

Руководство пользователя Системы мониторинга работы АВР

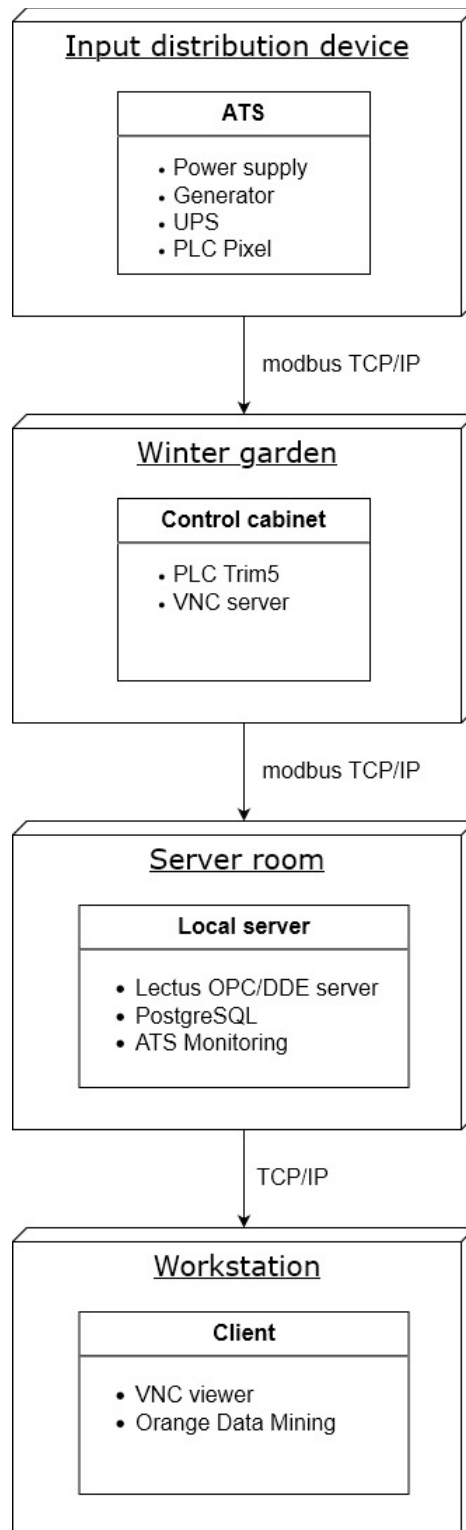
Объекты мониторинга:
Автоматический ввод резерва

г. Новосибирск
2021 г.

Содержание

Блок-схема.....	4
Контроллер.....	5
SMS уведомления.....	6
OPC сервер.....	7
База данных.....	8
Таблицы базы данных.....	9
Поставщик SMS шлюза.....	11
ATS Monitoring.....	12
Создание отчетов.....	13

Блок-схема



Контроллер

Контроллер Segnetics Pixel 25-xx-xx установлен в Вводно-распределительном устройстве (см. однолинейную схему ВРУ) и собирает данные о работе генератора и автоматического ввода резерва. Связь с ОПС сервером осуществляется посредством modbus TCP/IP.



На дисплее контроллера отображаются четыре строки:

- На первой строке отображается наличие/отсутствие питания от городской электросети.
- На второй строке отображается факт успешного старта генератора.
- На третьей строке отображается наличие/отсутствие сбоя старта генератора.
- На четвёртой строке отображается наличие/отсутствие неисправности работы генератора.

SMS уведомления

Мониторинг работоспособности генератора:

При неисправности генератора, на ваш номер приходит следующее SMS сообщение:

Авария! Генератор неисправен! Срочно произведите сервисные работы!

При восстановлении работоспособности генератора, на ваш номер приходит следующее SMS сообщение:

Работоспособность генератора в режиме трансляции питания от электросети восстановлена. Генератор исправен. Генератор работает.

Мониторинг работы автоматического ввода резерва:

При сбое питания от городской электросети и успешном старте генератора на ваш номер приходит следующее SMS сообщение:

Сбой питания от электросети. Успешный старт генератора.

При сбое питания от городской электросети и сбое старта генератора на ваш номер приходит следующее SMS сообщение:

Сбой питания от электросети. Сбой старта генератора.

При восстановлении питания от городской электросети и исправной работе генератора, на ваш номер приходит следующее SMS сообщение:

Питание от электросети восстановлено. Генератор исправен. Генератор работает.

При восстановлении питания от городской электросети и неисправности генератора, на ваш номер приходит следующее SMS сообщение:

Питание от электросети восстановлено. Генератор неисправен. Генератор не работает.

ОПС сервер

На локальном сервере работает Lectus OPC/DDE server, выполняя функцию сбора данных от контроллера и передачу их в СУБД PostgreSQL.

Установка и настройка Lectus OPC/DDE server

- Скачайте и установите [Lectus Modbus OPC/DDE server](#).
- Откройте файл конфигурации "modbus map for ats-monitoring.cfg".
- Создайте источник данных ODBC для СУБД PostgreSQL на основе [драйвера ODBC](#) для PostgreSQL (панель управления → администрирование → источники данных ODBC).
- Подключите внешнюю СУБД PostgreSQL в настройках ОПС сервера (настройка → параметры → разные → подключение к базе данных → настройка → в качестве провайдера выберите: Microsoft OLE DB Provider for ODBC Drivers).
- Запустите опрос переменных в ОПС сервере.

База данных

В качестве базы данных используется СУБД PostgreSQL, записывающая данные из контроллера посредством OPC сервера, и данные работы приложения мониторинга работы генератора, автоматического ввода резерва, отправки уведомлений - ATS Monitoring.

Установка и настройка PostgreSQL:

- Скачайте и установите [СУБД PostgreSQL](#), при установке выберите дополнительно установить pgAdmin и драйвера ODBC.
- Создайте базу данных.
- Настройте параметры доступа к базе данных.
- Настройте кодировку UTF8.
- Настройте использование кириллицы.
- Создайте следующие таблицы:

```
CREATE TABLE avr_control_insert (
  mains_power_supply int NOT NULL,
  start_generator int NOT NULL,
  generator_faulty int NOT NULL,
  generator_work int NOT NULL,
  connection int NOT NULL,
  mark timestamptz default current_timestamp
);
```

```
CREATE TABLE журнал_работы_приложения (
  событие text NOT NULL,
  время_и_дата timestamp default current_timestamp
);
```

```
CREATE TABLE события_авр (
  событие text NOT NULL,
  время_и_дата timestamp default current_timestamp
);
```


Таблицы базы данных

Таблица «avr_control_insert»

Служит для своевременной передачи данных из контроллера в приложение ATS Monitoring.

mains_power_supply	start_generator	generator_faulty	generator_work	connection	mark
1	0	0	1	1	2021-10-04 17:10:41.396786

- mains_power_supply - наличие питания от электросети (0 — нет, 1 — есть)
- start_generator - старт генератора (0 — сбой старта генератора, 1 — успешный старт генератора)
- generator_faulty — сбой работы генератора в режиме трансляции питания от электросети (0 — генератор исправен, 1 — генератор не исправен)
- generator_work — работа генератора (0 — питание на выходе генератора отсутствует, 1 — питание на выходе генератора присутствует)
- connection — наличие связи с контроллером (0 — нет связи, 1 — связь есть)
- mark — временная метка записи переменных (дата и время)

Таблица «журнал_работы_приложения»

Служит для ведения журнала приложения мониторинга работы генератора, автоматического ввода резерва, отправки уведомлений - ATS Monitoring.

событие	время_и_дата
Ошибка! Связь OPC сервера с ПЛК отсутствует!	2021-10-04 17:10:41.396786

Производится запись следующих событий:

- Авария! Генератор неисправен! Срочно произведите сервисные работы!
- Генератор в режиме трансляции питания от электросети работает исправно.
- Работоспособность генератора в режиме трансляции питания от электросети восстановлена.
- Отправлено SMS сообщение: /Авария! Генератор неисправен! Срочно произведите сервисные работы!/ на номер +79xx-xxx-xx-xx
- Отправлено SMS сообщение: /Работоспособность генератора в режиме трансляции питания от электросети восстановлена./ на номер +79xx-xxx-xx-xx
- Server error! Ошибка! SMS уведомление не было отправлено!
- Http request status error! Ошибка! SMS уведомление не было отправлено!
- Ошибка! Доступ к интернету отсутствует! Http запрос не был выполнен! SMS уведомление не было отправлено!
- Ошибка! Связь OPC сервера с ПЛК отсутствует!
- Ошибка! Связь СУБД PostgreSQL с OPC сервером отсутствует!

- Произошел сбой питания от электросети! Ожидание (90 секунд) подтверждения отсутствия питания от электросети.
- Питание от электросети есть.
- Подтверждение отсутствия питания от электросети.
- Успешный старт генератора.
- Сбой старта генератора!
- Питание от электросети восстановлено.
- Генератор исправен. Генератор работает.
- Генератор неисправен. Генератор не работает.
- Питание от электросети еще не было восстановлено, после отключения.
- Отправлено SMS сообщение: /Сбой питания от электросети. Успешный старт генератора./ на номер +79xx-xxx-xx-xx
- Отправлено SMS сообщение: /Сбой питания от электросети. Сбой старта генератора./ на номер +79xx-xxx-xx-xx
- Отправлено SMS сообщение: /Питание от электросети восстановлено. Генератор исправен. Генератор работает./ на номер +79xx-xxx-xx-xx
- Отправлено SMS сообщение: /Питание от электросети восстановлено. Генератор неисправен. Генератор не работает./ на номер +79xx-xxx-xx-xx

Таблица «события_авр»

Служит для записи событий работы автоматического ввода резерва.

событие	время_и_дата
Сбой питания от электросети. Успешный старт генератора.	2021-10-04 17:10:41.396786

- событие — одно из событий работы приложения
- время_и_дата — временная метка записи события

Производится запись следующих событий:

- Авария! Генератор неисправен! Срочно произведите сервисные работы!
- Работоспособность генератора восстановлена. Генератор исправен. Генератор работает.
- Генератор в режиме трансляции питания от электросети работает исправно.
- Питание от электросети есть.
- Сбой питания от электросети. Успешный старт генератора.
- Сбой питания от электросети. Сбой старта генератора.
- Питание от электросети восстановлено. Генератор исправен. Генератор работает.
- Питание от электросети восстановлено. Генератор неисправен. Генератор не работает.
- Авария! Генератор неисправен! Срочно произведите сервисные работы!
- Работоспособность генератора восстановлена. Генератор исправен. Генератор работает.
- Генератор в режиме трансляции питания от электросети работает исправно.

Поставщик SMS шлюза

Настройка SMS шлюза:

- Создайте учетную запись на сайте поставщика услуг [ClickSend](#).
- Пополните баланс.
- Настройте API для SMS.

Настройка API для SMS:

Используйте в приложении ATS Monitoring http get запросы:

`https://api-mapper.clicksend.com/http/v2/send.php?method=http&username=xxxx&key=xxxx&to=xxxx,yyyy,zzzz&message=xxxx`

Параметр	Обязательность	Описание
username	да	Ваше имя пользователя API.
key	да	Вы можете найти это в своей учетной записи в разделе «Учетные данные API» в верхней части экрана.
to	да	Мобильный номер получателя в международном формате (с начальным + и кодом страны). Разделите несколько получателей запятой (,), если применимо. Максимум 1000 получателей.
message	да	Сообщение для отправки. Максимум 960 символов.
senderid	нет	Пользовательский идентификатор отправителя (от имени / номера).
schedule	нет	Позволяет запланировать доставку сообщений. Должен быть в формате unix.
country	нет	Автоматически преобразует предоставленный вами местный номер для добавления кода страны.

Например:

```
let resp = request::blocking::get(
    "https://api-mapper.clicksend.com/http/v2/send.php?method=http&
    username=development-service@yandex.ru&key=1E82A334-89D8-985C-526B-712DB70A713D&
    to=+79139402913&message=Сбой+питания+от+электросети.
    +Успешный+старт+генератора." ).unwrap();
```

ATS Monitoring

Прикладное приложение ATS Monitoring предназначено для мониторинга работы генератора и автоматического ввода резерва, записи соответствующих событий в базу данных и отправки SMS уведомлений.

Приложение ATS Monitoring написано на языке программирования Rust. Исходные файлы доступны на [github](#).

Установка

- Установите Rust для Linux или macOS:

```
$ curl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh
```

- Для Windows посетите [эту страницу](#).
- Установите приложение:

```
$ cargo install ats-monitoring
```

Установка и настройка при необходимости корректировать http get запросы и строки конфигурации подключения к PostgreSQL

- Установите Rust для Linux или macOS:

```
$ curl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh
```

- Для Windows посетите [эту страницу](#).
- Клонировать репозиторий.
- Отредактируйте http get запросы:

```
let resp = request::blocking::get("https://api-mapper.clicksend.com/http/v2/send.php?method=http&username=development-service@yandex.ru&key=1E82A334-89D8-985C-526B-712DB70A713D&to=+79139402913&message=Сбой+питания+от+электросети.+Успешный+старт+генератора.").unwrap();
```

- Отредактируйте строки конфигурации подключения к PostgreSQL:

```
let mut client = Client::connect("postgresql://postgres:postgres@localhost/postgres", NoTls)?;
```

- Скомпилируйте локальные пакеты и все их зависимости:

```
$ cargo build --release
```

Создание отчетов

Для анализа и визуализации данных используйте Orange Data Mining.

Установка и настройка

- Скачайте и установите [Orange Data Mining](#).
- Настройте [чтение данных из БД](#).
- Установите дополнительный виджет для анализа данных «Time Series» (Options → Add-ons).

Отчет в формате «Таблица данных»

- Создайте новый проект в Orange Data Mining.
- Добавьте виджет Data → SQL. В настройках виджета снимите галочку Auto-discover categorical variables.
- Добавьте виджет Time Series → As Timeseries.
- Добавьте виджет Data → Select Columns.
- Добавьте виджет Data → Select Rows.
- Добавьте виджет Data → Data Table.
- Добавьте виджет Data → Save Data.
- Отсортируйте данные и сохраните отчет в удобном для вас формате.

Отчет в формате «График»

- Создайте новый проект в Orange Data Mining.
- Добавьте виджет Data → SQL.
- В настройках виджета SQL снимите галочку Auto-discover categorical variables.
- Добавьте виджет Time Series → As Timeseries.
- Добавьте виджет Data → Select Columns.
- Добавьте виджет Data → Select Rows.
- Добавьте виджет Data → Live Chart.
- Отсортируйте данные и сохраните отчет в удобном для вас формате.