SwitchScope03 in 'SERVICEs\SwitchScope03.pas',

DnAlarm03 in 'SERVICEs\DnAlarm03.pas',

DnGridEd03 in 'SERVICEs\DnGridEd03.pas',

SpectrScopes03 in 'SERVICEs\SpectrScopes03.pas',

DTF03 in 'SERVICEs\DTF03.pas';

# Главная управляющая форма (Имитатор канала передачи данных)

Главная управляющая форма (**Form1**) размещается в **UNIT MAIN**

Эта форма имеет следующий внешний вид:

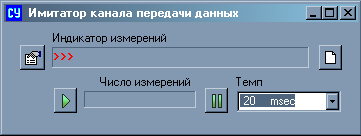


Рис. 2. Главная управляющая форма (**Form1**)

UNIT MAIN предназначен для имитации взаимодействия по схеме клиент-сервер между имитатором сервера (**UNIT** **SensorGroup03**) и подсистемой «Консоль технического персонала» (**UNIT ConsoleMAIN**).

**UNIT MAIN** задает **темп** обращений консоли технического персонала к имитатору сервера, выполняет чтение результатов измерений из имитатора сервера в структуру данных, определенную в **UNIT** **Common01**, после чего инициирует обработку и визуализацию этих данных косолью.

На главной управляющей форме расположены четыре кнопки:

 - Открыть панель имитаторов датчиков.

 - Открыть панель консоли технического персонала.

 - Начать измерения

 - Остановить измерения

## Имитаторы датчиков

**Кнопка** «Открыть панель имитаторов датчиков» открывает форму имитатора сервера (**UNIT** **SensorGroup03**). Эта форма (**SensorGroupForm**) имеет следующий внешний вид:

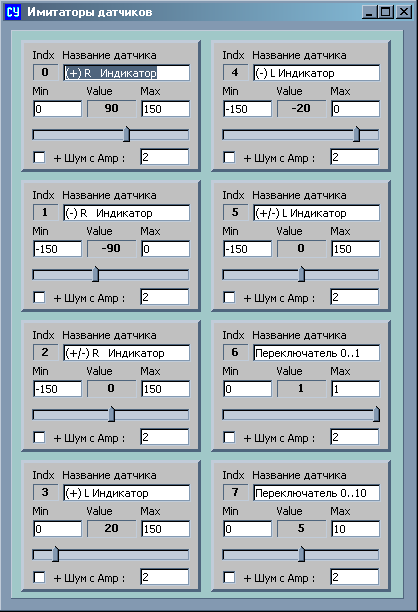


Рис. 3. Форма имитатора сервера (**SensorGroupForm**)

## Консоль

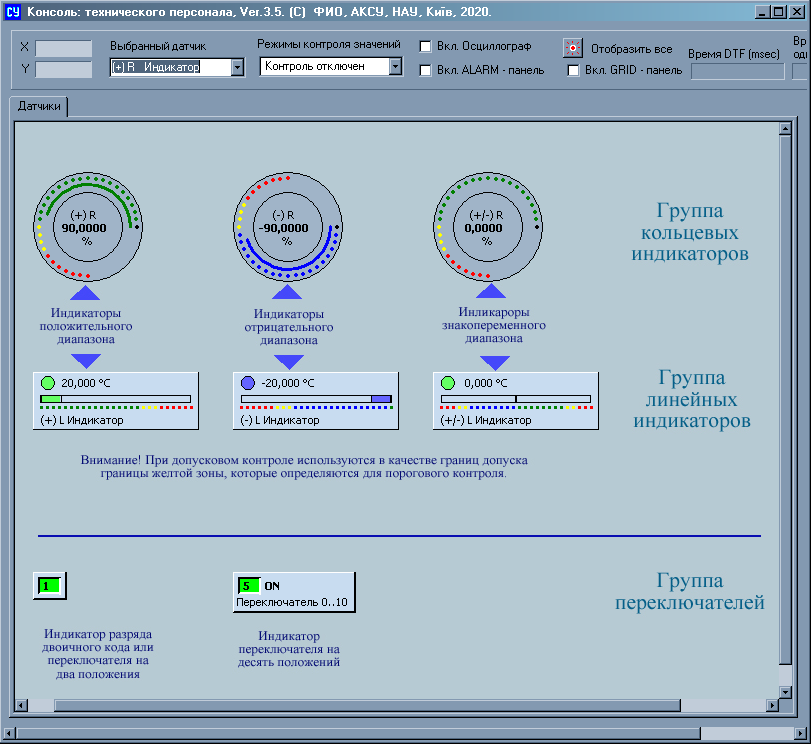
**Кнопка** «Открыть панель консоли технического персонала» открывает главную форму консоли (**UNIT** **ConsoleMAIN**). Эта форма (**ConsoleForm**) имеет следующий внешний вид:

Рис. 4. Консоль или форма имитатора клиента (**ConsoleForm**)

**Кнопки** «Начать измерения» и «Остановить измерения» соответственно начинают и останавливают передачу данных от имитатора сервера к консоли технического персонала.

## Основные R, L ,S - индикаторы консоли

На главной форме консоли отображаются три группы индикаторов (кольцевые, линейные и переключатели). В различных вариантах консоли можно использовать произвольное количество индикаторов из каждой группы.

|  |  |
| --- | --- |
| Все индикаторы располагают контекстным меню (кнопка свойств на мышке) следующего вида:  Рис. ХХ. Контекстное меню индикаторов |  |

Соответствующие стили отображения представлены на рисунках ниже.

|  |  |
| --- | --- |
|  | Рис. 5. Стили отображения кольцевых и линейных индикаторов |

Выборы позиций контекстного меню «Скрыть/Отобразить индикатор» и «Изменить прозрачность» определяют самоочевидные реакции и не требуют дополнительных комментариев.

## Основные панели и осциллограф сигнала

Основные панели консоли и осциллограф сигнала представлены на следующем рисунке:

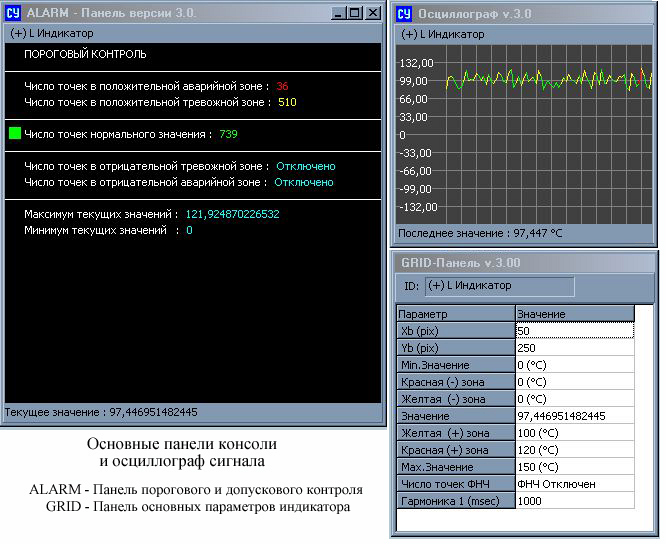


Рис. 6. Основные панели консоли и осциллограф сигнала

Все панели и осциллограф активируются либо деактивируются соответствующими чекбоксами (см. рис. «Консоль или форма имитатора клиента»). При активации панели или осциллографа они автоматически подключаются на сигнал выбранного индикатора. Выбор индикатора выполняется кнопкой действия мышки (обычно это левая кнопка) кликом на изображение индикатора или с помощью комбобокса «Выбранный датчик».

При выборе датчика, а следовательно и индикатора, все активированные панели подключаются на входной сигнал этого индикатора.

## GRID - Панель

GRID - Панель обслуживается инструментальным модулем **UNIT DnGridEd03** и предназначена для получения наиболее важных параметров выбранного индикатора.

Активация и деактивация GRID - Панели определение количества полей и разметка подписей к этим полям выполняется в **UNIT ConsoleMAIN** следующим методом:

// Включить/Выключить GRID - панель

**procedure TConsoleForm.chckGridOnOfClick(Sender: TObject);**

begin

if chckGridOnOf.Checked

then begin

if not Assigned(GRID01)

then begin

GRID01 := TDnGridEd.**Create**(260,260,False);

// Разметка подписей на панели

with GRID01 do

begin

**RowMaxInd** := 11;

//---------

RowTitle[1] := ' Xb (pix)';

RowTitle[2] := ' Yb (pix)';

//---------

RowTitle[3] := ' Min.Значение ';

RowTitle[4] := ' Красная (-) зона';

RowTitle[5] := ' Желтая (-) зона ';

//---------

RowTitle[6] := ' Значение ';

GRID01\_ValueRow := 6;

//---------

RowTitle[7] := ' Желтая (+) зона ';

RowTitle[8] := ' Красная (+) зона ';

RowTitle[9] := ' Max.Значение ';

//---------

RowTitle[10] := ' Число точек ФНЧ ';

RowTitle[11] := ' Гармоника 1 (msec) ';

end;

// Установка параметров в GRID – Патель

// по выбранному индикатору

**SetGRIDParam** (GRID01, SelectInd);

end;

end

else begin

if Assigned(GRID01)

then begin

GRID01.**Free**;

GRID01 := nil;

end;

end;

end;

Метод **SetGRIDParam** выполняет переключение параметрических значений на выбранный индикаитор.

## ALARM-Панель

ALARM - Панель обслуживается инструментальным модулем **UNIT DnAlarm03** и предназначена для ведения статистики по пороговому или допусковому контролю входного сигнала выбранного индикатора.

Активация и деактивация ALARM - Панели выполняется в **UNIT ConsoleMAIN** следующим методом:

// Включить/Выключить ALARM - панель сигнала

**procedure TConsoleForm.chckAlarmOnOfClick(Sender: TObject);**

begin

if chckAlarmOnOf.Checked

then begin

if not Assigned(ALARM01)

then begin

// Создать ALARM - панель

ALARM01 := TDnAlarm.**Create**(AlarmWidth, AlarmHeight);

// Установка параметров в ALARM – панель

// по выбранному индикатору

**SetAlarmParam** (ALARM01, SelectInd);

end;

end

else begin

if Assigned(ALARM01)

then begin

ALARM01.**Free**;

ALARM01 := nil;

end;

end;

end;

Метод **SetAlarmParam** выполняет переключение параметрических значений на выбранный индикаитор.

Для установки границ порогового и допускового контроля в объекте ALARM – Панель используются **свойства** класса **TDnAlarm = class(TObject)** этого объекта, а именно:

// Конечная (красная) граница в положительном диапазоне

property **P2Wall** : extended read fP2Wall write SetP2Wall;

// Начальная (желтая) граница в положительном диапазоне

property **P1Wall** : extended read fP1Wall write SetP1Wall;

// Начальная (желтая) граница в отрицательном диапазоне

property **N1Wal**l : extended read fN1Wall write SetN1Wall;

// Конечная (красная) граница в отрицательном диапазоне

property **N2Wall** : extended read fN2Wall write SetN2Wall;

// --------------------------

// Режимы использования границ для контроля значений

property **AlarmMode** : TAlarmMode read fAlarmMode write SetAlarmMode;

# Цветовая индикация значений

## Пороговый контроль

При пороговом контроле цветовая индикация значений определяется с помощью границ зон контроля и является общей как для круговых и линейных индикаторов так и для ALARM – Панели.

# Min

**P2Wall**

**P1Wall**

**N2Wall**

**N1Wall**

**0**

# Max

## Допусковый контроль

При допусковом контроле цветовая индикация значений определяется с помощью границ зон контроля и является общей как для круговых и линейных индикаторов так и для ALARM – Панели.

Если задан только **отрицательный поддиапазон**, то цветовая схема будет иметь следующий вид

# Min

**N2Wall**

**N1Wall**

**0**

Если задан только **положительный поддиапазон** то цветовая схема будет иметь следующий вид:

**P2Wall**

**P1Wall**

**0**

# Max

Если задан **отрицательный** и **положительный поддиапазоны**,то цветовая схема будет иметь следующий вид:

# Min

**P2Wall**

**P1Wall**

**N2Wall**

**N1Wall**

**0**

# Max

Подробнее о установке тревожных границ, а также концов положительного и отрицательного диапазонов будет описано в последующих разделах. В данном разделе важно отметить следующее:

* Для установки в индикатор только положительного поддиапазона необходимо отключить отрицательный поддиапазон, то есть, **Min =** **N2Wall =** **N1Wall = 0;**
* Для отключения в индикаторе порогового контроля в отрицательном поддиапазоне необходимо определить **Min < 0, N2Wall =** **N1Wall = 0;**
* Для установки в индикатор только отрицательного поддиапазона необходимо отключить положительный поддиапазон, то есть, **Max =** **P2Wall =** **P1Wall = 0;**
* Для отключения в индикаторе порогового контроля в положительном поддиапазоне необходимо определить **Max > 0, P2Wall =** **P1Wall = 0.**

# Спектральные измерения

Особенностью реализации приложения «Консоль технического персонала» является встроенная возможность выполнять спектральные измерения, что превращает консоль в измерительный инструмент для исследований или порогового контроля за произвольной гармоникой.

## Вкладка спектрометр

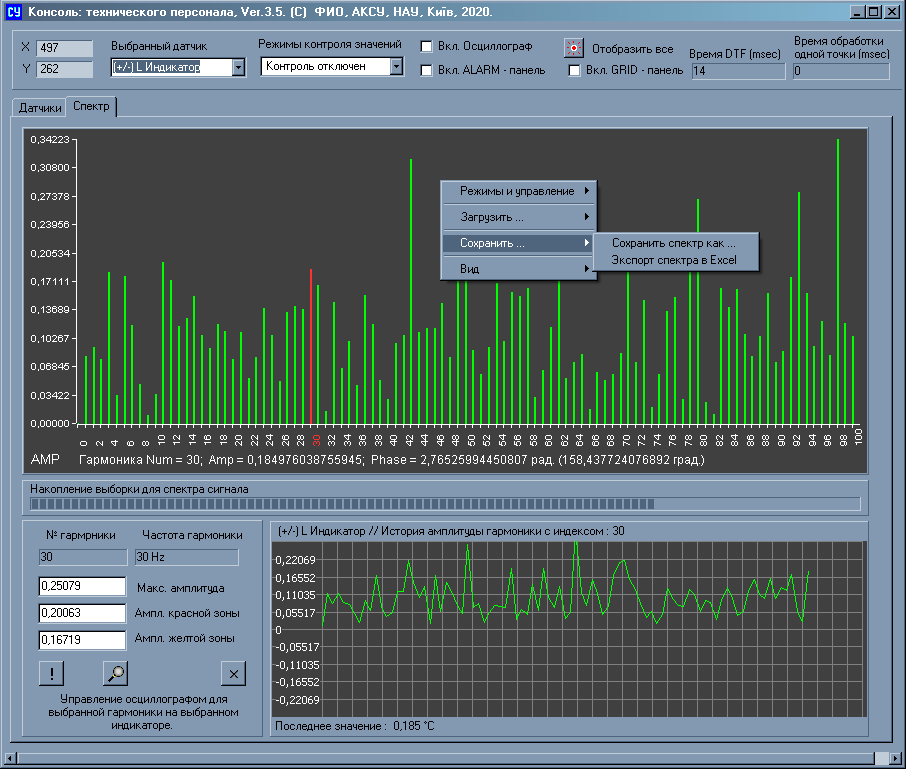
На рисунке ниже представлена вкладка «Спектр», которая включает спектральный анализатор и осциллограф истории выбранной гармоники.

Рис.7. Экранный образ вкладки «Спектр»

Анализатор спектра активируется, если в параметрах кругового или линейного индикатора установлен соответствующий запрос и оператор консоли выбрал для наблюдения такой индикатор. Выбор конкретной гармоники для наблюдения ее амплитуды выполняется мышкой с помощью кнопки действия по изображению этой гармоники в спектре гармоник анализатора.

Анализатор спектра реализуется модулем **UNIT SpectrScopes03,** который в свою очередь использует модуль **UNIT DTF03** дискретного преобразования Фурье

Для вычисления амплитуд и фаз для 100 гармоник анализатору требуется, как правило, менее 20 миллисекунд. Однако, для этой операции анализатору потребуется значительное число точек в выборке сигнала:

MaxSignal\_MaxPnt = 4 \* MaxGrmNum + 1;

Где:

MaxSignal\_MaxPnt – необходимое число точек;

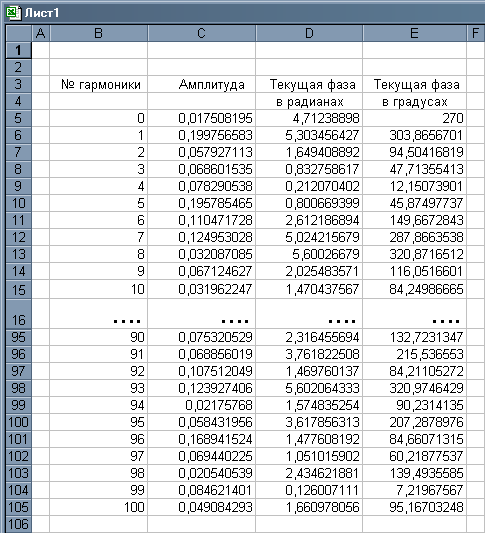
MaxGrmNum – номер старшей гармоники.

В случае 100 гармоник число точек выборки должно равняться 401 точке.

Внимание! При установке в индикатор необходимой длительности первой гармоники, может возникнуть необходимость в модель сервера включить процесс накопления выборки и ее буферизацию в течение периода такой первой гармоники.

Периодичность построения спектра будет прямо зависеть от темпа обращения консоли к имитатору сервера.

Вычисленный спектр можно экспортировать в Эксель при этом результат экспорта будет иметь вид:



Для управления осциллографом истории выбранной гармоники используется дополнительная панель, на которой можно указать параметры порогового контроля для выбранной гармоники. Параметры фиксируются кнопками:

 - Применить установленные значения амплитуд порогового контроля;

 - Подобрать автоматически значения амплитуд порогового контроля.

Кнопка «Применить» устанавливает в осциллограф текущие значения из полей «Макс. амплитуда», «Амплитуда красной зоны» и «Амплитуда желтой зоны»

Кнопка «Подобрать» инициирует поиск максимальной амплитуды гармоники в спектре и использует такую амплитуду для вычисления границ порогового контроля.

Индикация порогового контроля активируется в контекстном меню осциллографа истории выбранной гармоники. Там же можно установить вид осциллограммы (аналоговый или цифровой)

Дополнительным сервисом является кнопка очистки текущей осциллограммы без изменения индикатора или выбранной гармоники. Аналогичную операцию можно выполнить через контекстное меню осциллографа выбранной гармоники:

 - Стереть текущую осциллограмму истории выбранной гармоники.

Еще одним компонентом вкладки «Спектр» является индикатор прогресса по накоплению выборки. Скорость накопления выборки регулируется на главной форме приложения выбором темпа обращения консоли к имитатору сервера.

# Осциллографы сигнала и истории выбранной гармоники

Осциллограф истории выбранной гармоники и осциллограф сигнала реализуются общим модулем **UNIT Oscillograph03**. Отличие между объектами этих осциллографов заключаются в различии сигналов, к которым они подключены, а также способом создания таких объектов:

// Создать осциллограф на динамической форме

constructor **Create**(RqWidth, RqHeight : integer); **overload**;

// Создать осциллограф на арендуемой панели

constructor **Create**(RqPanel : TPanel); **overload**;

Так если осциллограф входного сигнала создается на динамической форме и выглядит как независимое окно Windows, то осциллограф истории выбранной гармоники размещает себя на предоставленной ему панели.

И в том и в другом случае функции контекстного меню осциллографов выполняются одинаково, только по отношению к разным источникам сигнала.

Одной из полезных функций контекстного меню являются «Режимы дополнительного отображения», которые включают в себя выбор одного из режимов отображения сигнала в нижней области панели осциллографа, а именно:

* Последнее значение
* Последнее приращение
* Текущее среднее значение
* Текущее максимальное значение
* Текущее минимальное значение
* Дисперсия по осциллограмме

Значения для последних четырех режимов вычисляются по всем точкам отображаемой в данный момент осциллограммы. Количество таких точек определяется в контекстном меню через «Число точек для осциллограммы».

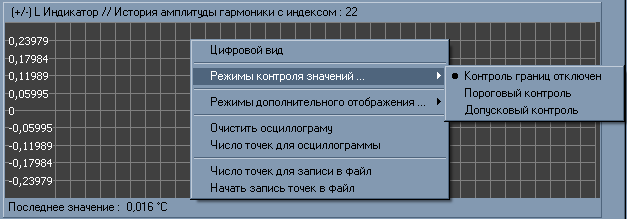
Экранный образ осциллографа истории выбранной гармоники представлен на следующем рисунке:

Рис.8. Экранный образ осциллографа истории выбранной гармоники

# Настройка приложения на конкретную техническую систему

**UNIT ConsoleDATA**

ПАРАМЕТРЫ И НАСТРОЙКИ ДЛЯ ГЛАВНОГО МОДУЛЯ

КОНСОЛИ ТЕХНИЧЕСКОГО ПЕРСОНАЛА

Для каждой конкретной реализации консоли в модуле UNIT ConsoleDATA необходимо определить базовое изображение (чертеж) консоли, параметры и размещение всех индикаторов, пиксельные габариты осциллографа сигнала и GRID – и ALARM - панели, номер последней гармоники в вычисляемом спектре.

## ФАЗА 1

Заголовок консоли настраивается с помощью определения следующих констант.

// -------------------------------------------

// Заголовок формы приложения

const AppType = ' Консоль';

AppName = ' Название технической системы,';

AppVers = ' Ver.3.5.';

AppAutor = ' Фамилия И.О.,'; // Разработчика консоли

AppCopyR = ' АКСУ, НАУ, Київ, 2020.'; // Копирайт

// -------------------------------------------

Далее необходимо выполнить компиляцию и убедиться в работоспособности приложения, то есть, выполнить верификацию исходных текстов шаблона приложения.

## ФАЗА 2

Разработать для консоли изображение одного из следующих видов:

* Стилизованное изображение технической системы;
* Конструкторское изображение технической системы
* Блок – функциональное изображение технической системы

Размер изображения должен учитывать высоту панели управления консолью (120 пикселей) и наиболее распространенный размер компьютерного экрана (1024х768).

Рекомендуется формировать изображение размером (900х600) пикселей. Разработанное изображение следует сохранить в формате \***.jpg** с именем **BaseImage.jpg** в папке проекта **PICTUREs**. Соответствующие подключению изображения константы имеют вид

// Имя файла базовой картинки (чертежа)

// (НАСТРАИВАЕТСЯ ДЛЯ КОНКРЕТНОГО ПРИЛОЖЕНИЯ!)

const ImgFileName = '\PICTUREs\BaseImage.jpg';

ImgWidth = 900; // Ширина картинки

ImgHeight = 600; // Высота картинки

// -------------------------------------------

Далее необходимо выполнить компиляцию и убедиться, что изображение технической системы корректно отображается и требует минимальной прокрутки (вертикальными и горизонтальными ползунками) в окне консоли. С точки зрения работоспособности приложения диапазоны прокрутки не являются принципиальными свойствами, однако для эргономики работы оператора технической системы с консолью это является важным.

## ФАЗА 3

Выполняя анализ технической системы, разработчик консоли должен определить количество, тип и параметры индикаторов, которые он будет использовать на базовом изображении.

Количество индикаторов задается как **общее количество-1** с помощью константы **Buffer00ArrMaxIndx**, которая определена в **UNIT Common01**. Например:

// Максимальный индекс в массиве данных буфера № 00

// Этот параметр определяет размер буфера данных для сервера и клиента

// ПАПРМЕТР НАСТРАИВАЕТСЯ ДЛЯ КАЖДОГО КОНКРЕТНОГО ПРИЛОЖЕНИЯ!

const Buffer00ArrMaxIndx = 7;

Далее вернемся в **UNIT ConsoleDATA.**

Каждый индикатор в этом юните описывается как отдельный элемент константного массива. Вначале рассмотрим структуру элемента массива, а потом приведем пример массива для 8-ми индикаторов.

Итак, структура элемента массива определяет тип и смысл каждого параметра одного индикатора:

// ------------------------

// Описатель массива индикаторов и параметров каждого индикатора

// (**ОПИСАТЕЛЬ НЕ ИЗМЕНЯТЬ!**)

//

type TDescScope = array[0..Buffer00ArrMaxIndx] of record

// ДЛЯ КАЖДОГО ИНДИКАТОРА:

SType : char; // Тип индикатора

// L-линейный R-кольцевой S-переключатель

// --------------------

Title : string; // Наименование датчика

Xb, YB : integer; // Пиксельная привязка индикатора на экране

Meas : string; // Размерность измеряемой величины

// ====================

// ПОРОГОВЫЙ КОНТРОЛЬ

// Положительный поддиапазон Max > P2R > P1Y > 0

// Если P2R = P1Y = 0, то пороговый контроль на

// поддиапазон откючен

Max : extended; // Конец положительного диапазона датчика

P2R : extended; // Начало положительной красной зоны

P1Y : extended; // Начало положительной желтой зоны

// --------------------

// Отрицательный поддиапазон Min < N1Y < N2R < 0

// Если N1Y = N2R = 0, то пороговый контроль на

// поддиапазон откючен

N1Y : extended; // Начало отрицательной желтой зоны

N2R : extended; // Начало отрицательной красной зоны

Min : extended; // Конец отрицательного диапазона датчика

// ====================

// ДОПУСКОВЫЙ КОНТРОЛЬ

// LB < LE во всем диапазоне

// Если LB = LE = 0, то допусковый контроль откючен

LB : extended; // В версии 3.5 всегда 0

LE : extended; // В версии 3.5 всегда 0

// --------------------

WS : TWallStyle; // Режим контроля значений

// Режим контроля значений задается

// именами следующих констант:

// wsNotUse Границы отключены

// wsDam Прорговый режим использования границ

// wsLimit Допусковый режим использования границ

// ====================

Compact : boolean; // Компактный вид отображения

//(только L, S -индикаторы)

Transp : boolean; // Прозрачность

// --------------------

NumLPF : byte; // Длина выборки для цифрового фильтра

Grm1P : extended; // Период первой гармоники в (msec)

// Отключить анализ спектра (-1 или 0)

// --------------------

Value : extended; // Стартовое значение на индикаторе

// --------------------

end;

Теперь приведем пример массива, который подлежит разработке для конкретной технической системы. В данном примере рассмотрена реализация массива для шаблонного проекта консоли:

// ---------------------------------------------------------

// 27.12.2019

// Массив статических параметров индикаторов

// Данные в этом массиве должны быть согласованы с данными

// аналогичного массива в SensorGroup

// (**НАСТРАИВАЕТСЯ ДЛЯ КОНКРЕТНОГО ПРИЛОЖЕНИЯ!**)

const DescScope : TDescScope =

(

// Индекс 0 ----------------------------------------------

( SType : 'R';

Title : '(+) R Индикатор';

Xb : 50; YB : 50;

Meas:'%';

// -----------

// Диапазон и пороговые области

Max : 150; P2R : 120; P1Y : 100;

N1Y : 0; N2R : 0; Min : 0; // Зоны отключены

// -----------

// Допусковая область

LB : 0; LE : 0;

// -----------

WS : wsDam; // Пороговый контроль

// -----------

Compact : True; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : -1; // Анализ спектра отключен

// -----------

Value : 90),

// Индекс 1 ----------------------------------------------

( SType : 'R';

Title : '(-) R Индикатор';

Xb : 250; YB : 50;

Meas:'%';

// -----------

// Диапазон и пороговые области

Max : 0; P2R : 0; P1Y : 0; // Зоны отключены

N1Y : -100; N2R : -120; Min : -150;

// -----------

// Допусковая область

LB : 0; LE : 0;

// -----------

WS : wsDam; // Пороговый контроль

// -----------

Compact : True; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : -1; // Анализ спектра отключен

// -----------

Value : -90),

// Индекс 2 ----------------------------------------------

( SType : 'R';

Title : '(+/-) R Индикатор';

Xb : 450; YB : 50;

Meas:'%';

// -----------

// Диапазон и пороговые области

Max : 150; P2R : 120; P1Y : 100;

N1Y : -100; N2R : -120; Min : -150;

// -----------

// Допусковая область

LB : 0; LE : 0;

// -----------

WS : wsDam; // Пороговый контроль

// -----------

Compact : True; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : -1; // Анализ спектра отключен

// -----------

Value : 0),

// Индекс 3 ----------------------------------------------

( SType : 'L';

Title:'(+) L Индикатор';

Xb : 50; YB : 250;

Meas:'°C';

// -----------

// Диапазон и пороговые области

Max : 150; P2R : 120; P1Y : 100;

N1Y : 0; N2R : 0; Min : 0; // Зоны отключены

// -----------

// Допусковая область

LB : 0; LE : 0;

// -----------

WS : wsDam; // Пороговый контроль

// -----------

Compact : False; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : 1000; // 1-я гармоника 1000 ms

// -----------

Value : 20),

// Индекс 4 ----------------------------------------------

( SType : 'L';

Title:'(-) L Индикатор';

Xb : 250; YB : 250;

Meas:'°C';

// -----------

// Диапазон и пороговые области

Max : 0; P2R : 0; P1Y : 0; // Зоны отключены

N1Y : -100; N2R : -120; Min : -150;

// -----------

// Допусковая область

LB : 0; LE : 0;

// -----------

WS : wsDam; // Пороговый контроль

// -----------

Compact : False; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : 1000; // 1-я гармоника 1000 ms

// -----------

Value : -20),

// Индекс 5 ----------------------------------------------

( SType : 'L';

Title:'(+/-) L Индикатор';

Xb : 450; YB : 250;

Meas:'°C';

// -----------

// Диапазон и пороговые области

Max : 150; P2R : 120; P1Y : 100;

N1Y : -100; N2R : -120; Min : -150;

// -----------

// Допусковая область

LB : 0; LE : 0;

// -----------

WS : wsDam; // Пороговый контроль

// -----------

Compact : False; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : 1000; // 1-я гармоника 1000 ms

// -----------

Value : 0),

// Индекс 6 ----------------------------------------------

( SType : 'S';

Title:'Переключатель 0..1';

Xb : 50; YB : 450;

Meas:'on/off';

// -----------

// Диапазон и пороговые области

Max : 1; P2R : 0; P1Y : 0; // Зоны отключены

N1Y : 0; N2R : 0; Min : 0; // Зоны отключены

// -----------

// Допусковая область

LB : 0; LE : 0; // Отключено

// -----------

WS : wsNotUse; // Контроль отключен

// -----------

Compact : True; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : -1; // Анализ спектра отключен

// -----------

Value : 1),

// Индекс 7 ----------------------------------------------

( SType : 'S';

Title:'Переключатель 0..10';

Xb : 250; YB : 450;

Meas:'Switch';

// -----------

// Диапазон и пороговые области

Max : 10; P2R : 0; P1Y : 0; // Зоны отключены

N1Y : 0; N2R : 0; Min : 0; // Зоны отключены

// -----------

// Допусковая область

LB : 0; LE : 0; // Отключено

// -----------

WS : wsNotUse; // Контроль отключен

// -----------

Compact : False; Transp : False;

// -----------

NumLPF : 0; // Фильтр отключен

Grm1P : -1; // Анализ спектра отключен

// -----------

Value : 5)

// ----------------------------------------------

); // of DescScope

**ВНИМАНИЕ!** Выполнение действий фазы 3 не является законченным перечнем действий для безошибочной компиляции проекта. Это связано с тем, что еще не установлены параметры в модуль **UNIT SensorGroup03**.

## ФАЗА 4

Фаза 4 выполняется в **UNIT SensorGroup03** и предназначается для настройки количества датчиков, а также координаты размещения на форме имитатора. Количество датчиков должно соответствовать количеству индикаторов. В приведенном ниже примере в константе SensorsDesc указан индекс последнего из датчиков (количество датчиков – 1).

В место явного указания старшего индекса, вы можете воспользоваться именем константы **Buffer00ArrMaxIndx**, которая определена в **UNIT Common01**.

Однако, это не отменяет необходимости вручную настроить массив размещения имитаторов датчиков на несущей панели. Суть такой настройки обычно сводится к тому, что часть геометрических привязок датчиков переносятся в закомментированную область или из закомментированной области в область определений. В приведенном ниже примере количество датчиков уже согласовано с количеством индикаторов шаблона проекта консоли.

// Оступ колонок сенсоров от верхнего края в пикселях

const SensorsTop = 10;

// Описание размещения имитаторов датчиков на несущей панели

// Значения X и Y задаются в пикселях.

const SensorsDesc : array [0..7] of TPoint =

**// Область определений**

( // Первая колонка

( X : 10; Y : SensorsTop ),

( X : 10; Y : SensorsTop + 140),

( X : 10; Y : SensorsTop + 280),

( X : 10; Y : SensorsTop + 420),

// Вторая колонка

( X : 200; Y : SensorsTop; ),

( X : 200; Y : SensorsTop + 140),

( X : 200; Y : SensorsTop + 280),

( X : 200; Y : SensorsTop + 420)

**// Закомментированная область**

**(\***

// Третья колонка

( X : 390; Y : SensorsTop; ),

( X : 390; Y : SensorsTop + 140),

( X : 390; Y : SensorsTop + 280),

( X : 390; Y : SensorsTop + 420)

**\*)**

);

Итак, теперь все необходимые настройки выполнены и проект можно компилировать. Если в процессе выполнения настроек Вы не сделали ошибок, то результатом компиляции будет готовое приложение.

Версия 3.5.

(С) Кафедра АКСУ, НАУ, Киев, 2017..2020 г.

(С) Source code Воронов С.И.

Редакция от 27.12.2019