

ПРОГРАММА ДЛЯ ЭВМ
**«Управление зданиями и территорией
производственных площадок»**

Руководство администратора

Том 1

**Москва
2021**

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 2

Содержание

Том 1. Руководство администратора ПЭВМ «УЗИТ» (модули «Энергетика», «РЗА»)

Установка системы	1
Окружение для установки.....	2
Установка на CentOS7	7
Краткая инструкция по установке.....	15
Создание погодных аккаунтов	16
DarkSky	16
Обновление системы	19
Обновление ПЭВМ «УЗИТ».....	19
Процедура обновления системы	20
Технические детали устройства системы.....	22
Структура каталогов системы ПЭВМ «УЗИТ»	22
Сервисное API.....	23
Подсистема самодиагностики	24
Настройка модуля service-api	25
Crontab и его генерация.....	26
Сбор и хранение данных	27
Архитектура системы сбора данных.....	27
Общий принцип хранения данных в системе	30
Конфигурация хранилища данных	31
Хранилище данных MySQL (MariaDB).....	31
Команды для работы с хранилищем и его обслуживания	34
Консольные команды и полезные утилиты.....	36
Экспорт и импорт сущностей в системе.....	36
Общие принципы	36
Экспорт сложных структур с привязанными файлами	36
Базовые объекты и их поиск при экспорте и импорте	36
Список базовых объектов (экспортируемых каждый по отдельности).....	37
Список вложенных объектов.....	37
Список моделей, использующих в качестве идентификатора не GUID	37
Предустановленные объекты.....	38
Полезные утилиты в системе.....	38
Информация о различных консольных командах системы и вспомогательных утилитах	38
Показатели системы для мониторинга	41
Протоколы обмена.....	43
Протокол 1СИМ (модемный)	43
Протокол 1СИМ (модификация для LAN-устройств)	49
Источники.....	50

Том 2. Руководство администратора ПЭВМ «УЗИТ» (модуль «Транспорт»)

1. Введение.....	1
1.1. Перечень сокращений и обозначений.....	1
1.2. Рекомендуемые параметры рабочей станции	2
1.3. Меню системы.....	2
1.4. Справка	3
1.5. Управление данными в таблице	3

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 3

1.6.	Управление древовидным компонентом	5
1.7.	Всплывающее окно календаря	6
2.	Пользователи и роли	7
2.1.	Управление пользователями	7
2.2.	Управление паролями	9
2.3.	Тарифы доступа	11
2.4.	Управление ролями	17
2.5.	Сессии	19
3.	Структура подразделений	20
4.	Журнал событий	24
5.	Настройки диспетчера	26
6.	Управление системой	27
6.1.	Управление системными заданиями	27
6.2.	Просмотр логов	28
6.3.	Управление архивом	29
6.4.	Запуск скриптов	30
6.5.	Менеджер отчетов	31
6.6.	Информация о JVM	32
6.7.	Настройки рассылки	33
7.	Экспорт и импорт данных	35
7.1.	Экспорт данных	35
7.2.	Импорт данных	35
7.3.	Импорт ТС	36
7.4.	Импорт договоров из Excel	38
7.5.	Загрузка лога отката	39
7.6.	Загрузка/выгрузка файлов	39
7.7.	Просмотр потоков импорта	40
8.	Интеграция	41
8.1.	API TIS	41
8.2.	Процессинг ТК	42
8.3.	Телемедик	47
8.4.	Геокодер	48
8.5.	Отправка СМС	49
8.6.	Отправка в Яндекс.Транспорт	50
8.7.	Внешние системы	51
9.	Мобильные приложения и устройства	55
9.1.	Приложения	55
9.2.	Устройства	56
9.3.	Журнал	58
9.4.	Разрешения	59
9.5.	Подключение мобильных устройств	59
10.	Статистика, расчеты	66
11.	API картографии и маршрутизации	68
12.	Установка и настройка системы	71
12.1.	Установка БД	71
12.2.	Установка серверных компонентов	72

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 4

12.3.	Установка Caucho Resin.....	72
12.4.	Настройки Caucho Resin	73
12.5.	Настройки приложения	75
12.6.	Настройки подключения модулей приложения	77
12.7.	Настройки авторизации через IIS	78
12.8.	Срок действия паролей в Oracle	89
12.9.	Настройки интеграции с сервером телематики.....	89
12.10.	
	Интеграция с модулем доступа партнеров (МДП).....	90
12.11.	Настройка загрузки/выгрузки файлов.....	90
12.12.	Настройки паспорта ТС	91
12.13.	Настройки лицензии amCharts	91
13.	Пользовательское соглашение	92
14.	Подключение и настройка оборудования	94
15.	Кластерный режим работы	95
16.	Обращение в техническую поддержку.....	97
	Приложение А: Сервис импорта данных в БД приложения (Импортер).....	98
A.1.	Общие сведения.....	98
A.2.	Настройка сервиса	98
A.3.	Запуск сервиса	105

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 5

Установка системы

Окружение для установки

ПЭВМ «УЗИТ» работает под управлением ОС Linux и требует для своего функционирования наличие некоторых сторонних компонент. Здесь приведено краткое описание нужных компонент с версиями, а также возможные заменители.

Операционная система

В качестве основной ОС используется CentOS 7, все инструкции в руководстве в дальнейшем будут предполагать, что используется именно она.

Настоятельно рекомендуется для системы выделить отдельного пользователя в ОС и устанавливать компоненты системы в домашний каталог этого пользователя. Возможны конфликты с механизмом selinux, однако проблемы решаются настройкой php-fpm и прав доступа на каталоги. Настройка selinux в данное руководство не входит, в дальнейшем предполагается что selinux на сервере выключен.

СУБД

Система использует MySQL в качестве основной СУБД. Подходит любая версия старше 5.1. Рекомендуется использовать MariaDB версии не ниже 10.1. Если потребуется кластеризация, рекомендуется использовать MariaDB Galera Cluster. Настройка MariaDB в режиме кластера будет описана отдельно.

Для всех таблиц системы требуется использовать InnoDB engine. Системе достаточно одной базы данных и пользователя с полными правами на неё.

Web-сервер

В качестве web-сервера используется nginx в связке с php-fpm. Конкретная версия nginx значения не имеет. Возможно использование и Apache.

PHP

В настоящий момент система работает под управлением PHP5.6.

В PHP должны быть установлены модули mbstring, mysqlnd и gd. Также должен быть установлен php-fpm.

Дополнительные библиотеки

Для работы web-интерфейса системы должны быть установлены следующие библиотеки:

- wkhtmltox - для рендеринга коммерческих отчетов в pdf
- xorg-x11-fonts-cyrillic.noarch xorg-x11-fonts-misc.noarch xorg-x11-fonts-truetype.noarch xorg-x11-fonts-100dpi.noarch xorg-x11-fonts-75dpi.noarch fonts-ISO8859-2.noarch fonts-ISO8859-2-100dpi.noarch fonts-ISO8859-2-75dpi.noarch freefont.noarch urw-fonts - пакеты шрифтов для рендеринга pdf

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 6

Почтовый сервер

Для рассылки уведомлений система может использовать почтовый сервер как установленный на сервере с системой, так и любой внешний по SMTP. Следует организовать пользователя на почтовом сервере, от имени которого система будет рассылать уведомления.

Сетевое окружение

Для работы подсистемы сбора данных сервер должен быть доступен из сети Internet по следующим портам:

- 21001 - основной порт сбора (протокол 1СИМ)
- 31002 - для работы с сетевыми адаптерами "Термотроник"
- 31003 - порт для работы по протоколу iRZ Collector
- 31004 - для адаптеров БАРС-02
- 31005 - для модемов с поддержкой платформы RobustLink и ее протокола
- 31006 - для модемов Teleofis RX100-B1 (с автономным питанием)
- 31007 - для блоков "АКСОН"
- 2060 - для адаптеров "ВЗЛЁТ АССВ"
- 40000-49999 - для прозрачных ТСР-клиентов

Если какие-то типы коммуникационного оборудования применяться не будут, соответствующие порты на сервере открывать не обязательно.

Также нужны порты 80/443 для основного web-интерфейса и 81/444 для api (первый порт - обычный http, второй - https, если что-то из этого не используется, порт можно не открывать).

Системе требуется доступ в сеть Internet с сервера на котором она установлена (для получения данных с погодного сервиса и карт). Доступ должен быть через NAT, как минимум на стандартные http / https порты.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 7

Установка на CentOS7

Для быстрой установки из git-репозитория воспользуйтесь соответствующей инструкцией.

Дистрибутив состоит из двух каталогов. Первый atm содержит в себе web-интерфейс системы и прочие PHP-компоненты. Второй deploy содержит в себе бинарные компоненты системы (сервер сбора данных и сервер конвертации протоколов).

Предполагается, что для системы будет создан пользователь atmuser и сама система будет развёрнута в /home/atmuser.

Настройка дополнительных репозиториев

Для установки MariaDB потребуется файл mariadb.repo с таким содержанием (информация о репозитории).

```
# MariaDB 10.1 CentOS repository list - created 2016-11-14 08:41 UTC
# http://downloads.mariadb.org/mariadb/repositories/ [1]
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.1/centos7-amd64 [2]
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB [3]
gpgcheck=1
```

Его надо скопировать в

```
/etc/yum.repos.d
```

Также для установки PHP потребуется отдельный репозиторий - Remi repository, его домашняя страница по ссылке <http://rpms.famillecollet.com/> [4].

RPM этого репозитория по ссылке <http://rpms.remirepo.net/enterprise/remi-release-7.rpm> [5]

После чего установить его командой

```
yum install remi-release-7.rpm
```

Установка MariaDB

После конфигурации репозитория MariaDB устанавливается одной командой.

```
yum --enablerepo=mariadb -y install MariaDB-server MariaDB-client galera socat
```

Пакет galera можно не устанавливать, если не нужен кластер.

Конфигурация MariaDB может выглядеть так (/etc/my/cnf.d/server.cnf)

```
[mysqld]
general_log = 0
general_log_file = /var/log/mysql/mysql.log
port = 3306
bind-address = 1.2.3.4
skip-name-resolve
socket = /tmp/mysql.sock
tmpdir = /home/mysql/tmp
#innodb_force_recovery = 1
back_log = 1024
```

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 8

```
max_connections = 1024
max_connect_errors = 30
table_open_cache = 2048
max_allowed_packet = 16M
binlog_cache_size = 32M
max_heap_table_size =
512M read_buffer_size =
2M read_rnd_buffer_size =
16M sort_buffer_size =
256M join_buffer_size =
256M thread_cache_size =
256
thread_concurrency = 32
query_cache_type = ON
query_cache_size = 128M
query_cache_limit = 2M
query_cache_min_res_unit =
2k ft_min_word_len = 4
default-storage-engine = InnoDB
thread_stack = 192K
transaction_isolation = REPEATABLE-
READ tmp_table_size = 512M
slow_query_log
long_query_time = 10
key_buffer_size =
512M
bulk_insert_buffer_size = 512M
skip-innodb_doublewrite
innodb_file_per_table
innodb_log_group_home_dir=/home/mysql/iblog
s
innodb_data_file_path =
ibdata1:2G:autoextend
innodb_flush_method=O_DIRECT
innodb-support-xa = 0
innodb_buffer_pool_size =
16G innodb_read_io_threads =
4
innodb_write_io_threads = 4
innodb_thread_concurrency = 8
innodb_flush_log_at_trx_commit =
0 innodb_log_buffer_size = 16M
innodb_log_file_size = 1G
innodb_log_files_in_group = 2
innodb_max_dirty_pages_pct = 75
```


Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 9

```
innodb_lock_wait_timeout = 120
max_sp_recursion_depth = 255
```

```
datadir=/home/mysql
# Recommended in standard MySQL setup
sql_mode=NO_ENGINE_SUBSTITUTION
log-error=/var/log/mysql/mysql.log
[mysqld_safe]
log-error=/var/log/mysql/mysql.log
pid-file=/var/run/mysql/mysql.pid
open-files-limit = 32768
[client]
socket          = /tmp/mysql.sock
```

Внимание на bind-address. Если СУБД и сама система расположены на одном сервере, рекомендуется использовать 127.0.0.1, в противном случае требуется указать корректный IP-адрес по которому основная система будет "видеть" СУБД.

Конфигурация предназначена под использование только InnoDB и на выделенный под MySQL сервер. Если используется общий сервер, рекомендуется уменьшить количество памяти, выделяемое MySQL.

Установка дополнительных пакетов

Устанавливаем пакеты по списку.

```
yum install xorg-x11-fonts-cyrillic.noarch xorg-x11-fonts-misc.noarch
xorg-x11-fonts-truetype.noarch xorg-x11-fonts-100dpi.noarch
xorg-x11-fonts-75dpi.noarch fonts-ISO8859-2.noarch
fonts-ISO8859-2-100dpi.noarch fonts-ISO8859-2-75dpi.noarch freefont.noarch
urw-fonts
```

Библиотека wkhtmltox может быть скачана с официального сайта. Рекомендуется качать версию 0.12.2 <http://download.gna.org/wkhtmltopdf/0.12/0.12.2/> [6] так как для неё есть собранный RPM для

CentOS. После этого его можно установить с помощью yum

```
yum install wkhtmltox-0.12.2_linux-centos7-amd64.rpm
```

Установка PHP

Поскольку репозиторий для него уже настроен, его можно поставить через yum. Следует обратить внимание, что php ставится в /opt/remi/php56 и его конфигурационные файлы будут лежать именно там.

```
yum install php56 php56-php-fpm nginx php56-php-mbstring php56-php-mysqld
php56-php-gd
```

В дальнейшем потребуется создать симлинк /usr/bin/php на /opt/remi/php56/root/usr/bin/php

Следует помнить, что сервис php-fpm называется php56-php-fpm.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 10

Установка Nginx

Nginx ставится из штатного репозитория CentOS.

Конфигурация окружения

Включаем все сервисы

```
systemctl enable mysql.service
systemctl enable php56-php-fpm.service
systemctl enable nginx.service
```

Настраиваем nginx

Для работы потребуется два виртуальных хоста: api и www.

Конфигурация первого хоста может выглядеть так:

```
server {
    listen 127.0.0.1:80;
    listen 127.0.0.1:443 ssl;
    keepalive_timeout 70;
    ssl_certificate
        /etc/letsencrypt/live/help.1sim.online/cert.pem;
    ssl_certificate_key /etc/letsencrypt/live/help.1sim.online/
    #enables all versions of TLS, but not SSLv2 or 3 which are weak and now
    deprecated.
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    #Disables all weak
    ciphers
    ssl_ciphers

    ssl_prefer_server_ciphers on;

    access_log /home/atmuser/logs/access.www.log
    main; error_log /home/atmuser/logs/error.www.log;
    server_name atm.local;

    set $root /home/atmuser/atm/www/html;

    location ~ /\.php$
    {
        root $root;
        try_files $uri $uri/
        =404; fastcgi_pass
        127.0.0.1:8888;
        fastcgi_read_timeout 180s;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 11

```

location / {
    root $root;
    index index.html index.htm index.php
    index.shtml; try_files      $uri $uri/
    /index.php?$args;
}
location ~ /\.ht {
    deny all;
}

include www_rewrites.conf;
}

```

Требуется обратить внимание на ip-адрес и порт nginx, а также прописать корректные пути к сертификату при использовании HTTPS.

Файл использует www_rewrites.conf такого содержания:

```

rewrite "^/script.js$" /inc/lib/speeder/mini.php?type=javascript last;
rewrite "^/style.css$" /inc/lib/speeder/mini.php?type=css last;

```

Второй виртуальный хост - api, его конфигурация выглядит следующим образом:

```

server {
    listen *:81;
    keepalive_timeout 70;
    large_client_header_buffers 8 16k;
    access_log /home/atmuser/logs/access.api.log
    main; error_log /home/atmuser/logs/error.api.log;
    server_name api.atm.local;
    set $root /home/atmuser/atm/api/html;
    location ~ /\.php$
    { root $root;
    try_files      $uri $uri/
    =404; fastcgi_pass
    127.0.0.1:8888;
    fastcgi_read_timeout 180s;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $root$fastcgi_script_name;
    include      fastcgi_params;
    }
    location /
    { root
    $root;
    index index.html index.htm index.php
    index.shtml; try_files $uri $uri/ =404;
    }
    location ~ /\.ht
    { deny all;
    }
}

```

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 12

```
include api_rewrites.conf;  
}
```

Он также включает в себя api_rewrites.conf, вот такой:

```
rewrite ^/ext/$ /ext/$arg_function ;  
rewrite ^/ext/get_objects /index.php?r=ext/get-objects last;  
rewrite ^/ext/get_codes /index.php?r=ext/get-codes last;  
rewrite ^/ext/get_alarms /index.php?r=ext/get-alarms last;  
rewrite ^/ext/get_data /index.php?r=ext/get-data last;  
rewrite ^/ext/get_reports_list /index.php?r=ext/get-reports-list last;  
rewrite ^/ext/put_report_request /index.php?r=ext/put-report-request last;  
rewrite ^/ext/get_report_result /index.php?r=ext/get-report-result last;  
rewrite ^/ext/put_codes /index.php?r=ext/put-codes last;  
rewrite ^/ext/put_data /index.php?r=ext/put-data last;  
rewrite ^/ext/get_archive_dates /index.php?r=ext/get-archive-dates last;  
rewrite ^/ext/put_archive_date /index.php?r=ext/put-archive-date last;  
rewrite ^/ext/([^/]+) /ext/index.php last;
```

Добавляем пользователя

```
useradd atmuser  
mkdir /home/atmuser/logs  
chown atmuser:atmuser /home/atmuser/logs  
  
mkdir /home/atmuser/sessions  
chown atmuser:atmuser /home/atmuser/sessions  
  
mkdir /home/atmuser/uploads  
chown atmuser:atmuser /home/atmuser/uploads
```

Настраиваем php-fpm

Конфигурация демона php-fpm может выглядеть вот так:

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 13

```
[www_pool]
listen = 127.0.0.1:8888
listen.backlog = -1
listen.allowed_clients = 127.0.0.1
user = atmuser
group = atmuser
pm = dynamic
pm.max_children = 16
pm.start_servers = 4
pm.min_spare_servers = 2
pm.max_spare_servers = 6
pm.max_requests = 500
request_terminate_timeout = 181s
request_slowlog_timeout = 0s
slowlog = /home/atmdev/logs/php-fpm_help.log.slow
catch_workers_output = yes
php_admin_flag[display_errors] = off
php_admin_value[error_log] = /home/atmuser/logs/php-error.log
php_admin_flag[log_errors] = on
php_admin_value[memory_limit] = 1G
php_admin_value[max_execution_time] = 90
php_admin_value[session.save_path] = /home/atmuser/sessions
php_admin_value[upload_tmp_dir] = /home/atmuser/uploads
php_admin_value[upload_max_filesize] = 32M
php_admin_value[post_max_size] = 32M
php_admin_value[date.timezone] = Europe/Moscow
php_admin_value[session.gc_maxlifetime] = 10800
php_admin_value[default_charset] = utf-8
php_admin_value[mbstring.func_overload] = 0
php_admin_value[user_ini.filename] = ""
```

Следует обратить внимание, что этот файл должен находиться в /opt/remi/php56/root/etc/php-fpm.d/www.conf. Также следует обратить внимание на пути.

Развёртывание и конфигурация самой системы

Предполагается что каталог atm дистрибутива расположен в /home/atmuser/atm (на него как на atm), каталог deploy - соответственно в /home/atmuser/deploy (на него следует ссылаться deploy).

Настройка config/common-local.php

Требуется скопировать atm/config/common-local.php.sample в atm/config/common-local.php. В этом файле расположены настройки доступа к почтовому серверу, которые и нужно указать. Для работы используется расширение yii swiftmailer, более подробно о котором можно узнать здесь <http://www.yiiframework.com/doc-2.0/ext-swiftmailer-index.html> [7].

Настройка config/local.php

Аналогично требуется скопировать local.php.sample в local.php. Расположенные там константы можно раскомментировать при наладке работы системы, при переводе в эксплуатацию - закомментировать

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 14

(включают debug-режим Yii).

Настройка shared/config-local.php

Копируем config-sample.php в config-local.php. В этом файле нужно настроить несколько параметров, отвечающих в основном за работу web-интерфейса системы.

- LOCAL_CONFIG_API - url по которому будет доступен web-хост api
- LOCAL_CONFIG_API_PROTOCOL - протокол по которому будет доступен этот хост (http/https);
- LOCAL_CONFIG_WWW, LOCAL_CONFIG_WWW_PROTOCOL - аналогичные настройки для хоста www
- LOCAL_CONFIG_DB_ - параметры доступа к базе
- LOCAL_CONFIG_WEATHER_ - ключи доступа к погодному сервису (без них не будут обновляться параметры "температура атмосферы" и "атмосферное давление")
- LOCAL_CONFIG_DEFAULT_THEME - тема оформления по умолчанию
- LOCAL_CONFIG_OWNER_TITLE, LOCAL_CONFIG_OWNER_SITE - данные для футера в web-интерфейсе
- LOCAL_CONFIG_SITE_TITLE - title для заголовка страниц
- LOCAL_CONFIG_REPORT_COPYRIGHT_ - параметры для вывода в отчетах

Настройка модуля сбора данных (eks_server)

Модуль eks_server принимает входящие соединения по протоколу IСИМ и реализует непосредственное общение с подключенными к системе устройствами. Пример конфигурационного файла расположен в deploy/eks_server/eks_server.cfg.sample (его можно скопировать в deploy/eks_server/eks_server.cfg).

Для настройки потребуется задать следующие параметры:

- DBLogin, DBPassword, DBHost, DB - параметры доступа к базе данных (здесь обязательно задавать IP-адрес, по доменному имени не работает)
- LogPath - путь по которому будут расположены основные логи модуля.
- FirmwaresPath - путь по которому модуль будет искать обновления прошивок для модемов.
- LogFacilityPath - путь к каталогу, в котором будут располагаться по-модемные и по-приборные логи (они отображаются пользователям в интерфейсе системы).

Параметры по умолчанию подходят для большинства случаев, обычно требуется задать только параметры доступа к БД.

Также следует проверить, что исполняемый файл eks_server имеет атрибут executable, а каталоги, указанные в его конфигурационном файле, существуют и доступны.

Настройка модуля конвертации протоколов (eks_fork)

Этот модуль служит для обмена данными с модемами (и другим коммуникационным оборудованием), не поддерживающим протокол Iсим. Он принимает входящие соединения, производит некоторый первичный обмен (например для идентификации модема), после чего устанавливает соединение с модулем eks_server и транслирует протокол IСИМ в другие протоколы по мере обмена.

Аналогично пример конфигурационного файла расположен в deploy/eks_fork/eks_fork.cfg.sample, его можно скопировать в deploy/eks_fork/eks_fork.cfg.

При первичной настройке потребуются параметры DBLogin, DBPassword, DBHost, DB, LogPath, LogFacilityPath. Их значение аналогично таковым у eks_server.

Настройка модуля обработки данных (data_processor)

Этот модуль служит для дополнительной обработки данных, поступающих с устройств (вычисление архивов

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 15

по интеграторам, интегрирование моментальных значений итп).

Пример его конфигурационного файла находится в `deploy/data_processor/data_processor.cfg.sample`, его можно скопировать в `deploy/data_processor/data_processor.cfg`.

Назначение основных параметров в конфигурационном файле аналогично `eks_server/eks_fork`.

Настройка автоматически выполняемых задач.

Система в работе полагается на периодический запуск некоторого набора фоновых задач. Для запуска используется стандартный механизм `cron`. Если он не установлен в системе (если она установлена из минимального образа) - достаточно установить его командой

```
yum install cron
```

После этого надо сформировать файл с фоновыми задачами. Для этого предлагается файл `atm/config/crontab.full`. Его требуется установить в качестве `cron`-файла для пользователя, от имени которого будет работать система (по умолчанию это `atmuser`).

Необходимо также увеличить лимит на оперативную память для скрипта в конфигурационном файле `/opt/remi/php56/root/etc/php.ini`. Одного гигабайта в данном случае достаточно.

Наполнение базы данных

С дистрибутивом также поставляется дамп базы данных, в котором в системе зарегистрирована ровно одна компания. Этот дамп нужно импортировать в MySQL.

Примечания

- [1] <http://downloads.mariadb.org/mariadb/repositories/>
- [2] <http://yum.mariadb.org/10.1/centos7-amd64>
- [3] <https://yum.mariadb.org/RPM-GPG-KEY-MariaDB>
- [4] <http://rpms.famillecollet.com/>
- [5] <http://rpms.remirepo.net/enterprise/remi-release-7.rpm>
- [6] <http://download.gna.org/wkhtmltopdf/0.12/0.12.2/>
- [7] <http://www.yiiframework.com/doc-2.0/ext-swiftmailer-index.html>

Краткая инструкция по установке

Здесь приведена краткая инструкция по развёртыванию системы из `git`-репозитория разработчиков или из предоставленных архивов с этими репозиториями.

Предполагается, что используемой системой является CentOS 7, для других систем нужно отдельно подбирать устанавливаемые пакеты и модифицировать скрипт установки согласно используемому пакетному менеджеру и системе инициализации.

Также, для упрощения предполагается, что система будет работать под пользователем `atmuser`, основным каталогом `web` будет `/home/atmuser/atm`, а каталогом бинарных компонентов системы - `/home/atmuser/deploy`.

Требуемые действия

Последовательность действий следующая:

1. Создать пользователя `atmuser`.
2. Поместить репозиторий [`ssh://git@gitlab.1sim.online:5122/1sim/deploy.git` `ssh://git@gitlab.1sim.online:5122/1sim/deploy.git`] в `/home/atmuser/deploy` (либо развернуть в этот каталог предлагаемый архив)

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 16

3. Поместить репозиторий [`ssh://git@gitlab.1sim.online:5122/1sim/atm.git`
`ssh://git@gitlab.1sim.online:5122/1sim/atm.git`] в `/home/atmuser/atm` (либо развернуть в этот каталог предлагаемый архив)
4. Предоставленный каталог `files` поместить в `/home/atmuser/files`. Он содержит изображения предустановленных элементов мнемосхем.
5. **От имени root** выполнить скрипт `/home/atmuser/deploy/install/install.sh`. Он установит в систему нужные пакеты, пропишет автозапуск `mysql`, `php-fpm` и `nginx`, а также создаст каталоги, требуемые для работы системы. Также скрипт откроет порты, требуемые системе для сбора данных (полный список можно посмотреть в `ports.sh`). Дополнительно, если требуется использование прозрачных tcp-клиентов, потребуется открыть порты `40000-50000 tcp`, скрипт по умолчанию их не открывает.
6. Создать в `mysql` базу (важно выставить кодировку `utf8` для базы в целом) и завести пользователя с доступом к ней. По умолчанию используется пользователь `atmuser` с паролем `atmpass`. Для создания пользователя и базы можно воспользоваться скриптом `deploy/install/db_user.sql`.
7. Загрузить в базу дампы `deploy/install/db.sql`. В нем содержится структура базы системы и требуемые таблицы-справочники.
8. После этого можно приступить к настройке конфигурационных файлов системы.

Возможные проблемы и методика их устранения

Не запускается nginx

Нужно проверить наличие каталога `/home/atmuser/logs`, а также доступность его для записи пользователю, от которого запускается `nginx`. Также если не выключен `selinux`, нужно прописать это разрешение и в нём.

При создании объектов / групп в интерфейсе выводятся знаки вопроса

Требуется выставить кодировку `utf8` для самой базы (помимо того, что она выставлена для отдельных таблиц).

Создание погодных аккаунтов

DarkSky

Заходим на [<https://darksky.net/dev> [1]]

Переходим на [<https://darksky.net/dev/register> [2]]

Заполняем 3 поля: email, пароль (минимум 8 символов), подтверждение пароля.

Предлагается использовать почту 1сим, например уже используется **weather_X@1sim.online**, где X занято от 1 до 12.

Форма вот такая посылается на сервере:

```
_csrf:11i0AaNV-0MS-kaaUyNZHYAaxEEGNJhXdcxg
email:v.basunkov@gmail.com
password:****
confirmation:****
```

На емейл приходит подтверждение, внутри ссылка. Перейти по ссылке нужно в течение за 30 минут.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 17

To complete the registration process,
just follow this link:

<https://darksky.net/dev/register/-2110892957/z...0wmQ3QmM0b211>

If you didn't register for a developer account with Dark Sky, simply ignore this email: no action will be taken.

This link will expire in 30 minutes.

Зайти на [<https://darksky.net/dev/login> [3]] ввести логин/пароль и окажемся на [<https://darksky.net/dev/account> [4]]

Далее видим **Your Secret Key: d21fc2f4ecd85d8009c10f412734c9c0**. Сохраняем его. Для успешной работы одной инсталляции 1 сим нужно **6!** погодных аккаунтов.

Необходимо создать файл **atm\modules\weather\config-local.php** на основе сэмпл файла рядом.

```
<?php
return [
    'params' => [
        'sendAlertEmail'    => false,
        'futureApiClass'    => 'darksky',
        'currentApiClass'   => 'darksky',
        'pastApiClass'      => 'darksky',
        'holesApiClass'     => 'darksky',
        'apiClasses'        => [
            'darksky'       =>
                [
                    'class'   => '\modules\weather\providers\Darksky',
                    'keys'    => [''''Сюда вписываем в кавычках наши 6 секретных ключей!'''],
                ]
        ],
        'enableForecast' => false,
        'currentDelay' => 1800, //30min, minimum seconds between two feeds
        'futureDelay' => 43200, //12hours, minimum seconds between two feeds
    ],
];
```

Параметр **enableForecast** можно сделать **true** в большинстве случаев. Сделать false если не хватает вычислительных мощностей сервера.

Пока еще доступен вариант использования ранее созданных wunderground ключей. В config файле все подстроки darksky нужно заменить на wunderground. Ключей, как и ранее, нужно 9 штук.

Не забываем проверить наличие записей в кроне на запуск погодных скриптов (после прописки ключей). Команда:

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 18

```
./yii crontab/build > ../crontab.new
crontab crontab.new

#####
# Module weather
#####
# Weather updater
* * * * * cd ~/atm; ./yii weather/api/update-current

# Weather from past
* * * * * cd ~/atm; ./yii weather/api/check-past

# Weather fill missing holes
* * * * * cd ~/atm; ./yii weather/api/check-holes

# Weather day weather
30 3 * * * cd ~/atm; ./yii weather/api/compute-day
```

Примечания

[1] <https://darksky.net/dev>

[2] <https://darksky.net/dev/register>

[3] <https://darksky.net/dev/login>

[4] <https://darksky.net/dev/account>

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 19

Обновление системы

Обновление ПЭВМ «УЗИТ»

Для обновления системы требуется выполнить несколько операций, перечисленных ниже. Следует сделать резервную копию базы данных и файлов системы, чтобы обеспечить возможность возврата к предыдущей версии.

Если система работает на нескольких серверах, настоятельно рекомендуется обновлять до актуальной версии все сервера (обновления структуры БД могут привести к неработоспособности старой версии системы с новой БД).

Обновление бинарных модулей

В данном случае имеются в виду модули `eks_server / eks_fork / data_processor`, расположенные по умолчанию внутри каталога `/home/atmuser/deploy`. Для их обновления нужно сделать следующее:

1. Убрать запуск модулей из `crontab` (чтобы они не перезапустились, пока идёт процесс обновления).
2. Остановить модули (можно просто `kill -9` соответствующий процесс).
3. Заменить модули обновлёнными версиями (в случае, если развернута копия `git`-репозитория, достаточно сделать `git pull` находясь на ветке `master`). Если `git pull` не срабатывает (выдает ошибку на локальные изменения), это скорее всего вследствие выставленного атрибута `eXecutable` на исполняемых файлах модулей. В таком случае достаточно сбросить изменения при помощи `git checkout -- <ИМЯ_ФАЙЛА>`.
4. Выставить атрибут `eXecutable` на исполнимых файлах модулей (`deploy/eks_server/eks_server` итп).
5. Вернуть модули в `crontab`.

Если после обновления модули не запускаются, обычно это бывает из-за добавления новых конфигурационных параметров, для проверки достаточно запустить соответствующий модуль вручную из консоли. Если каких-то параметров не хватает, он сообщит об этом в консоль. В таком случае рекомендуется посмотреть пример соответствующего параметра в соответствующем примере конфигурации (напр. `deploy/eks_server/eks_server.cfg.sample`) и создать соответствующий параметр в актуальном конфигурационном файле.

Для упрощения работы в случае развёртывания системы из `git`-репозитория, обновление бинарных модулей может быть выполнено при помощи утилиты обновления командой `python3 deploy/update/update.py --deploy_only`.

Обновление web-интерфейса

При обновлении интерфейса также обновляется БД системы. Рекомендуется сделать резервную копию перед этой операцией. Обновление выполняется следующей последовательностью операций.

1. Остановить все выполняющиеся фоновые задачи. Для этого в `runtime`-каталог системы требуется положить файл `crons-disabled` (по умолчанию `/home/atmuser/runtime/crons-disabled`). Также следует остановить все выполняющиеся `cron`-задачи. Для этого нужно посмотреть их `PID` в файлах `*.pid` в `runtime`-каталоге и поприбивать соответствующие процессы. Либо можно просто поискать процессы с `./yii` в названии и поприбивать их (`pkill -f ".yii"`).

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 20

2. Залить обновленный дистрибутив системы. При развёртывании из git-репозитория это просто команда `git pull`.
3. Если в `.sample` - файлах конфигурации появились новые параметры, нужно завести их в локальных файлах. Обычно этот шаг не требуется, но если после обновления появляются ошибки вида "Undefined index" при обращении к `Yii::$app->params['SOMETHING']` - имеет смысл проверить полноту конфигурации в первую очередь.
4. Обновить БД таблицы командой `./yii migrate/up`. Также если появились обновления БД в модулях, нужно применить и их. Это делается командой `./yii migrate/up --migrationPath=modules/MODULE_NAME/migrations`. Команду следует выполнять для каждого модуля в котором есть каталог `migrations`.
5. Обновить `crontab`. Новый `crontab` формируется командой `./yii crontab/build`. Вывод команды следует перенаправить в какой-нибудь файл. После чего этот файл можно отредактировать (если используются нестандартные пути или часть системы расположена на другом сервере) и установить в качестве `crontab`.

Процедура обновления системы

При обновлении системы требуется не только использовать свежую версию кода, но и аккуратно применить все изменения, внесённые в базу.

Регулярное обновление

Если система на сервере обновляется достаточно регулярно (хотя бы раз в 2 недели), достаточно сделать `git pull` основного репозитория и применить миграции из основных каталогов.

1. Изменения структуры базы

Хранятся в директории: **`/migrations/structure`**

Команда для развёртывания: **`yii migrate-structure`**

В этих миграциях нельзя использовать модели проекта, можно использовать чистые запросы в базу. Включают в себя изменения в структуре таблиц, полей.

Необходимо выполнять развёртывание в первую очередь.

2. Другие изменения

Хранятся в директории: **`/migrations`**

Команда для развёртывания: **`yii migrate`**

Эти миграции могут использовать модели проекта. Включают в себя обновления данных, разные настройки проекта.

Во избежание конфликтов при обновлении не рекомендуется добавлять команды изменения структуры таблиц базы.

В связи с переходом на систему меток для обновлений, все новые миграции добавляются в `/migrations`, а миграции структуры базы оставлены для обратной совместимости. Начиная с обновления `update13` (см. ниже) отдельных миграций структуры базы не предполагается.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 21

Обновление системы на standalone серверах (или что делать, когда обновляется очень старая версия)

Для того, чтобы решить проблему "код обновился и старые миграции сломались" в истории гита выставляются метки вида updateNN, где NN - какое-то число. При масштабном обновлении требуется переходить на эти метки по очереди и после каждого перехода накатывать миграции из пункта 1. После чего можно переходить на финальную версию (ветка master) и накатывать миграции в последний раз.

Также требуется накатывать миграции модулей (из каталогов modules/XXXXXX/migrations) - в некоторых модулях есть свои самостоятельные миграции (чтобы не захламлять те экземпляры системы, которым не нужны миграции того или иного модуля).

Утилита обновления

Для автоматизации процесса есть утилита обновления. Она находится в репозитории deploy, в update/update.py. Запуск осуществляется командой:

```
python3 ../deploy/update/update.py
```

Запуск требуется осуществлять из основного каталога репозитория **atm** (по умолчанию это /home/atmuser/atm). При запуске из другого каталога требуется поправить путь к update.py.

Для использования требуется python версии не менее 3.4 с модулями flock и psutil. Стандартная процедура установки устанавливает его из репозитория CentOS.

По умолчанию утилита берет последний примененный апдейт из файла last_update.txt. Также она сохраняет текущую ветку (на которой был репозиторий до её запуска). При необходимости можно использовать параметры:

- -b (--branch) - какую ветку выставить в конце обновления.
- -s (--start_from) - с какой метки updateNN начинать обновление.
- --deploy_only - обновлять только бинарные модули системы (/home/atmuser/deploy). Иногда полезно, если в бинарных модулях есть какое-то обновление, которое хотелось бы получить, а полноценную процедуру обновления системы проводить нежелательно.

При использовании утилиты в основном каталоге репозитория появится файл last_update.txt. Он нужен для того, чтобы при следующем запуске обновления не требовалось бы пытаться применить обновления, которые уже установлены в текущий экземпляр системы.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 22

Технические детали устройства системы

Структура каталогов системы ПЭВМ «УЗИТ»

Предполагаем, что всё развёрнуто в home-каталоге некоторого пользователя (по умолчанию atmuser), все каталоги далее считаются от него.

- atm - основная кодовая база системы
- files - разделяемые файлы (пока просто каталог, в кластерном режиме должен быть доступен со всех серверов кластера - использовать NFS или распределенные ФС)
 - web_files - файлы, загруженные пользователями (руководства к прибором, фотографии объектов итп)
 - server_logs - логи обмена модулей сбора с модемами и приборами
- deploy - бинарные части системы
- runtime - различные временные файлы используемые php-частью системы
- logs - логи вебсервера, php и крон-скриптов
 - access.api.log - лог запросов к api
 - error.api.log - лог ошибок api
 - access.www.log - лог запросов к www
 - error.www.log - лог ошибок www
 - jobs.log - лог фоновых задач (обязательна ротация, лог очень большой)
 - php-error.log - лог ошибок PHP (для api и www вместе)
 - log_old - каталог в который складываются старые логи обмена модулей с приборами и модемами
- sessions - каталог для хранения сессий PHP
- uploads - каталог для временного хранения файлов, загружаемых пользователями

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 23

Сервисное API

Сервисное API предназначено для удаленного выполнения административных функций, а также для обеспечения взаимодействия нескольких систем между собой.

Общие принципы работы

API работает поверх HTTP. Запросы производятся на url вида <HOST>/service-api/*. Url определяет функцию или запрос, который требуется выполнить. Параметры запроса передаются в теле в формате JSON. Конкретный формат передаваемых параметров зависит от url.

Авторизация

Каждая система имеет ключ, прописанный в конфиге. В HTTP-запрос к API обязательно требуется добавить 2 заголовка: Client-Time - unix timestamp времени на клиенте и Signature. Signature вычисляется как sha256 от конкатенации \$a.\$b.\$c, где \$a - тело запроса, \$b - Client-Time, а \$c - ключ системы.

Система может ответить с ошибкой авторизации в случае если хэш не совпадает либо если время на системе сильно отличается от Client-Time. Такая мера нужна для защиты от атаки вида "повтор запроса". В случае успешной авторизации система отвечает на запрос.

Возможные запросы к системе

Получение списка модемов GET /data-fetcher/list

Запрос представляет собой пустой JSON-объект {}.

В ответе передается список моделей DataFetcher.

Получение списка устройств по идентификаторам модема (коммуникатора) GET /plc/list

В запросе передается identifiers - список идентификаторов модемов (data_fetchers.identifier).

В ответе выдается массив. Ключом является идентификатор модема, значением - список моделей PLC с атрибутами (тех, которые привязаны к этому модему).

Выгрузка очереди пакетов с данными GET /dt-pack/list.

В запросе передается plc_ids - массив идентификаторов приборов, по которым требуется получить входящие пакеты (из очереди tbl_traff_dtpack), а также last_id - идентификатор последнего пакета, уже известного клиенту и limit - сколько пакетов выдать. Пакеты выдаются с id > last_id в порядке увеличения id, общим количеством не более limit.

В ответе передается список пакетов. Каждый пакет состоит из следующих полей:

- id - идентификатор
- plc_id - идентификатор прибора
- data_type - тип данных
- dt - момент времени, за который данные
- reg_dt - время регистрации пакета
- with_limits - флаг необходимости пересчета аварий
- data - данные. Массив

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 24

- code - код параметра
- node_num - номер УУ
- data_type - тип данных
- value - значение
- fault - код ошибки

Получение списка зарегистрированных параметров на приборе GET /value/index

В запросе передается объект с полем plc_id - идентификатор прибора, на котором требуется получить Values. В ответе будет передан JSON - массив моделей Value, сериализованных как обычно.

Для моделей не передаются поля, обновляемые в ходе обработки данных системой (т.е. модель выдается "как при экспорте").

Получение архива данных по параметру GET /value/data/:id

В запросе также передается start_dt - таймстамп, начиная с которого нужны данные. В ответе выдается JSON-массив объектов с такими полями:

- datetime - время регистрации значения (в формате времени, а не таймстампа дабы убрать неоднозначность, связанную с неумением приборов обрабатывать часовые пояса)
- val - само значение
- fault - код ошибки

Подсистема самодиагностики

ПЭВМ «УЗИТ» включает в себя небольшую систему самодиагностики, задачей которой является периодическое отслеживание выхода параметров системы за приемлемые рамки и сообщение об этом администратору системы.

Подсистема состоит из следующих компонентов.

1) Периодически выполняющаяся задача (консольная команда, производная от CronController), перебирающая все возможные *сигналы тревоги* по списку, проверяющая их и генерирующая уведомление в случае срабатывания.

2) Собственно, *сигналы тревоги* – это условия, которые требуется проверять.

3) *Уведомления*. В данный момент они не выносятся в отдельную иерархию классов. Уведомление происходит только по электронной почте.

Реализовано это следующим образом:

- Все уведомления лежат в отдельно namespace: atm>alerts.
- Для каждого вида сигналов тревоги заведен класс, производный от atm>alerts\Alert. В этих классах должны быть методы checkCondition - возвращает true, если условие сработало и getMessage - возвращает сообщение.
- Существует класс atm>alerts\Processor, с единственным публичным методом process. Он выбирает настроенные уведомления из конфигурации. Если сигнал тревоги срабатывает, нужно отправить уведомление. В настоящий момент уведомление - это письмо на емейлы, список которых задается в конфигурации.

Какие настроены уведомления:

- Слишком много необработанных пакетов в очереди
- Со связи пропало слишком много приборов за последний час (вычислять среднее количество

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 25

событий "прибор не на связи" за последние 3 часа, если последний час более чем на 30% отличается от двух предыдущих - генерировать сигнал тревоги.

- Случился exception в jobs.log или php-error.log. Время последнего exception хранится в файле-кэше в runtime.
- Резко изменилось среднее количество входящих пакетов с данными.

Система отправляет эти уведомления на адрес администратора, указанный в конфигурационном файле.

Настройка модуля service-api

Модуль service-api реализует сервисное API системы. Одной из основных задач этого модуля является обеспечение возможности перенаправления полученных данных между экземплярами системы.

Сервисное API в отличие от обычного работает на более низком уровне, в частности оно игнорирует настроенные права доступа, поэтому следует быть максимально осторожным раздавая куда-либо доступы к сервисному API.

Пример конфигурации модуля находится в modules/service-api/config-local.php.sample.

Описание параметров модуля

- apiKey - ключ доступа к API этой системы. Клиенты должны указывать его в запросах (см описание сервисного API).
- importEndpoints - список систем, из которых требуется получать данные
 - tag - тег (короткое mnemonic название, используется в некоторых консольных командах)
 - url - URL сервисного API системы, из которой будут получаться данные
 - apiKey - ключ API той системы
 - importValues - вытягивать ли описания параметров с прибора (если 1, будут копироваться с системы-источника, если 0 - будут использоваться только те, которые уже есть локально)
 - importDevices - массив описаний приборов, которые требуется импортировать (если не указан, будут импортироваться все доступные)
 - identifier - идентификатор data_fetcher
 - address - сетевой адрес прибора (не учитывается если не задан)
- registrationReportUrl - URL для оповещения о регистрации новых модемов. Нужен, если планируется автоматическое переключение модемов между серверами при регистрации.
- ownName - название системы для сервисного API
- existingSystems - список систем, от которых данная будет получать информацию о зарегистрированных модемах
 - name - название системы
 - apiKey - ключ системы (им она будет подписывать запросы-уведомления, нужен для проверки)
 - serverList - список серверов для обновления в модем

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 26

Команда `service-api/import-data-packets`

Перекидывает пакеты из систем, указанных в конфигурации модуля (см. `modules/service-api/config-local.php.sample`) в эту систему. Импортируются только пакеты из очереди на обработку (таблица `tbl_traff_dtrack`). Эта таблица очищается от пакетов, старше чем 3 дня, потому более старые архивы данных не будут выгружены.

Соответствие приборов определяется по паре "идентификатор `data_fetchar` + адрес прибора". Поэтому требуется следить за соответствиями идентификаторов и адресов приборов. Соответствие параметров определяется по тройке `code-node_num-data_type`.

Между запусками команды сохраняет последний импортированный `id`, при повторных запусках начинает с него.

Стоит заметить, что если время на системах сильно отличается, работать система не будет (см описание сервисного API).

Для регулярной работы импорта эта команда добавляется в `crontab`.

Команда `service-api/import-data`

Команда имеет 3 параметра:

- `endpoint` - тег системы, из которой получить данные
- `local_plc_id` - ID прибора в локальной системе, на который нужно получить архив
- `start_date` - дата начала архива

Команда запрашивает архив данных по указанному прибору с указанной системы начиная с заданной даты. В этом случае данные импортируются из хранилища, что позволяет получать архивы за длительное время.

Crontab и его генерация

При обновлениях системы периодически возникает проблема, связанная с необходимостью вносить изменения в `crontab`. Чтобы решить эту проблему, `crontab` будет формироваться в ходе выполнения процедуры обновления системы отдельной командой `crontab/build`.

При этом итоговый `crontab` формируется из следующих частей:

- Основной файл `config/crontab.full`, единый для всех экземпляров систем.
- Дополнительный файл `config/crontab.local` (на случай, если конкретно этому экземпляру системы требуются отдельные крон-задачи)
- Кронтабы установленных модулей. Если в модуле нужен свой кронтаб, модуль должен реализовывать интерфейс `atm\interfaces\CrontabProvider`. Каким образом при этом будет выдаваться кронтаб модуля - решительно нет разницы.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 27

Сбор и хранение данных

Архитектура системы сбора данных

ПЭВМ «УЗИТ» поддерживает различное оборудование и различные способы его подключения к системе. Здесь и далее окончательным оборудованием (или прибором) называем устройство, с которого непосредственно считываются показания (теплосчетчик, плк, расходомер и т.п.). Модемом будем называть устройство, обеспечивающее транспорт. Обычно это собственно GPRS/3G модем, или же конвертер RS232/485

<-> TCP. Иногда прибор имеет в своём составе ethernet-порт / gsm модем, в таком случае можно говорить о встроенном в прибор модеме (например ПСМ-300 в составе теплосчетчика МКТС или блок телеметрии АКСОН).

В базе данных прибору соответствует запись в таблице tblplc, модему - запись в таблице data_fetchers. Важно поле data_fetchers.identifier - по нему отличаются модемы друг от друга. В большинстве случаев в этом поле прописывается IMEI модема, но если модем работает не по основному протоколу, в поле может быть дополнительная информация. Формат этого поля для разных случаев описан в отдельной статье.

Основной протокол обмена. Модуль eks_server.

Основным способом получения данных с приборов является так называемый протокол 1СИМ. Модем, реализующий этот протокол, при загрузке устанавливает соединение с сервером ПЭВМ «УЗИТ» (обычно на порт 21001), на котором слушает модуль eks_server. После этого модуль eks_server считывает из модема его идентификационные данные, определяет список приборов, связанных с этим модемом и запускает опрос показаний. Для реализации опроса в протоколе реализованы команды вида "установить настройки rs-линии", "отправить указанные байты в rs-линию", "произвести чтение из rs-линии". Такая схема позволяет добавлять в систему поддержку новых устройств без обновления программного обеспечения модемов.

Более детально протокол обмена модема и модуля eks_server описан в соответствующей статье.

Стоит отметить, что этот протокол может быть реализован и в устройстве, не являющимся GPRS/3G модемом, например, в конвертере интерфейсов ethernet / rs232. Особенности работы по ethernet также предоставлены отдельно.

Дополнительные протоколы модемов. Модуль eks_fork.

Для поддержки большого количества моделей модемов реализован модуль eks_fork. Он прослушивает соединения на различных портах, при подключении клиента сам устанавливает соединение с eks_server и конвертирует протоколы различных модемов в протокол 1СИМ. Т.е eks_fork в некотором смысле служит прокси-сервером для разных модемов. Детали трансляции у каждого протокола свои, при этом в некоторых случаях модулю eks_fork требуется подключение к базе (для хранения каких-либо настроек между коннектами).

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 28

Прозрачный протокол

Ввиду того, что на рынке выпускается большое количество оборудования, поддерживающего режим прозрачного канала между rs232 / rs485 и tcp-соединением, в систему введена поддержка этого режима. Для этого при регистрации устройства с его "модемом" (записью в data_fetchers) связывается IP-адрес сервера, куда будет подключаться устройство и номер TCP-порта. Модуль eks_fork периодически обновляет у себя информацию о таких "модемах" и осуществляет прослушивание соединений на всех требуемых портах. Таким образом, устройства, работающие в режиме tcp-клиента идентифицируются по ip-адресу сервера, на который они подключились и порту.

Также система позволяет работать с устройствами, которые сами работают как TCP сервер. В таком случае с "модемом" связывается IP адрес и порт, на которых устройство прослушивает соединения. Модуль eks_fork пытается установить TCP-соединение с указанным адресом и портом и при успехе соединяется с eks_server. Таким образом эти устройства идентифицируются по своему собственному IP и порту.

Протокол iRZ Collector

Это протокол фирмы "Радиофид", реализованный в их модемах ATM2 / ATM3. По умолчанию eks_fork слушает соединения этого протокола на порту 31003. Поскольку в рамках протокола передаётся IMEI модема, идентификация ведётся по нему.

Так как протокол 1СИМ предполагает изменение настроек rs-линии "на лету", а протокол iRZ Collector только путем переподключения модема и выхода в настроечный режим, изменение настроек записывается в базу. Поэтому не рекомендуется изменять настройки rs-линии в модеме после его подключения к системе, а также привязывать к одному такому модему несколько приборов с разными настройками rs.

Протокол АССВ-030

Этот протокол используется при обмене с приборами через модемы АССВ-030 фирмы Взлёт. Стандартным портом в большинстве версий прошивок АССВ является 2060, потому eks_fork слушает именно этот порт.

Протокол БАРС-02

Блоки БАРС-02 производства компании Промсервис используют свой протокол обмена (штатная система для них - САДКО-Тепло). Отличием является наличие двух портов на модеме (rs232 и rs485), поэтому при регистрации устройства в системе требуется выбрать, на каком из портов модема оно находится. Этот выбор пишется в tblplc.port.

Порт по умолчанию для этого протокола - 31004.

Протокол RobustLink

Это протокол, поддерживаемый некоторыми модемами Robustel. Механизм работы системы с ним аналогичен iRZ Collector. Однако в этом протоколе доступно переключение настроек по ходу обмена, поэтому проблем, приведённых выше в нём нет.

Порт по умолчанию для этого протокола - 31005.

Протокол 1СИМ (старый)

Компания "Термотроник" некоторое время назад выпустила ethernet-адаптеры для своих теплосчетчиков и расходомеров с поддержкой старой версии протокола 1СИМ (версия 1, см. описание). Для унификации обмена с ними была реализована соответствующая трансляция в eks_fork.

Порт по умолчанию для этого протокола - 31002.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 29

Расширения протокола 1СИМ для обмена с eks_fork

Для того, чтобы eks_fork мог эффективнее отслеживать границы пакетов при обмене с приборами, введены дополнительные команды протокола 1СИМ, которые применяются только при общении eks_server и eks_fork. Под эти команды зарезервированы коды 20-50 (см. Протокол 1СИМ).

Для определения, что к eks_server подключился именно eks_fork, а не модем, eks_server проверяет третью строку инициализации (CID сим-карты). Для подключения от eks_fork в ней всегда будет "eks_fork".

Команда 20 - подсказка об используемом далее протоколе.

После кода команды передается одна нуль-терминированная строка - название протокола, который eks_server будет в дальнейшем использовать для общения с прибором. При переключении приборов (если на одном коммуникаторе их несколько) eks_server может отправить эту команду еще раз.

При получении этой команды eks_fork запоминает, какой протокол считается "текущим". Знание об этом протоколе может быть использовано, в частности, для определения границы сообщений при исполнении команды 0x06 (прием данных). Имеющиеся протоколы на текущий момент:

- unknown - протокол по умолчанию, никаких явных предположений сделать нельзя
- modbus_rtu - стандартный modbus rtu
- spt961 - магистральный протокол фирмы "Логика", используемый в приборах СПТ961 и некоторых других В ответ на команду 20 eks_fork присылает 1 байт - подтверждение о получении этой команды.

Встроенные модемы / адаптеры

Некоторые устройства со встроенными модемами используют свой отдельный протокол для обмена с сервером. Способ получения данных в каждом из таких случаев свой.

- МКТС СБ-04 / СБ-05 с модемом РСМ-300. В этих модемах был реализован так называемый протокол BASLink. Он работает поверх HTTP, запросы отправляет модем, а в ответе он получает команду, какую информацию нужно вернуть в следующем запросе. Обработчик этого протокола расположен в web-приложении API, в каталоге baslink
- Блок телеметрии АКСОН. Он имеет собственный протокол, обработчик которого реализован внутри eks_fork. Соединение выполняется на отдельный tcp-порт (по умолчанию 31006).
- Батарейный модем Телеофис RX100-B1. Работает аналогично АКСОНу, по умолчанию на порту 31007.

Пакеты данных и передача их на обработку

Вне зависимости от способа подключения прибора, данные с него попадают в так называемые пакеты. Пакеты лежат в таблице tbl_traff_dtpack. С ней же связана таблица tbl_traff_valsfloat - это значения из соответствующего пакета. В tbl_traff_valsfloat хранится идентификатор параметра (tblvaluesactive), его значение и код ошибки. В самом пакете дополнительно хранится метка времени, связанная с данными (например 2017-01-01, если это посуточный архив за 1 января) и время регистрации пакета.

При создании пакета его поле status выставляется в created, после заполнения всех связанных записей (tbl_traff_valsfloat) его нужно выставить в prepared. После этого пакет готов к обработке, после которой его status будет выставлен в finished.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 30

Общий принцип хранения данных в системе

На каждом устройстве, зарегистрированном в системе, формируется набор параметров (из предустановленных, либо путём добавления через интерфейс, либо путём копирования или импорта с других устройств). У каждого параметра есть идентификатор (поле id в таблице tblvaluesactive), который является уникальным в рамках системы. Также параметр можно уникально определить по набору полей (plc_id, code, node_num, data_type).

Поле data_type определяет тип данных параметра. Типы данных бывают следующими:

- c - (current) - моментальные. Это данные с устройства, собранные непосредственно в какой-то момент времени. Такими данными могут быть накопленные массы/объёмы/энергии, текущая температура, моментальный расход итп.
- h - (hour) - почасовые. Это данные, усреднённые или суммированные за определённый час. Причём они могут быть как прочитаны из памяти устройства, так и получены путём вычитания интеграторных показаний либо вообще сформированы системой по заданным формулам.
- d - (day) - посуточные. Аналогичны почасовым, но содержат данные за сутки.
- i - (integral) - интеграторные. Это показания, суммированные от некоторого условного момента начала (обычно это дата производства устройства) до заданного момента времени. В руководствах на устройства такие данные называются накопительным итогом или просто итогами.

Система хранит по каждому параметру набор записей следующего вида:

- dt - момент времени, за который получены данные (более подробно о моментах времени ниже);
- value - значение параметра в момент времени dt - вещественное число;
- fault - код ошибки и флаги (64-битное целое). В этом числе младшие 48 бит отведены под код ошибки, возникший при чтении параметра (такие коды сообщаются устройствами при опросе и сигнализируют о проблемах с датчиками, питанием и другими вещами), а старшие 16 бит под флаги. Флаги (начиная с младшего бита флагов) имеют следующий смысл:
 - 0 - флаг автопересчёта. Выставляется данным, полученным путём вычисления сумм или разностей данных с устройства.
 - 1 - флаг автозаполнения. Выставляется данным, полученным путём "округления" временных меток полученных от устройства (см. ниже), а также данным, полученным путём заполнения недостающих интеграторных показаний.
 - 2 - флаг прогноза. Выставляется данным, которые являются прогнозом на будущее (погода например). В зависимости от типа параметра смысл метки времени dt в данных различается:
- c - для этих данных dt - момент времени, в который они были получены. Само dt может быть произвольным. При этом dt предполагается по времени сервера;
- h - для почасовых данных dt - это начало часа, за который получена указанная сумма (или среднее). Само dt обязательно кратно часу (т.е. минуты и секунды равны 0). Например, посуточное значение за 2019-01-20 05:00:00 - это соответственно сумма или среднее на отрезке [2019-01-20 05:00:00; 2019-01-20 06:00:00);
- d - для посуточных данных метка времени аналогична почасовым, однако кратна дню, а не часу и также указывает на начало искомого дня;
- i - для интеграторных данных dt также кратно часу и обозначает момент, до которого считается сумма (т.е. интеграторные данные с меткой 2019-01-20 05:00:00 обозначают накопленный итог на интервале (-inf; 2019-01-20 05:00:00]).

При этом можно заметить, что посуточный и почасовой параметры могут быть получены путём вычитания некоторых интеграторных и наоборот, интеграторный может быть получен путем прибавления / вычитания

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 31

почасовых и посуточных из интеграторного за другой момент времени. Система автоматически (в модуле `data_processor`) насчитывает эти суммы и разности и сохраняет их в базу, помечая флагом автопересчёта.

Также, на некоторых устройствах можно собирать только моментальные показания (и накопленный итог на момент времени опроса). Чтобы по таким устройствам можно было формировать отчётность, метки времени при опросе округляются, например, если были опрошены данные за 2019-01-01 13:24:54, то формируется интеграторное показание за 2019-01-01 14:00:00 (а при необходимости и далее до следующего момента опроса). Такие данные помечаются флагом автозаполнения.

Конфигурация хранилища данных

На текущий момент для хранения данных используется СУБД MySQL, однако разработаны механизмы для использования других `sql` и `nosql` баз данных. По мере реализации альтернативных хранилищ данных эта документация будет дополняться.

Хранилище данных MySQL (MariaDB)

Как непосредственно хранятся данные

Для хранения используются таблицы с названиями `values_YYYYMM`, каждая такая таблица содержит записи о показаниях за месяц `MM` года `YYYY`. Такое решение обусловлено тем, что со слишком большими таблицами очень неудобно работать в плане администрирования БД (длительное восстановление при сбоях, невозможность восстанавливать данные частично и т.п.).

На данный момент поддерживается два формата хранения данных в таких таблицах MySQL: старый (исторически сложившийся) и новый (расходует меньше места при пересчётах и прочих повторных обновлениях). Настоятельно рекомендуется во всех активных инсталляциях использовать новый формат (а если остались инсталляции со старым - конвертировать из старого в новый).

Старый формат хранения

Каждая `values`-таблица имеет следующую структуру:

- `id BIGINT` - идентификатор записи. Используется в частности для определения наиболее актуального значения
- `tva_id INT` - идентификатор параметра
- `dt TIMESTAMP` - метка времени
- `value DOUBLE` - собственно значение
- `fault BIGINT` - код ошибки и флаги

Первичным ключом является поле `id`, также есть индекс по (`tva_id`, `dt`, `id`) для быстрой выборки значений параметра.

При этом если по параметру в хранилище добавляется новое значение, то заводится новая запись с таким же `dt` и `tva_id`, соответственно при выборке значений из хранилища требуется выбирать запись с максимальным `id`.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 32

Новый формат хранения

В новом формате каждая таблица имеет следующую структуру:

- tva_id INT - идентификатор параметра
- dt TIMESTAMP - метка времени
- value DOUBLE - значение
- fault BIGINT - код ошибки

В этом формате первичным ключом является пара (tva_id, dt), соответственно при получении нового значения запись требуется обновлять.

Использование нескольких баз

Хранилище может использовать несколько баз, данные из которых собираются и агрегируются при доступе к ним. Это бывает полезно при конвертации баз между разными форматами (на время процедуры конвертации используются обе базы) или при восстановлении после сбоев (чтобы не тратить время на восстановление всего массива данных, можно запустить систему с новой базой, а имеющиеся данные восстанавливать из резервной копии параллельно).

Конфигурация хранилища в web-интерфейсе системы

За доступ к хранилищу отвечает класс MySQLStorageDriver. Его конфигурация находится в config/common.php (и можно переопределить в config/common-local.php).

Пример конфигурации выглядит следующим образом:

```
'storageDriver' => [
    'class' => 'atm\components\storage\MySQLStorageDriver',
    'writeDatabase' => [
        'class' => 'yii\db\Connection',
        'dsn' =>
'mysql:host='.LOCAL_CONFIG_DB_HOST.';dbname='.LOCAL_CONFIG_DB_NAME,
        'username' => LOCAL_CONFIG_DB_USER,
        'password' => LOCAL_CONFIG_DB_PASSWORD,
        'charset' => 'utf8',
    ],
    'defaultReadDatabases' => [
        [
            'class' => 'atm\components\storage\DbConnection',
            'dsn' =>
'mysql:host='.LOCAL_CONFIG_DB_HOST.';dbname='.LOCAL_CONFIG_DB_NAME,
            'username' => LOCAL_CONFIG_DB_USER,
            'password' => LOCAL_CONFIG_DB_PASSWORD,
            'charset' => 'utf8',
            'tag' => 'old',
            'active' => 1,
            'format' => 'old',
        ],
    ],
],
```


Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 33

Здесь:

- class - название класса (для этого драйвера всегда такое)
- writeDatabase - база в которую будет вестись запись новых данных. На текущий момент такая база может быть только одна. Конфигурация базы осуществляется аналогично стандартным настройкам yii-класса yii\db\Connection.
- defaultReadDatabases - массив конфигураций баз, из которых требуется читать данные. Каждый элемент массива - описание базы, оно также аналогично стандартному yii\db\Connection, однако имеет некоторые дополнительные параметры.
 - class - необходимо использовать atm\components\storage\DbConnection (он содержит дополнительные поля, используемые драйвером)
 - dsn - строка подключения (стандартный параметр)
 - username - имя пользователя (стандартный параметр)
 - password - пароль (стандартный параметр)
 - charset - кодировка (стандартный параметр)
 - tag - короткое мнемоническое название базы (используется в командах обслуживания хранилища как идентификатор). Теги должны быть уникальными.
 - active - активность (0 или 1). Неактивные базы могут быть использованы в командах обслуживания (например для переноса данных или оптимизации хранения), но не будут использоваться при получении данных. Это бывает полезно при инициализации новых баз (добавляем в конфигурацию их неактивными, задаем структуру, после чего активируем).
 - format - old или new. Формат, в котором эта база хранит данные.

При переопределении параметров с локальной конфигурации (common-local.php) вместо defaultReadDatabases требуется использовать readDatabases (это сделано для того, чтобы можно было целиком переопределить массив с конфигурациями баз).

Конфигурация хранилища в подсистеме сбора и обработки

Хранилище настраивается в конфигурационном файле модуля data_processor, который по умолчанию расположен в /home/atmuser/deploy/data_processor/data_processor.cfg.

Пример конфигурации:

```
StorageDriver = {
    Type = "composite"
    Drivers = [
        {
            Type = "mysql"
            DBLogin = "user"
            DBPassword = "password"
            DBHost = "127.0.0.1"
            DB = "atm"
            Format = "old"
        }
    ]
}
```

Здесь:

- StorageDriver - конфигурация хранилища (set-параметр)
 - Type - тип (на данный момент бывает composite и mysql)
 - Для типа composite:

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 34

- Drivers - список конфигураций вложенных драйверов (каждый имеет такую же структуру как параметр StorageDriver)
- Для типа mysql:
 - DBLogin - логин в базу
 - DBPassword - пароль в базу
 - DBHost - хост с базой
 - DB - имя мб
 - Format - old или new, формат хранения

Для использования нескольких баз требуется задать драйвер типа composite и ему в Drivers прописать конфигурации всех нужных БД.

Команды для работы с хранилищем и его обслуживания

Команды для работы с хранилищем

Создание таблиц хранилища

Для того, чтобы создать values-таблицы с нужной структурой, используйте команду

```
./yii generate-tables
```

Она создаст таблицы во всех базах, помеченных как readDatabases в конфигурации хранилища (в неактивных в том числе).

Удаление таблиц хранилища

Если база хранилища больше не нужна (или, например, хранилище было вынесено в отдельную базу и хочется из предыдущей базы удалить таблицы), используйте следующую команду

```
./yii generate-tables/drop <TAG>
```

Здесь:

TAG - тег базы (из конфигурации) из которой надо удалить таблицы

Перенос данных между базами

Для переноса данных из одной настроенной базы в другую (например, при выносе таблиц хранения в отдельную базу) используйте команду

```
./yii value-data/move <FROM_TAG> <TO_TAG> <TABLE>
```

Здесь

- FROM_TAG - тег базы-источника (откуда переносим данные)
- TO_TAG - тег базы-приёмника
- TABLE (по умолчанию пустая строка) - какую таблицу переносить. Если не указано, будут перенесены все таблицы по очереди.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 35

Вынос базы хранения данных из основной БД системы

Для выноса базы хранилища в другую базу (например, на отдельный сервер или с другими подобными целями) можно воспользоваться следующей последовательностью шагов:

1. Создаём новую базу, прописываем её в конфигурацию web-приложения, активность ставим в 0.
2. Создаём в ней таблицы при помощи соответствующей команды.
3. Прописываем новую базу в конфигурацию подсистемы обработки, перезапускаем data_processor.
4. Прописываем в конфигурации web-приложения её активной, меняем writeDatabase на вновь созданную базу. После этого шага новая база начнёт наполняться данными.
5. Переносим данные из старой базы в новую соответствующими командами. Этот шаг может выполняться долго, в зависимости от объёма данных в системе.
6. Убираем старую базу из конфигурации модуля обработки, деактивируем в конфигурации web.
7. Удаляем в старой базе таблицы соответствующей командой.
8. Удаляем старую базу из конфигурации web.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 36

Консольные команды и полезные утилиты

Экспорт и импорт сущностей в системе

Общие принципы

Более-менее каждую модель можно сериализовать в json и десериализовать оттуда. Возникает только вопрос в том, как хранить связанные объекты. Здравый смысл подсказывает нам, что они должны быть вложенными в основной. Но что делать, если привязанных объектов может быть несколько?

Если какому-то объекту А нужна ссылка на какой-то другой объект В, то мы генерируем объекту В GUID (пишем в отдельное поле) и при экспорте выписываем этот GUID. Соответственно при импорте мы эти GUID-ы разрешаем в настоящие ссылки (если можно). Для некоторых моделей вместо GUID будем использовать другие идентификаторы (список, где что приводим ниже). При этом некоторые объекты мы порождаем как вложенные.

При импорте объектов сначала создаём сам объект, потом все вложенные (и разрешаем ссылки). При этом внешний объект (внутри которого создаётся текущий) назовём контекстом импорта. Перед тем как создавать новый объект, нужно проверить, нет ли в текущем контексте импорта вложенного объекта с таким же GUID например. Если таковой найдётся, его надо обновить вместо создания нового объекта.

Например, при экспорте приборов с объекта им всем задается GUID. Далее, при импорте производится проверка, есть ли прибор с таким GUID на объекте. И если есть, то обновляется.

При этом для параметров в качестве такого уникального идентификатора используем `code + data_type + node_num`. По историческим причинам считаем параметры вложенными в прибор, а не в узел учёта (это облегчит экспорт отдельно списка параметров).

Также чтобы не генерировать огромные вложенные структуры в json, будет использоваться конструкция вида "json внутри json", когда вложенный объект вкладывается в родительской в виде уже сериализованной строки. Набор вложенных объектов таким образом становится массивом строк в содержащем их объекте.

Экспорт сложных структур с привязанными файлами

Объекты, содержащие в себе много вложенных, особенно если вложенные объекты могут быть большими (картинки, файлы итп) экспортируются в виде zip-архива, в котором содержатся экспортированные json-ки и другие файлы. При этом внутри json ссылки на файлы делаются по относительному пути.

При импорте соответственно требуется распаковать этот архив и десериализовывать файлы согласно путям в архиве.

Базовые объекты и их поиск при экспорте и импорте

Назовём базовым объектом системы объект, который мы не будем считать вложенным ни в какой другой объект. При импорте и экспорте базовых объектов они ищутся глобально в системе по их uuid (или другому идентификатору, если таковой для объекта имеет смысл (см. список ниже)). Если для переноса некоторой сущности (прибор, компания, объект и т.п.) требуются связанные с этой сущностью базовые объекты, результатом экспорта должен быть архив, в котором будет сериализованная версия экспортируемой сущности и отдельно экспортированные базовые объекты.

Заметим, что возможно экспортировать не базовый объект, но при этом в архиве экспорта иметь дополнительно сериализованные отдельно базовые, связанные с ним. В частности, при экспорте прибора с какого-то объекта, этому прибору нужен датафетчер. Но при этом прибор не является базовым объектом, а

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 37

датафетчер является. Такая же ситуация и с экспортом объекта (который вроде как вложен в компанию).
При этом для корректного импорта базовых объектов им может требоваться некоторый контекст

Список базовых объектов (экспортируемых каждый по отдельности)

1. User - пользователь
2. Company - компания
3. DataFetcher - модем (технически он принадлежит компании, но если его не сделать базовым, не будет работать экспорт и импорт приборов)

Список вложенных объектов

1. PLC - прибор, вложен в PageObject
2. PageObject - объект, вложен в Company
3. Value - параметр, вложен в PLC
4. Limit - авария, вложена в PLC
5. Node - узел учета, вложен в PLC
6. NotificationRule - настройка уведомления, вложена в Limit
7. StringParam - строковый параметр, вложен в PLC
8. Report - форма отчета, вложена в Company или PageObject или Group
Замечание: в качестве контекста для импорта формы отчета может служить компания, объект или группа. Базовые формы отчета цепляются к компании целиком, макроотчеты соответственно к прибору или группе.
9. ReportItem - элемент формы отчета, вложен в Report
Замечание: ReportItem в макроотчетах ссылается я на конкретные параметры конкретных приборов. Для корректного разрешения ссылок используются uuid приборов и тройки code/node/data_type параметров.

Список моделей, использующих в качестве идентификатора не GUID

1. User - пользователь, идентифицируем по email
2. DataFetcher - модем, идентифицируем по identifier
3. Value - параметр, идентифицируем по (code, node_num, data_type)
4. StringParam - строковый параметр, идентифицируем по (code, node_num)
5. PageRegion. Поскольку uuid введен позже, регион определяется последовательностью name регионов, начиная от корня. (например ["Россия", "республика Карелия", "г. Петрозаводск"]).
6. schemes/Object - "tag" - уникальный идентификатор объекта (еще и как название папки используется, для новых объектов)
7. schemes/ObjectClass - "tag" - уникальный идентификатор класса объектов

Команды для работы с импортом/экспортом

Для импорта и экспорта существуют команды import/* и export/*. Конкретный action определяется типом объекта, который нужно импортировать или экспортировать.

В команде экспорта параметром обычно служит id экспортируемого объекта и имя файла с результатом, в команде импорта - id объекта-контекста и имя файла, откуда импортировать.

Например, `./yii export/object 12345 a.zip` - экспортировать объект 12345 в файл a.zip

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 38

Предустановленные объекты

Формы отчетности

Предустановленные формы отчетности лежат в репозитории, в каталоге `modules/reports/predefined`. Для их импорта можно использовать команду `./yii reports/import/predefined ./modules/reports/predefined`

Для выгрузки предустановленных форм отчетности требуется использовать команду `./yii reports/export/predefined PATH_TO`

Стоит заметить, что `PATH_TO` должен быть каталогом, поскольку команда сгенерирует набор файлов с предустановленными формами.

Мнемосхемы

Для начала надо загрузить классы объектов из `modules/schemes/predefined/classes`, вот такой командой: `./yii schemes/import/predefined-classes ~/atm/modules/schemes/predefined/classes`

Потом можно загружать сами схемы из `modules/schemes/predefined/schemes`: `./yii schemes/import/predefined-schemes ~/atm/modules/schemes/predefined/schemes`

Полезные утилиты в системе

Информация о различных консольных командах системы и вспомогательных утилитах

Если не указано обратное, предполагается, что запускать все команды нужно от имени `atmuser`. При этом все команды, начинающиеся с `./yii` запускаются из каталога `/home/atmuser/atm`.

Управление компаниями

Удаление компании

Чтобы удалить компанию, используйте команду

```
./yii company/delete <COMPANY_ID>
```

Здесь:

- `COMPANY_ID` - идентификатор компании для удаления

Команда удалит компанию, всех её пользователей, объекты, группы и устройства, которые в ней есть. Настоятельно рекомендуется для удаления компаний пользоваться именно этой командой, чтобы не оставлять мусора в базе системы.

Деактивация / активация событий и аварий в компании

Чтобы деактивировать все события/аварии (кроме "на связи") в компании, используйте команду

```
./yii company/deactivate-all-limits <COMPANY_ID>
```

Здесь:

- `COMPANY_ID` - идентификатор компании для

деактивации Для активации всех событий используем

команду

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 39

```
./yii company/activate-all-limits <COMPANY_ID>
```

Работа с данными

Экспорт данных

Архив данных с прибора можно экспортировать командой

```
./yii data/json/export <DEVICE_ID> <START_DATE> <END_DATE>
```

Здесь:

- DEVICE_ID - идентификатор устройства, с которого надо вытянуть данные
- START_DATE - дата начала выгрузки (в формате YYYY-MM-DD)
- END_DATE - дата окончания выгрузки (в формате YYYY-MM-DD)

Команда выведет в консоль архив данных, пригодный для последующего импорта. Поскольку размер архива может быть большим, рекомендуется перенаправлять результат работы команды в файл.

Импорт данных

Экспортированный архив можно импортировать командой

```
./yii data/json/import <DEVICE_ID> <FILE_NAME>
```

Здесь:

- DEVICE_ID - идентификатор устройства, с которого надо вытянуть данные
- FILE_NAME - имя файла с данными

Команда загрузит в систему данные из архива как будто бы они были прочитаны с устройства (с обработкой аварий и зависимых параметров). После срабатывания команды возможно потребуется некоторое время на обработку данных.

Пересчёт вычисляемых данных

```
./yii data/recount <START_DATE> <END_DATE> <DEVICE_ID> [--cleanup]
[--data_type=<DATA_TYPE>]
```

Здесь:

- START_DATE - дата начала пересчёта
- END_DATE - дата окончания пересчёта
- DEVICE_ID - идентификатор устройства
- --cleanup - если указано, из системы будут удалены ранее вычисленные данные
- --data_type - если указано, будут пересчитаны только данные типа DATA_TYPE. Если вычисление посуточных данных занимает много времени, а ошибок в них нет, можно пересчитать только интеграторы.

Команда запускает повторное вычисление значений получаемых путём расчёта архивов и интеграторов на основе друг друга, значений получаемых как сумма и среднее на интервале времени итп. Флаг cleanup нужно указать, если требуется очистить хранилище от старых вычисленных данных (если в них есть ошибки например).

Перезапуск подсистемы обработки данных

Для перезапуска модуля обработки данных (после обновлений или если есть подозрение о его некорректной работе, можно воспользоваться командой):

```
./yii data/process/restart [--force]
```

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 40

Здесь:

- --force - обычно команда проверяет, что перезапускаемый процесс удерживает блокировку (которую он должен удерживать для защиты от повторных запусков). Если блокировка не удерживается, это может свидетельствовать о том, что процесс уже не работает. При указании этой опции будут перезапущены даже процессы, которые не удерживают блокировку. Следует быть аккуратным - указание этой опции может перезапустить в итоге процесс, который не имеет никакого отношения к модулю обработки данных.

Утилиты для диагностики и устранения неполадок в системе

Починка статуса устройства "на связи"

Если устройство не на связи, но соответствующая плашка в интерфейсе не отображается (и параметр "связь" выставлен в "Есть"), может помочь следующая команда:

```
./yii fix/connected
```

Она запускает форсированный пересчёт значений переменной "Связь" на всех устройствах.

Восстановление предустановленных параметров

Может оказаться так, что по какой-то причине с устройства были удалены предустановленные параметры. Для их восстановления можно воспользоваться командой:

```
./yii fix/preset-values <DEVICE_ID>
```

Здесь:

- DEVICE_ID - идентификатор устройства, на котором надо восстановить параметры

Команда восстановит все предустановленные параметры на устройстве согласно тому, как они прописываются при регистрации. Дополнительные настроенные параметры она не трогает, однако сбрасывает формулы у предустановленных составных на те, которые прописаны в шаблоне.

Восстановление предустановленных аварий

Аналогично параметрам, можно восстановить предустановленные аварии командой:

```
./yii fix/preset-limits <DEVICE_ID>
```

Здесь:

- DEVICE_ID - идентификатор устройства, на котором надо восстановить аварии

Восстановление значений компании-владельца для dataFetcher'ов и вывод списка коллизий

Проверяет все dataFetcher на наличии "битой" ссылки id_p - т.е. ссылки на компанию. Пробует автоисправить. Если автоисправление невозможно (привязаны несколько приборов из разных компаний), то выведет строку ругани (всегда).

Запуск без параметра или с параметром = 1 автоисправлять ссылки на компании не будет, но отобразит возможные автозамены.

```
./yii fix/data-fetcher-company <dryRun>
```

Здесь:

- dryRun - 0 или 1 - отобразить ли только список автозамен или применить их сразу.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 41

Техническое обслуживание системы

Синхронизация очередей задач

Если в системе есть "зависшие" задачи в таблице tasks, а при этом для ускорения очередей используется Redis (т.е. почти всегда), возможно проблема в том, что в ходе обработки какой-либо задачи, запись из Redis уже была извлечена, но задача не была обработана.

Для исправления этой ситуации можно запустить

```
./yii task/tasks/sync
```

Очистка БД от лишних записей

В ходе работы системы из-за разных мелких сбоев и ошибок в базе могут оказываться лишние записи. Например, параметры или узлы учёта, не привязанные ни к каким приборам, приборы не привязанные к объектам итп. Для их удаления можно использовать команду

```
./yii cleanup/all <DRY_RUN>
```

Здесь:

- DRY_RUN - флаг тестового запуска (по умолчанию равен 1). При тестовом запуске утилита выдаст список записей, которые она удалит, если же выставить флаг в ноль, записи будут удалены из базы.

Также есть вариации этой команды, очищающие некоторые из возможных лишних записей:

- ./yii cleanup/orphaned-values <DRY_RUN> - удаляет параметры, не привязанные ни к какому прибору
- ./yii cleanup/orphaned-limits <DRY_RUN> - удаляет аварийные критерии, не привязанные ни к какому прибору
- ./yii cleanup/orphaned-limit-logs <DRY_RUN> - удаляет записи о срабатывании аварийных критериев, не привязанные к критерию
- ./yii cleanup/orphaned-devices <DRY_RUN> - удаляет приборы, не привязанные к объектам
- ./yii cleanup/orphaned-nodes <DRY_RUN> - удаляет узлы учёта, не привязанные к приборам

Смысл флага DRY_RUN такой же как и в основной команде.

Показатели системы для мониторинга

Для интеграции с разного рода системами мониторинга можно использовать описанные ниже показатели. По каждому из них описан механизм его получения (обычно путём выполнения команд из консоли), а также предполагаемые значения при штатной работе.

Количество пакетов с данными в очереди на обработку

Показатель можно получить выполнив команду `./yii task/queue/count dtpacks`

Эта команда выводит в консоль одно число - значение показателя.

Это количество так называемых пакетов с данными, которые должна обработать система. При штатной работе это значение должно быть около нуля, хотя при выполнении разного рода импорта данных может кратковременно подскакивать. Рекомендуется устанавливать триггер на превышение значения в 1000 как warning и 10000 как error.

Количество задач на досчёт параметров

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 42

Показатель можно получить выполнив команду `./yii task/queue/count data_processor`

Эта команда выводит в консоль одно число - значение показателя.

Показатель отражает количество задач для модуля пересчёта данных (`data_processor`). Это значение при штатной работе также должно быть около нуля, но может кратковременно увеличиваться. Рекомендованные значения для триггеров - 1000 (`warning`) и 10000 (`error`).

Количество фоновых задач web-интерфейса

Показатель можно получить выполнив команду `./yii task/queue/count www`

Эта команда выводит в консоль одно число - значение показателя.

Это количество фоновых задач, порождённых действиями пользователей. Такими задачами могут быть генерация отчётов, массовые групповые операции, рассылки уведомлений итп (всё, что может потребовать значительного времени на выполнение). Значение параметра должно быть почти всегда небольшим, рекомендуется делать триггер `error` на превышение значения в 100.

Работа бинарных модулей системы

Рекомендуется также отслеживать состояние бинарных модулей системы (`eks_server / eks_fork / data_processor`). Способ это сделать - считать количество процессов командой типа `ps ax | grep eks_server`

В таком варианте триггер стоит делать на уменьшение этого значения меньше 2 (один будет всегда `shell`, запущенный из `cron`, второй непосредственно модуль).

Количество емейлов в очереди на отправку

Показатель вычисляется командой `./yii email-queue/count`

Она также выводит в консоль единственное число.

Высокое значение этого показателя означает проблемы с почтовым сервером (с его настройкой или производительностью). В любом случае, это повод проверить его работу. Рекомендуется выставлять `warning`, если этот показатель больше 100 и `error`, если он стал больше 1000.

Количество параметров, которые не устарели, хотя должны были

Показатель вычисляется командой `./yii data/values/count-wrongly-fresh` Она также выводит в консоль единственное число.

Для определения ситуации, когда устройство пропало со связи используется так называемый механизм "устаревания" параметров (грубо говоря, параметр обращается в `null`, если некоторое время по нему не обновляются данные). Это критично для работы соответствующих уведомлений, в частности. Показатель отражает, какое количество параметров не имеют флага "устарел", хотя по косвенным признакам должны его иметь. Рекомендуется выставлять `error` если это значение больше 1000.

Если этот показатель растёт, рекомендуется также проверить размеры очередей на обработку данных, а если очередь небольшая, то проверить (и при необходимости перезапустить) задачу `./yii data/process/expiration`

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 43

Протоколы обмена

Протокол 1СИМ (модемный)

Здесь будет описан протокол общения сервера 1СИМ и модемов, используемый для опроса приборов. Основная идея - модем представляет собой "СОМ-порт, способный нивелировать задержки при передаче данных через GSM". Модем может реализовывать любой протокол, но более старшие версии протокола очевидным образом предоставляют больше возможностей.

Далее в описании словами "необходимо" и "должен" отмечается обязательная функциональность, словами "желательно" и "может" - опциональная.

Общие требования к реализации протокола модемом

Модем должен иметь возможность «резолвить» текстовое доменное имя (к примеру, google.com) в IP адрес (к примеру, 173.194.39.98). Доменное имя может само по себе являться IP адресом, в этом случае оно «резолвится» в само себя. Для «резолва» используются DNS серверы, чьи IP адреса получены у DHCP сервера при запуске. Конкретно «резолв» отдельно от коннекта не нужен, т.е. если функция TCP коннекта поддерживает доменные имена в качестве цели коннекта, то этого достаточно.

Требуется, чтобы модем мог хранить в энергонезависимой памяти четыре доменных имени вместе с портом при суммарной длине до 31 символов для каждой пары. Порт записывается через ':' после доменного имени (строка от 2 до 5 десятичных цифр). Домен может быть как текстовой ASCII строкой (к примеру, "google.com:80"), так и IP адресом в текстовом виде (к примеру, "173.194.39.9:12000").

Модем должен поддерживать внутренний буфер для фонового приема данных, доступных на RS линии. Это требуется в первую очередь для RS232 моделей в режиме согласования, чтобы прибор не прекратил передачу данных, если ему покажется, что его "никто не слушает". Достаточно 4096 байт для этого буфера и он должен заполняться во время всех потенциально долгих операций (обмен с GSM чипом и запись в RS линию, особенно в случае наличия согласования на RS линии).

Желательно, чтобы модем мог заметить входящий CSD звонок и ответить на него при поднятой GPRS сессии и при поднятом TCP коннекте. Удерживать их одновременно не требуется, достаточно возможности заметить сам факт звонка и принять его, разорвав TCP соединение. При приеме CSD звонка необходимо обеспечить прозрачный режим передачи между CSD данными и RS линией. После обрыва звонка уйти в reboot.

В модеме должен быть реализован полноценный TCP стек (по крайней мере в рамках одного коннекта) с IP фрагментацией, TCP ACK/SYN/FIN отправками, повторами при потере пакетов и контролем windowing. Т.е. gsm_write при установленном TCP коннекте должна вести себя так же, как socket_send в стандартных BSD sockets (в частности, gsm_can_write при установленном соединении должна возвращать 0, если забит буфер на стороне сервера (tcp window control)).

Как со стороны RS232 линии, так и со стороны TCP коннекта модем должен быть способен корректно принимать/отправлять любые последовательности байт, включая последовательности содержащие байты 0, последовательности содержащие строки вида "AT", "ATZ", "ATH", "+++", "\r\n", "CONNECT", "DISCONNECT", "OK" и т.п., не смешивая при этом данные в TCP коннекте с управляющими командами GSM чипа.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 44

Модем выпускается в 2-х вариантах - с интерфейсом передачи данных RS232 или RS-485. На RS232 варианте требуется возможность полного аппаратного RTS/CTS согласования. Так же желательна возможность управления сигналом DTR/DCD.

Общие принципы работы модема

При запуске модем инициализирует GPRS сессию. Для инициализации требуется APN, предполагается прописать дефолтный APN как internet.mts.ru;username=mts;password=mts. Хранить APN требуется в энергонезависимой памяти. Изменение APN действует через отправку SMS с текстом "APN=internet.mts.ru;username=mts;password=mts". Реагирование на SMS со сменой APN должно происходить всегда, когда модем не находится на связи с сервером ЕКС.

Если выйти на связь с сервером не получается, модем уходит в reboot. Перезагрузка должна быть "железной" по питанию. Модем должен выдерживать до 8000 reboot в год в течение минимум 3 лет. Автоматический reboot выставляется на раз в 12 часов, даже если со связью все в порядке.

После установления GPRS сессии модем пытается установить TCP соединение с одним из известных ему 4х серверов, пробуя их по кругу 0,1,2,3,0,1,2,3,... В случае 12 неуспешных TCP коннектов с определенными паузами и таймаутами - reboot.

После установки соединения модем проверяет, что сервер, с которым он только что соединился, действительно является нужным сервером. Это проверяется путем получения 4 байт из TCP соединения от сервера сразу после его установки. Эти 4 байта должны формировать стоку "ISIM" в ASCII кодировке, т.е. должно прийти 4 байта с кодами: 0x31, 0x53, 0x49, 0x4D.

После этого модем отправляет в TCP коннект свои "опознавательные знаки", а именно - 3 нуль-терминированные строки друг за другом. Если строки "abc", "def" и "123" соответственно (без кавычек), то в TCP коннект должны быть отправлены байты 0x61, 0x62, 0x63, 0x00, 0x64, 0x65, 0x66, 0x00, 0x31, 0x32, 0x33, 0x00. Первая строка - номер версии протокола, потом пробел и версия прошивки.

Версия протокола передается в текстовом виде (так как версия 1, 2 или 3, то соответственно байты 0x31, 0x32 или 0x33).

Версия прошивки строится следующим образом: тип модема (например, sim900), потом нижнее подчеркивание и версия прошивки (например, дата выпуска). То есть версия прошивки может иметь вид sim900_20130101. Более подробно об этом – в описании команды 0xFE.

Вторая строка - IMEI GSM чипа на модеме. Третья строка - CID сим-карты. После этого модем впадает в цикл получения команд от сервера.

В случае отсутствия документированных команд от сервера в течение 5 минут, либо при отсутствии обмена в течение 5 минут в прозрачном режиме (команда 0x01, см. ниже), TCP соединение должно быть разорвано. При поступлении недокументированной команды коннект должен разрываться.

Каждая итерация реагирования на команды сервера заключается в получении одного байта из TCP соединения с кодом команды. Дальнейшие операции по обработке команды зависят от того, какой байт был принят.

Важно заметить, что модему может быть прислано несколько различных команд подряд (без ожидания ответа модема). В таком случае модем должен обрабатывать все полученные команды в том порядке, в котором они были присланы. Можно предполагать, что одновременно будет прислано не более 1Кб команд.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 45

Протокол версии 1

1. Команда 0x01 - прозрачный режим

После получения такой команды модем отвечает в соединении байтом 0x01, после чего переходит в прозрачный режим между TCP соединением и RS линией. То есть все байты, поступающие в TCP соединение, отправляются в RS линию, а все байты, полученные по RS линии, отправляются в TCP соединение. В случае, если байты доступны на чтение и там, и там, приоритет отдается чтению RS линии.

Выходом из прозрачного режима может являться либо обрыв TCP коннекта (включая осознанный обрыв со стороны сервера), либо слишком долгая тишина в трафике (15 минут), либо таймаут на авторебут (12 часов). В цикл обработки команд сервера из прозрачного режима вернуться нельзя (только через переконнект или перезагрузку).

2. Команда 0x02 - обновление списка серверов

Модем получает из TCP коннекта 4 нуль-терминированные строки, так же как и три строки после установки коннекта (см. выше), каждая максимум 32 байта (вместе с терминирующим нулем). Если список серверов не отличается от уже известного модему, модем отвечает байтом 0x00 и возвращается в цикл обработки команд сервера. Если же список серверов хоть как-то отличается, то модем обновляет свой список в EEPROM, отправляет в TCP коннект байт 0x01, разрывает TCP соединение и перезагружается.

3. Команда 0x03 - искусственная длинная пауза

Модем получает сначала старший, затем младший байт паузы. Пауза даётся в секундах! (во всех остальных местах протокола звучат миллисекунды). После этого модем отправляет в TCP коннект байт 0x01, разрывает TCP коннект и засыпает на указанное количество секунд (вплоть до 60'000, т.е. почти сутки), после этого уходит в ребут. Во время ожидания модем должен иметь возможность реагировать на входящие CSD звонки, т.е. принимать звонок и после этого входить в прозрачный режим между CSD данными и RS линией, потом ребут.

10. Команда 0xFF - пинг

В ответ модем отвечает байтом 0x01, после этого модем возвращается в цикл обработки команд сервера.

Протокол версии 2

Все команды протокола версии 2 (кроме 0xFE и 0x14) являются обязательными к реализации. Если модем не реализует команду 0x14, необходимо корректно её разбирать и игнорировать (поскольку у сервера нет возможности узнать, поддерживает ли конкретно этот модем такую команду).

Протокол версии 2 включает в себя все команды протокола версии 1, а также следующие:

4. Команда 0x04 - настройка RS линии

Модем получает нуль-терминированную строку с настройкой COM порта с RS линией (максимум 32 байта вместе с терминирующим нулем). Стока имеет вид "9600;8N1;1", где первое число - скорость, второе число - количество бит с данными, следом - контроль четности (N, E, O), следом количество стоповых бит, следом - наличие RTS/CTS согласования (0, 1), присутствует как для RS232, так и для RS485 версий, 1 для RS485 версии должна игнорироваться. Относится к количеству бит во фрейме, а не во всей посылке. Числа в ASCII, разделяются ASCII символом «;».

В случае успешной установки параметров RS линии, модем отправляет в TCP коннект байт 0x01. В случае неуспешной - байт 0x00. Попытка установить согласование на RS485 модели должна игнорироваться (если же с точки зрения кода прошивки модема разницы между rs232 и rs485 нет, то требуется в любом случае попытаться установить согласование).

После установки параметров порта (хоть успешной, хоть неуспешной), модем должен очистить внутренний

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 46

буфер приема RS данных.

5. Команда 0x05 - отправка данных

Модем получает старший байт таймаута на отправку, затем младший байт. Таймаут задан в миллисекундах. После этого модем получает старший и младший байт количества байт на отправку. Следом идет указанное количество байт (от 1 до 1024).

Старшие два бита количества зарезервированы под флаги, т.е. количество байт формируется младшими 14 битами переданного 16-битного числа. 15й бит используется как флаг, что перед отправкой требуется очистить внутренний RS буфер, 14й бит всегда 0.

После приема всех байт из TCP коннекта модем отправляет их в RS линию, отслеживая таймаут.

После отправки модем записывает в TCP коннект сначала старший, затем младший байт количества успешно отправленных байт в RS линию, при этом, для количества используются только младшие 14 бит отправляемого 16-битного числа. Если отправка завершилась из-за таймаута, 15й бит должен быть выставлен в 1.

6. Команда 0x06 - прием данных

Модем получает старший, затем младший байт суммарного таймаута на прием (в миллисекундах). Затем модем получает старший и младший байты таймаута между приемом двух последовательных символов (в миллисекундах) - данный таймаут может равняться нулю, в этом случае он не используется. Затем модем получает длину терминирующего блока (от 0 до 32) в одном байте. Затем модем получает сам терминирующий блок, чья длина была указана в последнем полученном байте. Если длина 0, терминирующий блок не используется. Затем модем получает старший и младший байты максимального читаемого количества символов (от 1 до 1024). Старшие два бита этого количества зарезервированы под флаги, т.е. количество байт формируется младшими 14 битами переданного 16-битного числа. 15й бит используется как флаг, что перед приемом требуется очистить внутренний RS буфер.

После этого модем начинает принимать байты с RS линии, не забывая также использовать внутренний RS буфер для заполнения буфера чтения. Если накапливается указанное в запросе количество принятых байт или если пауза между подряд идущими символами превышает соответствующий таймаут (и если он ненулевой), или если последние принятые байты в точности равняются байтам терминирующего блока, прием завершается. Приём также завершается в случае, если дополнительный анализ ответа рапортует о законченном кадре (см. описание команды 0x14).

После этого в TCP коннект отправляется количество принятых байт - сначала старший байт, потом младший. При этом, количество записано в младших 14 битах передаваемого числа, старшие два бита являются флагами, описывающими причину конца приема. Если прием завершился по заполнению буфера, они должны равняться

00. Если прием завершился по общему таймауту, они должны равняться 10 (т.е. 15й бит 1, 14й бит 0). Если прием завершился по таймауту между двумя последовательными символами, они должны равняться 01. Если прием завершился по соответствию терминирующему блоку, они должны равняться 11.

После этих двух байт в TCP-коннект отправляются собственно принятые байты.

7. Команда 0x07 - очистка RS буфера

Модем очищает RS-буфер и отправляет 2 байта - длину буфера в момент очистки (младший, потом старший байт).

8. Команда 0x08 – управление пинами RS-линии

После команды идет нуль-терминированная строка, обозначающая, какой пин и в какое логическое значение нужно выставить. Формат этой строки - <название пина>=<значение>, например DTR=1 или CTS=0.

Модем отправляет серверу: в случае запроса RTS/DTR - байт 0x00 или 0x01 (в зависимости от состояния

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 47

пина), в случае запроса CTS/DCD/DSR - байт 0x00 в случае успешной установки пина, байт 0x01 в случае неуспешной (предполагается, что пины RTS/DTR для модема работают на чтение, пины CTS/DCD/DSR - на запись).

9. Команда 0xFE - обновление прошивки

После этой команды следует новая прошивка и ее контрольная сумма. Со стороны сервера прошивка должна представлять собой два файла: файл с прошивкой и файл с контрольной суммой. Формат как прошивки, так и контрольной суммы определяется разработчиком (в частности, считать контрольную сумму можно произвольным образом, с учетом требований ниже).

После команды 0xFE сервер отправляет 4 байта – размер файла прошивки (старший байт сначала), затем файл с контрольной суммой прошивки (его длина должна быть равна 32 байтам), после этого сам файл с прошивкой.

Требования к процессу автоматической прошивки:

- а) Достаточно защищенный от сетевых помех контроль целостности прошивки. CRC по этой причине не принимается. Как относительно простой в реализации вариант - MD5.
- б) Возможность принять прошивку, проверить ее целостность, и только потом применить. Вариант, когда принимаемая из сети прошивка "налету" затирает текущую, не принимается.

Во время приема прошивки можно не слушать RS линию (в частности, можно использовать RS буфер для приема байт прошивки), т.к. после прошивки всегда идет ребут.

10. Команда 0x0B - экстренное переподключение

Команда не имеет аргументов. При ее получении модем должен отправить серверу в ответ любой байт, после чего закрыть соединение и инициировать соединение заново. Команда используется в случаях когда серверу требуется переинициализировать контекст соединения (обычно в случае изменения данных, связанных с данным модемом).

11. Команда 0x14 - задание типа протокола.

После команды идет одна нуль-терминированная строка, содержащая название типа протокола, по которому будет идти дальнейший обмен с прибором.

Для ускорения обмена через модем, модемы могут реализовывать частичный анализ протоколов обмена с приборами (чтобы определять, когда прибор уже прислал ответ и не дожидаться таймаута опроса целиком). В частности, если например протокол modbus_rtu, при получении ответа модем может проверять контрольную сумму полученных байт согласно спецификации modbus и как только она сойдётся, отправлять ответ на сервер. Если протокол модему неизвестен, модем должен работать как будто этой команды не приходило (т.е. честно дожидаться истечения каждого таймаута).

Какие могут быть переданы протоколы:

- unknown - передаётся если протокол неизвестен или не поддерживается его анализ
- mercury20x - электросчетчик Меркурий 20х
- mercury23x - электросчетчик Меркурий 23х
- modbus_rtu - протокол Modbus RTU
- pulsar - счетчики импульсов Пульсар
- spbus - протокол СПСеть (фирма Логика)

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»	
Код документа:		Стр. 48

Протокол версии 3

Это черновая спецификация. В настоящий момент она обсуждается/дописывается.

Каждое сообщение протокола третьей версии имеет следующий вид

- 1 байт - код команды (от 100 до 200)
- 2 байта - длина отправляемых параметров команды (старший байт, потом младший)
- n байт - параметры команды

Если первым байтом модему приходит байт не в диапазоне от 100 до 200, его нужно рассматривать как команду 2-й версии протокола и обрабатывать соответствующим образом. При этом нужно помнить, что от сервера может придти много команд разом, а модем в таком случае должен их обрабатывать в порядке получения.

Ответ модема соответственно должен иметь следующий вид:

- 2 байта - длина ответа (старший байт, потом младший)
- n байт - собственно ответ

Если отправляемая команда модему не известна (или не поддерживается по аппаратным или еще каким-то причинам), модем должен отправить ответ длиной 0 байт (то есть отправить два нулевых байта). В остальном формат параметров команды и ответа определяется непосредственно командой.

В дальнейшем при описании команд конструкция \x00 обозначает нулевой байт.

Команды протокола версии 3:

- 0x64 (100) - получение текущих значений дополнительных входов модема. Параметров у команды нет (т.е. длина параметров равна 0). В ответ модем должен выдать строку вида Name1=Value1\x00Name2=Value2\x00 и т.д. Конкретные имена дополнительных входов определяются типом модема (равно как и возможные их значения).
- 0x65 (102) - получение архивных значений дополнительных входов модема. Параметры команды представляют собой строку вида T\x00YYYYMMDDHH. Здесь:
 - T - тип архива (один символ, 'h' - почасовой, 'd' - посуточный, 'i' - интегральный)
 - YYYY - год
 - MM - месяц
 - DD - день
 - HH - час (для посуточного архива должен быть нулём)

Стоит обратить внимание, что в запросе на архив, 2013050104 предполагаются данные за интервал 2013-05-01 04:00:00 - 04:59:59. Аналогичный принцип для посуточных архивов. Для интегральных архивов предполагается значение в момент 2013-05-01 04:00:00. Наличие/отсутствие каких-то архивов и глубина архивирования определяются моделью модема и опционально версией прошивки.
- 0x66 (101) - запись значений в управляемые выходы модема. Параметром команды служит строка, по формату аналогичная ответу модема на команду 100. В ответ модем посылает значения своих входов, как по команде 100 (для того, чтобы обеспечить более быструю обратную связь).
- 0x67 (103) - чтение текущего времени на модеме. В ответ модем выдает строку вида 'YYYY MM DD HH II SS' (числа допускаются без ведущих нулей), обозначающую время модема (AT+CCLK).
- 0x68 (104) - запись текущего времени на модеме. Параметром команды служит строка с датой-временем в формате MySQL (YYYY-MM-DD HH:II:SS). После успешной записи времени модем должен перезагрузиться

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 49

Протокол 1СИМ (модификация для LAN-устройств)

При обмене данными между сервером системы и Ethernet<->RS232 адаптером (LAN-устройством) используются в основном те же принципы, что и при работе с модемами. Здесь будут приведены особенности, характерные именно для ethernet-устройств и отличия от модемов.

Общее описание принципов работы устройства

Как и модем, LAN-устройство должно иметь возможность трансформировать доменные имена в IP-адреса при помощи DNS-серверов, получаемых автоматически по DHCP. LAN-устройство должно иметь возможность хранить адреса четырех доменных имен серверов вместе с портами в энергонезависимой памяти.

В отличие от модема LAN-устройство должно поддерживать работу как в режиме DHCP (в котором настройки TCP/IP получаются автоматически от сервера провайдера), так и в режиме PPPoE (в котором LAN-устройство должно инициировать PPP-сессию с сервером провайдера). Устройство должно позволять настройку требуемого режима работы, а также параметров PPPoE-сессии (имя пользователя и пароль, до 20 символов каждое). Настройки должны сохраняться в энергонезависимой памяти.

LAN-устройство должно поддерживать внутренний буфер для фонового приема данных, доступных на RS линии. Это требуется в первую очередь для RS232 моделей в режиме согласования, чтобы прибор не прекратил передачу данных, если ему покажется, что его "никто не слушает". Достаточно 4096 байт для этого буфера и он должен заполняться во время всех потенциально долгих операций. Для TCP-соединения требуется аналогичный буфер, размер которого достаточно ограничить 512 байтами.

Устройство должно поставляться как с RS232 портом, так и с RS485. В случае с RS232 требуется возможность организации аппаратного согласования RTS/CTS, а также управления сигналами на пинах DSR и DCD (см. описание команд протокола).

Цикл работы устройства

При запуске устройство в зависимости от выбранного режима (DHCP или PPPoE) получает настройки TCP/IP или инициирует PPP сессию (успешная инициализация которой завершается получением настроек TCP/IP). В случае неуспешного получения настроек/установки сессии устройство должно перезагрузиться, после чего повторять эту операцию.

Далее устройство пытается установить TCP соединение с одним из известных ему 4х серверов, пробуя их по кругу 0,1,2,3,0,1,2,3,... В случае 12 неуспешных TCP коннектов с определенными паузами и таймаутами - ребут.

Дальнейшее общение между устройством и сервером происходит аналогично общению модема (порядок отправки-получения пакетов и формат команд), со следующими отличиями:

- 1) Вместо IMEI GSM-чипа отправляется MAC-адрес адаптера. Требуется возможность просмотра MAC-адреса при конфигурации устройства (либо наличие наклейки с MAC-адресом на приборе).
- 2) Вместо CID SIM-карты отправляется строка, содержащая серийный номер устройства. При невозможности его получения — пустая строка.
- 3) В протоколе версии 2 допускается не реализовывать команду 0xFE, в таком случае версия прошивки устройства при инициализации не передается.

Наименование ПС:	ПЭВМ «Управление зданиями и территорией производственных площадок»
Код документа:	Стр. 50

Источники

Окружение для установки *Источник:* <https://wiki.1sim.online/index.php?oldid=852> *Редакторы:* Ftc

Установка на CentOS7 *Источник:* <https://wiki.1sim.online/index.php?oldid=1158> *Редакторы:* Ftc

Краткая инструкция по установке *Источник:*

<https://wiki.1sim.online/index.php?oldid=1309> *Редакторы:* Ftc

Создание погодных аккаунтов *Источник:*

<https://wiki.1sim.online/index.php?oldid=1487> *Редакторы:* Zurg

Обновление ПЭВМ «УЗИТ» *Источник:*

<https://wiki.1sim.online/index.php?oldid=1626> *Редакторы:* Ftc

Процедура обновления системы *Источник:* <https://wiki.1sim.online/index.php?oldid=1627> *Редакторы:* Baranovpl, Ftc

Структура каталогов ПЭВМ «УЗИТ» *Источник:* <https://wiki.1sim.online/index.php?oldid=862> *Редакторы:* Ftc

Сервисное API *Источник:* <https://wiki.1sim.online/index.php?oldid=1406> *Редакторы:* Ftc

Подсистема самодиагностики *Источник:*

<https://wiki.1sim.online/index.php?oldid=866> *Редакторы:* Ftc

Настройка модуля service-api *Источник:*

<https://wiki.1sim.online/index.php?oldid=1408> *Редакторы:* Ftc

Crontab и его генерация *Источник:*

<https://wiki.1sim.online/index.php?oldid=1297> *Редакторы:* Ftc

Архитектура системы сбора данных *Источник:*

<https://wiki.1sim.online/index.php?oldid=1288> *Редакторы:* Ftc

Общий принцип хранения данных в системе *Источник:* <https://wiki.1sim.online/index.php?oldid=1521> *Редакторы:* Ftc

Конфигурация хранилища данных *Источник:* <https://wiki.1sim.online/index.php?oldid=1529> *Редакторы:* Ftc

Команды для работы с хранилищем и его обслуживания *Источник:* <https://wiki.1sim.online/index.php?oldid=1531>

Редакторы: Ftc

Экспорт и импорт сущностей в системе *Источник:*

<https://wiki.1sim.online/index.php?oldid=1356> *Редакторы:* Ftc **Полезные**

утилиты в системе *Источник:*

<https://wiki.1sim.online/index.php?oldid=1545> *Редакторы:* Ftc, Zurg

Показатели системы для мониторинга *Источник:*

<https://wiki.1sim.online/index.php?oldid=1630> *Редакторы:* Ftc **Протокол**

1СИМ (модемный) *Источник:*

<https://wiki.1sim.online/index.php?oldid=1420> *Редакторы:* Ftc

Протокол 1СИМ (модификация для LAN-устройств) *Источник:* <https://wiki.1sim.online/index.php?oldid=311> *Редакторы:*

Ftc