

Модуль спектрального анализа

Модуль предназначен для обработки данных акселерометров или других датчиков за определенные интервалы времени для построения спектра колебаний и нахождения в полуавтоматическом режиме основных форм колебаний и вычисления параметров их пиков. Различает одиночные и двойные пики.

Технические требования

Операционная система Ubuntu 20/22, Linux OS 64-битная. PHP версии 7.3, модуль работы с PostgreSQL и ClickHouse.

Требования к вычислительным ресурсам: 1 ядро, поддерживающее 64-битные вычисления, 500 Мбайт на жестком диске для временных данных.

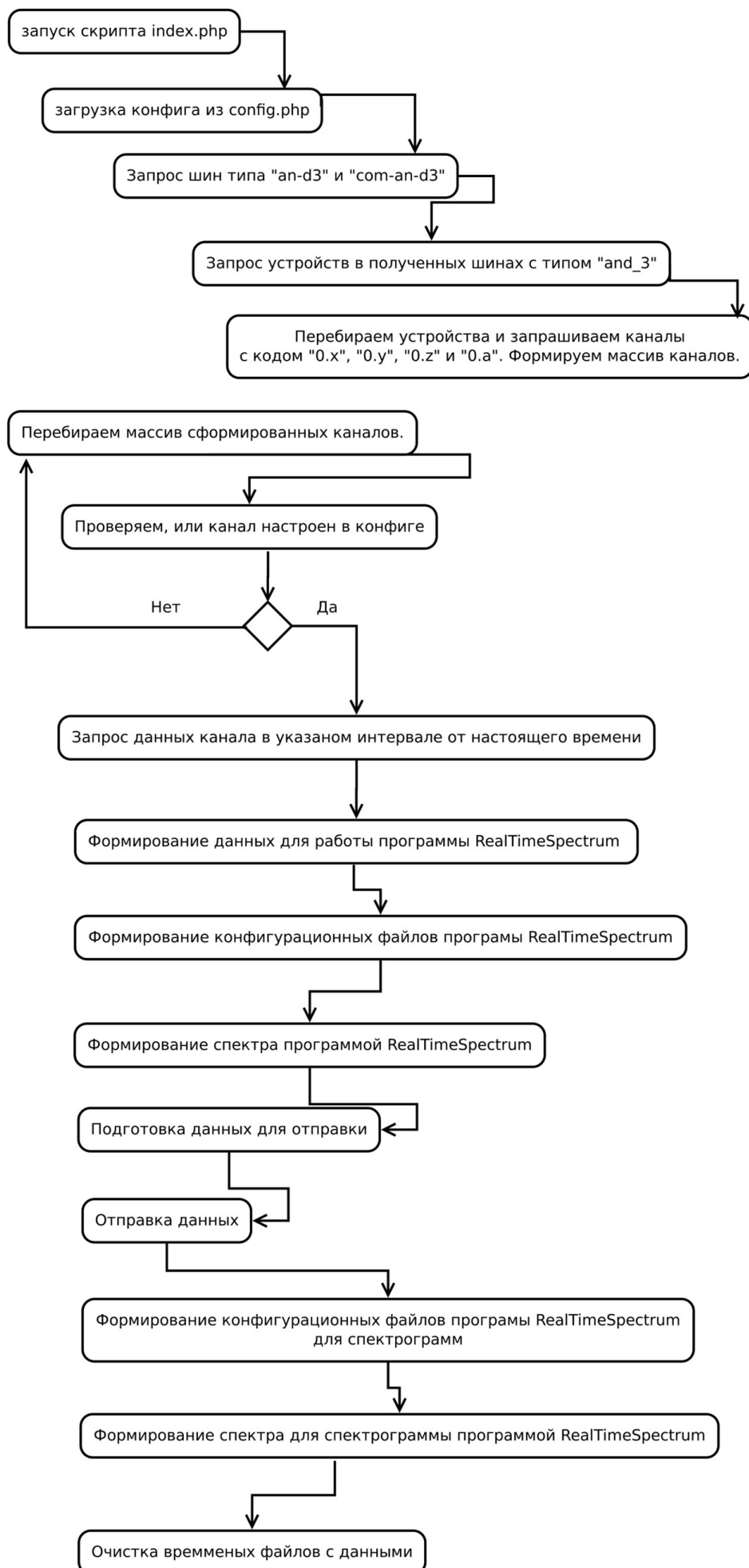
Описание работы

Структурно работу модуля можно поделить на две части:

- сбор, подготовка данных и отправка готовых данных, которую выполняет PHP-скрипт *index.php*;
- обработка данных, формирование спектров и поиск основных форм колебаний, которые выполняет программа *RealTimeSpectrum*.

Модуль может быть вызван или из системы, или из консоли в контейнере *docker «php»*. Для запуска работы модуля запускается PHP-скрипт *index.php*.

Блок-схема работы модуля:



Вначале работы скрипта *index.php* загружаются конфигурационные данные модуля из файла *config.php*, содержащего данные к подключению к базам данных, он находится на директории выше модуля и файла конфигурации модуля *config.php*, лежащего в директории модуля.

Далее выбираются шины типа «AN-D3» и «COM-AN-D3» (шины для подключения акселерометров). В выбранных шинах выбираются устройства типа «and_3». Далее просматриваются выбранные устройства и создается массив каналов этих устройств, у которых код равен «0.x» — ось X, «0.y» — ось Y, «0.z» — ось Z, «0.a» — модуль ускорения, обозначается через A.

После формирования массива обрабатываются оси. Проверяем, есть ли настройка для каждого канала в конфигурационном файле модуля *config.php*. Если данных настройки нет, то обработка данной оси пропускается.

Если канал сконфигурирован, то запрашиваются данные по этому каналу в интервале «\$time_interval» до времени, запрошенного в начале запуска скрипта (переменная «\$timestamp»).

Далее проверяются и создаются папки для выходных файлов программы *RealTimeSpectrum* «/var/www/html/public/Spectr».

Данные программой *RealTimeSpectrum* обрабатываются в двух вариантах. Первый — для полноценной спектральной мощности, данные записываются в директорию «/data», второй — для цветовой спектрограммы, данные записываются в директорию «/data_partial». Для спектрограммы спектральная мощность строится с меньшим частотным разрешением для увеличения быстродействия. Далее данные записываются в директории с именем равным id устройства, далее id канала, после год, и последняя директория – месяц.

Для примера формирования пути, если id устройства 67, id канала 433, 2022 год и месяц сентябрь 9-й месяц, тогда путь будет выглядеть для первого случая «/var/www/html/public/Spectr/data/67/433/2022/09», для второго случая «/var/www/html/public/Spectr/data_partial/67/433/2022/09».

Далее формируются конфигурационные файлы *ofs_options.ini* и *config.path* для работы программы *RealTimeSpectrum* и записываются полученные данные в бинарном виде для обработки. Далее данные обрабатываются программой *RealTimeSpectrum*, и полученные данные размещаются как указано выше, а полученные данные по пикам основных форм колебаний отправляются в базу данных. Обработка, формирование файлов и отправка данных выполняются для двух случаев подряд, так как для них используются одни и те же данные, чтобы результаты были одинаковыми.

Основные файлы модуля:

- *config.php* - файл настроек программы обработки спектров, должен быть в *gitignore*.
 - *config_example.php* - пример файла настроек программы обработки спектров.
-

- *config.path* - динамический файл настроек датчиков для работы программы обработки спектров *RealTimeSpectrus*, формируется в процессе работы модуля, должен быть в *gitignore*.
- *ofs_options.ini* - динамический файл общих настроек работы программы обработки спектров *RealTimeSpectrus*, формируется в процессе работы модуля, должен быть в *gitignore*.
- *.gitignore* - содержит список файлов и директорий, которые должны игнорироваться и не попадать в индекс git.
- *RealTimeSpectrum* - программа для обработки данных датчиков и формирования спектров и поиска пиков основных форм колебания.
- *index.php* - PHP-скрипт, вызываемый для обработки спектров. Непосредственно в нем берутся данные из базы данных, подготавливается для обработки и вызывается программа *RealTimeSpectrum*. Скрипт формирует пути, куда записывается спектр, сформированный *RealTimeSpectrum*, и отправляет результаты определения частот и декрементов колебаний в базу данных.
- *data_processing.sh* - bash-скрипт для запуска модуля через cron в контейнере.
- *source* - директория с временными рабочими файлами модуля.
- *index_check.php* - вспомогательный PHP-скрипт, позволяет проверить сколько в системе установлено устройств типа АН-ДЗ.
- *data_processing_test.sh* - вспомогательный bash-скрипт для запуска *index_check.php* в контейнере.

RealTimeSpectrum

Программа *RealTimeSpectrum* обрабатывает данные, сформированные акселерометрами, и строит спектры колебания.

Физический принцип, на котором основана работа программы, состоит в получении исходных данных ускорений с акселерометров от сборщика данных. Далее с помощью преобразования Фурье формируется спектральная мощность колебаний.

Метод определения основных форм колебаний и их декрементов является полуавтоматическим. Для его работы необходимо задать интервал частот, в котором необходимо искать пик колебаний, а также указать, ищется ли одинарный или двойной пик. Далее с помощью метода наименьших квадратов аппроксимируется пик или два пика в зависимости от настроек, и по данным аппроксимации вычисляются частота максимума пика и по ширине на полувысоте аппроксимированного пика - декремент колебаний.

Настройка модуля

Настройка модуля производится в файле *config.php*, все параметры подробно описаны в *config_example.php*, его содержание приведено ниже.

```
$api_token = 'A8WQ0N4zYPQvxbBMvPVz2ZnhGGyfgjhbh' - API-токен для доступа по REST API;
```

`$time_interval =40960` - интервал в секундах от текущего времени, за который запрашиваются данные с *clickhouse*;

`$length_zap_full=8192` - длина одной реализации для построения одного спектра;

`$length_zap_spectrograms=1024` - длина одной реализации для построения цветовой спектрограммы (для повышения производительности длина уменьшена для быстрого рендера картинки);

`$number_records_full=2` - количество реализаций, по которым происходит усреднение для построения одного спектра;

`$number_records_spectrograms=16` - количество реализаций, по которым происходит усреднение для построения цветовой спектрограммы;

`$basic_waveforms=5` - количество основных форм колебания, которые обрабатываются PHP-скриптом и отправляются в базу данных. Если данных на *i*-ю форму нет после работы программы построения спектра, то она заполняется нулями.

Настройка каналов для обработки

В массив `$device_code` добавляется код устройства, которому принадлежит канал.

В `$min_frequency_all` и в `$max_frequency_all` с ключом, равным коду устройства, указывается минимальная и максимальная граница интервалов, в которых нужно искать пики основных форм колебания. В `$double_frequency_all` указывается количество пиков для поиска.

```
$device_code=array('433', '434', '501', '502');  
  
$min_frequency_all['433']='0.3 0.6 1.3';  
$max_frequency_all['433']='0.35 0.8 1.6';  
$double_frequency_all['433']='1 1 1';  
  
$min_frequency_all['434']='0.3 0.6 1.3';  
$max_frequency_all['434']='0.35 0.8 1.6';  
$double_frequency_all['434']='1 1 1';  
  
$min_frequency_all['501']='0.3 0.6 1.3';  
$max_frequency_all['501']='0.35 0.8 1.6';  
$double_frequency_all['501']='1 1 1';
```

```
$min_frequency_all['502']='0.3 0.6 1.3';  
$max_frequency_all['502']='0.35 0.8 1.6';  
$double_frequency_all['502']='1 1 1';  
$device_code=array('433');
```

`$device_code` - набор кодов устройств, где устройство означает одну ось датчика, например 001X. Т.е. устройство с кодом 433 - это датчик с номером 001, ось X.

Для устройства 433 указываем интервалы, в которых нужно искать пики колебаний.

Интервалы:

```
$min_frequency_all['433']='0.3 0.6 1.3';  
$max_frequency_all['433']='0.35 0.8 1.6';
```

Они означают 3 интервала, в которых искать пики. Интервал 1 0.3-0.35; интервал 2 0.6-0.8; интервал 3 1.3-1.6; И количество пиков, которые необходимо искать `$double_frequency_all['433']='1 1 1'`; Здесь указаны три интервала, в них искать по одному пику. Варианты значений 1 либо 2.

Файл для работы программы обработки спектров, общие настройки.

`$file_options='number_records = '.$number_records.'` - количество реализаций, по которым будет строиться аппроксимация спектра и находятся параметры колебаний сооружения;

`length_zap = '.$length_zap.'` - длина одной реализации в количестве отсчетов, должно быть кратно 2^n , 2 в степени n, где n целое число;

number of records = 30 - сколько в одном файле реализаций;

sampling time = 0.1 - время дискретизации датчиков в секундах;

time = «22_01_10_23_59_43_021» - время, с которого считывать данные, данные до этой временной ометки будут игнорироваться, если указать 00_00_00_00_00_00_00, то будут обрабатываться все данные;

time_start = «2022_02_12_07_21_00_729» - время, с которого считывать данные, данные до этой временной ометки будут игнорироваться, если указать 00 00 00 00 00 000, то будут обрабатываться все данные;

```
time_end = «'.»$year«.»ckgedit>.«$month«.»ckgedit>.»$day«.»ckgedit>.  
«$hour«.»ckgedit>.»$minute«.»ckgedit>.«$second».'_000« big_files = 0 -  
параметр, отвечающий за то, что мы берем большие файлы данных за сутки. 1 - мы
```

берем, другое значение, не берем;

debug_mode = 0 - режим отладки, выводятся промежуточные данные обработки, по умолчанию отключено;

processe_all_files = 0 - обработка всех файлов сразу, по умолчанию обрабатывается начиная с самого последнего;

rusian = 1 - включение русского языка, по умолчанию включен;

file_size_for_processing = 1 - устанавливаем ограничение обработки файлов за один запуск;

time_shift=0.2457333333333333 - сдвиг по времени в секундах. Пропуск времени между соседними спектрами при обработке всех файлов сразу, величина в секундах;

detector_option_file = «Config.path» - имя файла с настройками для датчиков;

fast_processing = 0;

file_format=4 - формат файлов данных;

monitoring_horizont=1'.

Файл для работы программы обработки спектров, настройка датчиков

```
'$file_options_conf='{'  
  
'limitation_standart_deviation=0'' // среднеквадратичное отклонение;  
  
'limitation_ejection=50'' // выброс в процентах от диапазона;  
  
'measuring_range=3600'' // тип датчика (диапазон) AN3600, AN7200;  
  
'axis='.$detector_name.''' // номер датчика и его оси;  
  
'namefile="'.dirname(FILE).'/source/''' // относительный путь к  
данным;  
  
'savadata="'. $dir.'/''' // относительный путь, куда сохранять результаты  
моделирования;  
  
'datatemperature=".'/''' // относительный путь, откуда читать данные по  
температуре;  
  
'min_frequency ='.$min_frequency.''' // минимальная частота интервалов,  
в которых ищутся пики, в Гц, пример 0.4 0.8 2.3;
```

```
' 'max_frequency ='. $max_frequency.' ' ' // максимальная частота
интервалов, в которых ищутся пики, в Гц. Пример 0.6 1.1 2.5;

' 'double_frequency ='. $double_frequency.' ' ' // количество пиков в
интервале, допустимые значения 1 - один пик, 2 - два пика в интервале;

' 'signal_level=0' ';

}' ;
```

Настройка запуска по расписанию

Для запуска модуля из консоли нужно использовать скрипт *data_processing.sh*, который запускает модуль в контейнере *docker «php»*.

Для запуска модуля по расписанию настраивается запуск скрипта *data_processing.sh* через cron от имени root в требуемый интервал.

При новой установке необходимо проверить права на запуск как исполняемой программы *data_processing.sh RealTimeSpectrum*.

Содержание скрипта *data_processing.sh*:

```
«/opt/monitoring/dc exec -T php php
/var/www/html/SpectralAnalysis/Spectr/index.php» Данные означают: -
/opt/monitoring/dc - запуск docker;
```

- *exec* — команда выполнить; - *-T* - не создавать виртуальное tty-устройство для виртуальной консоли; - *php* — запуск контейнера *php*; - *php /var/www/html/SpectralAnalysis/Spectr/index.php* - запуск PHP-скрипта *index.php*, путь указан в среде контейнера; **Тестирование и юнит тесты** Для тестирования можно запускать из консоли скрипт «*data_processing_test.sh*», который запускает в контейнере PHP-скрипт «*index_check.php*».

Подключения модуля в системе мониторинга

Подключение модуля осуществляется автоматически при создании виджета класса *SpectrogramWidget* или *SpectrWidget* и выборе канала устройства типа акселерометр.