



## **ТехАвтоматика**

Общество с ограниченной ответственностью  
658204, Алтайский край, г. Рубцовск, ул. Азовская, 34, тел/факс: (38557) 2-55-15, 8-913-089-8593,  
E-mail: [ta@ngs.ru](mailto:ta@ngs.ru), URL: <http://www.ta22.ru>

---

# **Руководство системного программиста на систему управления и сбора данных**

Рубцовск 2016 г.

## Оглавление

Общие сведения.....	3
Структура системы.....	4
Список файлов и их описание.....	5
Список TCL-скриптов.....	5
Список PHP-скриптов.....	6
Список таблиц БД:.....	7
Структура таблицы `values_spr`.....	7
Структура таблицы `categories_spr`.....	7
Структура таблицы `events` и `operations`.....	7
Структура таблиц данных датчиков.....	8
Описание конфигурационных массивов.....	8
Настройка и проверка.....	12
Резервирование и восстановление.....	13
Список файлов находящихся на Flash-диске.....	13
Создание и использование загрузочного диска.....	14
Пароли и их замена.....	16

## Общие сведения

Программный комплекс предназначен для сбора данных и выдачи команд управления в технологических процессах. Предоставляет интерфейс пользователя, позволяющий управлять пуском и остановом оборудования в составе маршрутов движения продукта. Так же программа обеспечивает отображение состояния механизмов и датчиков, ведение журнала аварийных ситуаций и действий оператора, сбор и хранение показаний технологических датчиков, предоставляя доступ к данным посредством WEB-интерфейса. Работа осуществляется под управлением операционной системы Linux с использованием следующих компонентов:

- Сервер баз данных MySQL — используется для хранения показаний измерительной аппаратуры, счётчиков и служебной информации (состояние связи, журнал работы). По умолчанию разрешён только локальный доступ, для программы и для системы отчётов. Удалённый доступ — по необходимости.
- Сервер удалённого доступа SSH — предоставляет консоль управления для административных целей.
- WEB сервер Lighttpd и сценарный язык PHP — необходимы для функционирования подсистемы отчётов. Lighttpd предварительно настраивается для использования PHP который, в свою очередь, на использование библиотек mysqli, gd, json.
- Утилита phpMyAdmin — используется как административный интерфейс к локальной базе данных, предоставляя доступ посредством WEB-интерфейса.
- Интерпретируемый язык Tcl/Tk — необходимая среда для исполнения программы управления (Дополнения — modbusctl, libimg, ttk, tcllib).
- Опционально возможна установка сервера удалённого доступа x11vnc — для удалённого мониторинга (по требованию — управления) работы системы.
- Произвольный современный WEB-браузер для просмотра отчётов.
- Графический сервер Xorg.
- Оконный менеджер OpenBox.

Программа предоставляет интерфейс пользователя для управления и мониторинга технологического процесса, считывает данные с измерителей и регуляторов, по необходимости осуществляет обмен данными с управляющими программируемыми логическими контроллерами (далее ПЛК), модулями дискретного и аналогового ввода-вывода. Связь осуществляется посредством сети RS-485 через преобразователь RS-Ethernet (например MOXA NPort). Непосредственно для связи программа использует сеть Ethernet.

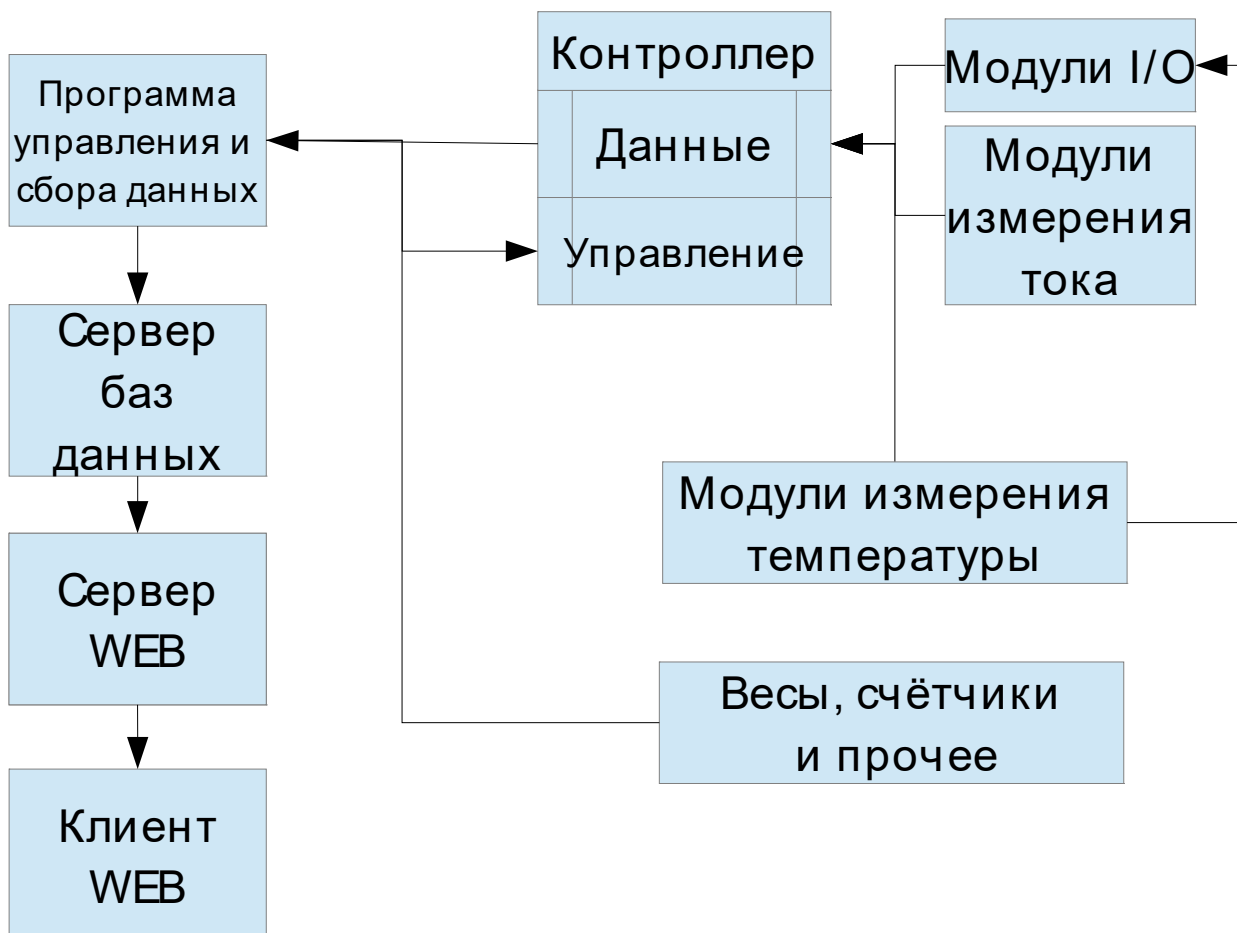
Поддерживаются следующие протоколы обмена: ModBusRTU, ModBusTCP, ModBusASCII, DCON, ЦЕНТА. По необходимости возможно добавить поддержку дополнительного протокола.

# Структура системы

Программная часть системы состоит из:

- Программа мониторинга и сбора данных о ходе технологического процесса:
  - Основной модуль (он же реализует интерфейс пользователя).
  - Модуль синхронного опроса устройств. Запускается по одному экземпляру на порт.
  - Модуль отправки архивных данных на SQL сервер.
  - Модуль реализации технологии и автоуправления
  - Конфигурационные файлы системы.
  - Модули протоколов связи.
  - Модули диалогов настройки параметров системы.
  - Модули обработки расписания и сохранения данных.
- Подсистема отчётов.
  - Модуль построения графиков.
  - Календарный модуль.
  - Модули формирования запросов к БД и выдачи результатов.
- База данных для данного сервера.
  - Таблицы хранения показаний датчиков.
  - Таблицы хранения времени работы механизмов.
  - Служебные таблицы (журналы, справочники, состояние связи).

Примерная схема системы приведена на последующем рисунке:



# Список файлов и их описание

## Список TCL-скриптов

Программа располагается в папке /home/WORK/program/:

- `conf/` — директория с файлами конфигурации системы. Любой файл с именем вида `config*.tcl` будет загружен автоматически при запуске программы, остальные проигнорированы (если не были указаны непосредственно в одном из конфигурационных файлов). Базовые файлы конфигурации:
  - `conf/configIO.tcl` — файл конфигурации каналов ввода — вывода и основных структур данных программы, содержит массивы соответствия каналов ввода — вывода технологическим элементам, список элементов для инверсии показаний датчиков, соответствия считываемых параметров таблицам в базе данных, вспомогательные массивы. Основной конфигурационный файл системы. Массивы конфигурации каналов могут быть вынесены в отдельный файл (например `IOCHANS.tcl`)
  - `conf/configDEV.tcl` — файл конфигурации параметров связи с устройствами, общий для всех портов обмена. Инициализация структур данных для обмена с устройствами. Ассоциативные массивы (словари) описания шаблонов устройств и конфигурации сети могут быть вынесены в отдельные файл (`DEVTYPES.tcl CONNECTIONS_DEVS.tcl`)
  - `conf/configLINES.tcl` — файл конфигурации условий для вывода линий движения продукта на мнемосхеме.
  - `conf/configTM.tcl` — файл конфигурации технологических маршрутов: описание начальных и конечных точек, точек ветвления, таблица связности графа механизмов, описание аспираций и обслуживаемых ими механизмов, условия для блокировок работы и останова по переполнению конечных точек.
  - `conf/configUST.tcl` — файл конфигурационных массивов задержек на запуск-останов устройств при автоматическом запуске (по маршрутам), массив задержек на срабатывание датчиков и массив уставок измеряемых числовых величин. Заполняется изначально автоматически, редактируется интерфейсом настройки в программе. Программа должна иметь право на запись в этот файл
  - `conf/configGLYPHS.tcl` — файл конфигурационных массивов с координатами и описанием элементов, выводимых на мнемосхему (мнемонические изображения датчиков, кнопок выдачи управляющих сигналов, индикаторы числовых величин, линии движения продукта). Заполняется автоматически, модифицируется интерфейсом настройки в особом режиме запуска.
- `img/` — директория с файлами глифов, отображаемых на мнемосхеме.
- `inc/` — директория с файлами модулей реализации технологической программы и автоуправления, обработки маршрутов и связанной с ними логики, сигналов датчиков промбезопасности и прочее. Здесь же обработка ошибок связи, преобразование значений регистров в состояния каналов ввода — вывода, вывод сообщений оператору и формирование архивных данных. Любой файл с именем вида `inc*.tcl` будет загружен автоматически при запуске программы, остальные проигнорированы. Список базовых модулей:
  - `inc/incUtils.tcl` — функции записи отчётов, сохранения данных в базу, ведения журнала, итераторы и прочие вспомогательные.
  - `inc/incIFACE.tcl` — функции визуализации данных, вывода диалоговых окон.
  - `inc/incIO.tcl` — функции ввода-вывода.
  - `inc/incTECH.tcl` — модуль с технологической подпрограммой, функции обработки

логики аварий, индикации.

- lib/ — вспомогательные модули: низкоуровневая реализация протокола ModBus, подсчет контрольных сумм, графики, сетевая отладочная консоль, различные общеупотребимые функции.
- proto/ — модули высокоуровневой части ModBus, протоколов связи с различными устройствами. Модуль опроса устройств загружает все tcl файлы отсюда при запуске.
  - proto/incModBus.tcl — модуль связи в блоках ТРМ, измерения тока, контроллером, работающими по протоколу ModBus.
  - proto/incADK.tcl — модуль связи в блоках ЦЕНТА.
- tmp/ — директория временных файлов (кэш запросов к базе, ошибок выполнения запросов) и журнала системы, дублируемого в базе данных. Программа должна иметь права на запись в данный каталог.
- main.tcl — основной блок программы, инициализация данных, запуск процессов обмена с устройствами и базой, основных циклов программы.
- manual.tcl — форматирование и вывод краткой инструкции на экран.
- exchange.tcl — модуль ввода — вывода, обмена данными с устройствами. Запускается по экземпляру на порт из основной программы, опрашивает все устройства, подключённые к данному порту и прописанные в конфигурации, и возвращает в основную программу.
- iface.tcl — модуль, реализующий основной интерфейс программы.
- sql.tcl — модуль обмена с базой данных, вызывается из основной программы. Периодически отправляет накопившиеся в кеше запросы в базу, при неудаче пытается провести восстановление базы и повтор запроса, при повторной неудаче сохраняет запрос в файл ошибок.
- sqldbrepair — сценарий, вызываемый модулем обмена с базой данных при ошибке выполнения запроса.
- glyphconfig.tcl — модуль интерфейса расстановки элементов мнемосхемы, используется в особом режиме запуска программы.
- mnemoshema.png — фоновое изображение мнемосхемы.
- netstate — файл счётчиков ошибок связи с устройствами, обнуляется при успешном считывании данных. Заполнение включается по необходимости
- netwatch.sh — сценарий для просмотра администратором файла netstate. Периодически выводит его содержимое.
- start.sh — «ручной» запуск программы
- stop.sh — «ручной» останов программы.

## **Список PHP-скриптов**

Корень WEB-сервера располагается в директории /srv/http/

- index.html – набор фреймов заглавной страницы
- banner.html – верхний фрейм с заголовком страницы
- menu.php – левый фрейм с меню выбора категории датчиков, даты-времени начала и конца периода выборки данных для отчётов, интервала для отчётов по показаниям датчиков и кнопкой, передающей введённые данные для дальнейшего формирования отчёта (в скрипте select.php). Последними расположены кнопка вывода в отдельном окне браузера журнала работы системы за выбранный период и журнала действий оператора.

- `select.php` – правый фрейм после нажатия кнопки “отправить” в меню. Предоставляет интерфейс выбора датчиков заданной категории для отчёта, информацию о выбранных ранее параметрах, выбор типа отчёта. Имеется кнопка возврата фрейма на `info.php`
- `info.php` – начальный правый фрейм, где с периодом в несколько секунд выводятся последние несколько записей в журнале работы системы
- `report.php` – скрипт вывода отчёта о показаниях выбранных датчиков за указанный промежуток времени с установленным интервалом между измерениями.
- `geraccumt.php` – скрипт вывода отчёта о накоплении показаний датчиков за период. Разность показаний выбранных датчиков в конце и начале выбранного периода. Имеет смысл только для показаний накопительных величин (общий вес, количество доз и т.д.).
- `graph.php` – скрипт построения графика величин (параметры аналогичны используемым в отчёте о показаниях датчиков).
- `journal.php` – скрипт вывода журнала работы системы за выбранный промежуток времени.
- `operator.php` – скрипт вывода журнала действий оператора за выбранный промежуток времени.
- `mysqlparam.php` – параметры связи с сервером баз данных
- `calendar` – модуль вывода календаря и диалога выбора даты
- `pChart` – модуль построения графиков

### **Список таблиц БД:**

- `values_spr` — справочник датчиков
- `categories_spr` — справочник категорий датчиков
- `events` — журнал работы
- `operations` — журнал действий оператора
- `NAME№` - данные измерителя номер №

Для всех таблиц используется тип MyISAM, кодировка текстовых данных — utf8.

### **Структура таблицы `values\_spr`**

``textname` varchar(64) NOT NULL` — наименование счётчика

``tablename` varchar(64) NOT NULL` — имя таблицы, в которой хранятся значения данного счётчика

``category` int(10) NOT NULL` — соответствие датчика категории.

``neterr` int(11) NOT NULL DEFAULT '0'` — счётчик ошибок связи (зарезервирован)

### **Структура таблицы `categories\_spr`**

``textname` varchar(64) NOT NULL` — наименование категории

``nmbr` int(10) NOT NULL` — номер категории для сопоставления счётчикам

### **Структура таблицы `events` и `operations`**

``dtwhen` datetime NOT NULL` — текстовые дата и время события

``srcname` varchar(64) NOT NULL` — источник события (система, датчик, сеть)

``msg` varchar(256) NOT NULL` — текст сообщения

``errtype` varchar(64) NOT NULL` — тип сообщения (ошибка, оповещение)

## Структура таблиц данных датчиков

`dtwhen` datetime NOT NULL — текстовые дата и время записи значения

`value` int(10) unsigned NOT NULL — значение датчика

## Описание конфигурационных массивов

Конфигурация программы размещена в файлах с именами вида **config\*.tcl** из директории `conf`, где \* - произвольная последовательность символов. Также программа автоматически подключает при старте файлы в директории `inc` с именами вида **inc\*.tcl** в качестве дополнительных модулей, для расширения функциональности.

Конфигурация состоит из следующих ассоциативных массивов TCL и списков:

- **confib, confob, confin, confon** — конфигурационные массивы соответствия входных и выходных каналов опрашиваемых устройств дискретных (ib, ob) и числовых (in, on) данных в программе
- **dib, dob, din, don** — массивы для хранения соответствующего вида данных
- **devs** — конфигурационный массив устройств, с которыми ведётся обмен данными
- **databuf** — массив буферов приёма-передачи опрашиваемых устройств
- **dibtime** — массив таймеров срабатывания дискретных датчиков
- **ust** — массив уставок числовых величин
- **sqltables** — массив соответствия числовых датчиков таблицам в базе
- **connections** — настройки соединений для устройств с синхронным опросом.
- **main\_conf** — общесистемные настройки, временные характеристики системных циклов, параметры связи с базой данных
- **color** — палитра цветов
- **glyphs** — массив описания дискретных элементов, выводимых на мнемосхему
- **numbers** — массив описания цифровых элементов, выводимых на мнемосхему
- **indconf** — массив описания дополнительных индикаторов для цифровых элементов
- **lines** — массив координат узлов линий движения продукта
- **graphic** - конфигурационные массив фрейма вывода графиков

Общий формат ассоциативного массива:

```
array set имя_массива {  
    имя_элемента значение  
    имя_элемента {значение значение значение}  
    имя_элемента {{значение значение} значение  
    .....  
    «значение значение» значение}  
}
```

Технологические элементы обозначаются в программе следующим образом:

**тип.№\_назначение** , где

- **тип** — установленное наименование класса элементов, к которому относится данный
- **№** — порядковый номер, соответствующий принятому на технологической схеме



- назначение — функциональное назначение элемента, соответственно каналу ввода-вывода

Например nr1\_ar — нория 1, сигнал аварии по подпору, vent3\_p — сигнал запуска вентилятора №3.

Обозначения элементов, принятые в программе:

- Классы — nr, shn, tr, shl, z, vent, kp, vu, gor, b. Соответствуют нориям, шнекам, транспортёрам, шлюзовым затворам, задвижкам, вентиляторам, перекидным клапанам, выпускным устройствам, горелке, бункерам.
- Назначения — p, w, s, kf, af, ar, ap, as, dnu, dsu, dvu, o, z, ao, az. Соответствуют сигналу пуска, работы, останова элемента, контролю фазы, сигналу аварии по датчику фазы, скорости, подпора, схода ленты, датчику нижнего, среднего, верхнего уровней, открытого, закрытого положения задвижки, аварий открытия, закрытия задвижки

Подробное текстовое описание элемента содержится в самом конфигурационном массиве.

Массивы вида **имя значение** достаточно тривиальны и не заслуживают описания.

### Пример массива настроек соединений для синхронного опроса:

```
array set connections {
1 {
  name MOXA1p1
  address 192.168.0.201
  port 4001
}
}
```

где:

- **name** — условное обозначение соединения
- **adress** — ip адрес устройства связи
- **port** — порт

### Массив общих настроек:

```
array set main_conf {
  cycle 50
  store_sql_delta 1000
  netstate_file "netstate"
  dbuser "user"
  dbpassword "uplift"
  dbank "buckwheat"
  dbhost "localhost"
}
```

где:

- **cycle** — задержка (мс) между итерациями основного цикла синхронного опроса.
- **store\_sql\_delta** — период (мс) сброса накопленных данных в базу
- **netstate\_file** — имя (и путь) файла состояния связи для административной консоли
- **dbuser** — имя пользователя, имеющего доступ к базе данных
- **dbpassword** — его пароль. При изменении администратором на стороне базы, необходимо привести в соответствие
- **dbank** — имя базы данных
- **dbhost** — адрес сервера баз данных

## Участок массивов параметров синхронно опрашиваемых устройств:

```
array set devs {
  1 {
    connection 1
    netnum 1
    timeout 10
    errmax 3
    rfunc MODBUSint16_get
    type OWEN
    numvals 27
    inreg 0x0000
    wfunc MODBUSint16_set
    outvals 1
    outreg 0x000F
    name "Контроллер OWEN"
  }
  2 {
    connection 1
    netnum 21
    timeout 10
    errmax 3
    rfunc TPM_get
    type TPM
    numvals 2
    inreg 0x0001
    name "TPM202 Зона нагрева 1"
  }
}
```

### Разрешённые пары **параметр значение**:

- **connection** — номер соединения (индекс элемента из массива конфигурации соединений)
- **netnum** — сетевой номер устройства в сети RS485
- **timeout** — задержка на ожидание ответа от устройства
- **errmax** — количество подряд идущих попыток считывания данных с устройства, достаточных для установления факта потери связи
- **rfunc** — имя функции считывания данных
- **type** — тип устройства
- **numvals** — количество считываемых данных (регистров)
- **inreg** — шестнадцатеричный адрес начального регистра считываемых данных в устройстве, начиная от него считывается **numvals** регистров
- **wfunc** — имя функции записи данных
- **outvals** — количество записываемых регистров
- **outreg** — шестнадцатеричный начальный адрес записываемых регистров
- **name** — текстовое имя счётчика, для отчётов и прочих выходных данных

### Формат записей в массивах конфигурации каналов ввода-вывода контроллеров (**confib** и **confob**):

**индекс\_имя** {«№контроллера №регистра №бита» «Текстовое наименование»}

где номер контроллера — индекс устройства из массива **devs**, номер регистра — порядковый номер (начиная с 1) регистра в устройстве, начиная с адреса из **inreg** или **outreg**, номер бита в регистре, состояние которого соответствует элементу в программе с именем «индекс\_имя».

Пример для ввода: `kp25_p1 {"1 10 13" "Клапан №25 положение 1"}`

где элементу программы с именем `kp25_p1` соответствует значение 13го бита 10го относительно **inreg** двухбайтового регистра устройства с порядковым номером 1

Пример для вывода: `ges_p {"1 15 1" "Пуск рециркуляции"}`

где значение элемента программы с именем `ges_p` соответствует 1му биту 15го относительно **outreg** двухбайтового регистра устройства с порядковым номером 1

**Формат массива конфигурации выводимых на мнемосхему дискретных элементов:**

Индекс\_имя {Имя\_файла\_изображения Цвет\_вкл Цвет\_выкл КоордX КоордY}

пример: `kompr87_dd {dd.xbm 0 5 3699.0 717.0}`

Файлы с изображениями датчиков, выводимые на мнемосхеме должны быть в формате XBM.

Цвета указываются по индексам из массива `color`. Координаты задаются относительно верхнего левого угла (0;0) фонового изображения мнемосхемы.

**Формат массива конфигурации выводимых на мнемосхему числовых элементов:**

Индекс\_имя {Подпись\_величины Цвет\_норма Цвет\_превышения КоордX КоордY}

пример: `trm30_2 {% green red 1140.0 284.0}`

Цвета указываются непосредственно. Подпись соответствует единице измерения указанной величины.

**Формат массива конфигурации дополнительных индикаторов для цифровых элементов:**

Индекс\_имя {Команда\_отрисовки Параметры\_команды Уникальный\_идентификатор X Y}

пример: `du411 {{tk::progressbar} {-orient vertical -length 140} idu411 56.0 610.0}`

Идентификатор необходим для работы интерфейсного модуля и не должен совпадать с именем цифрового элемента, поскольку оно используется для отрисовки числового поля.

**Формат массива координат узлов линий направления движения продукта:**

Идентификатор {X1 Y1 X2 Y2 .....Xn Yn}

должен содержать по меньшей мере 2 пары координат, для начальной и конечной точек.

**Формат массива фрэйма вывода графиков:**

`array set graphic {`

`1 180 #количество точек графика, длина`

`every 60000 #период перерисовки в миллисекундах`

`miny -20 #минимальная выводимая ордината`

`maxy 180 #максимальная выводимая ордината`

`graphs {tzn1_1 tzn1_2 tzn2_1 tzn2_2 tzo tas} #имена элементов — источников данных`

`colors {green white yellow black cyan red} #соответствующие им цвета`

`names {"T1 зоны1" "T2 зоны1" "T1 зоны2" "T2 зоны2" "Т охл." "Т агента"} #и подписи легенды`

`}`

## Настройка и проверка

В пусконаладочный период необходимо, используя документацию на соответствующие устройства, произвести следующие действия:

- Настроить устройства (параметры связи, режимы работы). Необходимое условие для устройств на последовательной линии — одинаковые параметры связи для всех устройств на одной ветке сети RS-485. Сетевые номера и адреса устройств должны соответствовать указанным в конфигурации.
- Соответственным образом настроить порты MOXA NPort — параметры связи порта должны совпадать с таковыми на устройствах ветки сети RS485 подключённой к данному порту.
- Настроить параметры сети Ethernet преобразователя NPort и управляющего компьютера. Они должны находиться в одной подсети.
- Проверить параметры связи (адреса преобразователей, сетевые номера подключённых к ним устройств, и прочих) и, по необходимости, перенастроить или внести изменения в конфигурацию программы.

При отсутствии физических проблем со связью, правильно настроенная программа должна выдавать в журнал работы только служебные сообщения и выводить в файл состояния сети нулевые значения счётчиков ошибок связи.

В случае отсутствия ответа от устройства, в журнал системы выводится сообщения вида:

**2012-02-20 15:22:48 | Контроллер ПЛК110 ОВЕН | Не удалось опросить устройство №1**

где указанный номер соответствует записанному в конфигурации сетевому номеру устройства.

В случае ошибки связи с преобразователем протоколов в журнал системы выводится сообщение вида:

**2012-02-20 15:28:39 | MOXA1p2 | Ошибка соединения с 192.168.0.254:4002**

где указывается IP адрес и порт преобразователя.

При этом соответствующий счётчик в файле состояния связи растёт. Если подобного не происходит а сообщения периодически возникают, это означает низкое качество связи (большое количество помех).

Необходимо учитывать, что при полном отсутствии связи с Ethernet устройствами возникают длительные задержки в отображении интерфейса системы, необходимо переключиться на журнал системы и дождаться вывода сообщения о причинах. Как правило данная ситуация возникает либо при проблемах со связью между управляющим компьютером и концентратором Ethernet, либо при выключении концентратора.

В случае вывода отдельного окна с сообщением об ошибке, необходимо обратиться к разработчику.

## Резервирование и восстановление

Система предусматривает возможность автоматического ежедневного создания резервной копии базы данных на внешний сетевой ресурс (при его наличии). По умолчанию данные сохраняются в виде сжатого SQL файла (datastorage.ARM.sql.gz) в папке /home/PUBLIC/Bdbbackup/ доступной так же по адресу //IP/PUBLIC, где IP — адрес системы (по умолчанию 192.168.0.66 и 192.168.127.66 для основного интерфейса и автоматическое получение для дополнительного при его наличии).

Система не требовательна к аппаратным ресурсам и адаптирована для работы на широком диапазоне аппаратных платформ. Поэтому, при выходе из строя компьютера, в случае работоспособного жёсткого диска, для восстановления работоспособности системы достаточно подключить старый жёсткий диск в качестве загрузочного к новому компьютеру. Желательно использовать аппаратную платформу не новее используемой ранее.

Для резервирования системы в целом, используется специально сформированный загрузочный Flash-диск, используемый как для создания полного образа системы, так и для восстановления (переноса на другой компьютер).

Для создания диска восстановления подойдёт любой flash-диск, ограничение только по объёму. Поскольку начальная резервная копия занимает около 1Гб, необходимо использовать диск  $\geq 2$ Гб.

Для переноса системы на новую аппаратную платформу (в случае неработоспособности прежней с повреждением жёсткого диска), необходим x86 компьютер с жёстким диском не менее 250Гб. При использовании только стандартного оборудования никакой дополнительной настройки (драйверов, таблицы разделов) не требуется. Необходимое условие — системный жёсткий диск должен определяться в компьютере первым (в случае IDE — как Primary master). **Внимание: операция восстановления системы уничтожит всю имеющуюся информацию на первом жёстком диске компьютера!**

### Список файлов находящихся на Flash-диске

- Arch/boot — директория загрузчика, и всего что к нему относится
- Arch/boot/bootinst.bat — командный файл установки загрузчика (из под Windows)
- Arch/boot/bootinst.sh — скрипт установки загрузчика (из под Linux)
- Arch/boot/syslinux/syslinux.cfg — конфигурация загрузчика
- Arch/x86\_64 — директории установочных образов системы Arch linux
- recover — содержит скрипты резервирования и восстановления системы
- recover/makebackup — скрипт создания архива восстановления
- recover/recover — скрипт восстановления системы из архива
- recover/arch.tgz — архив восстановления системы. Создаётся скриптом makebackup
- recover/archprev.tgz — предыдущий архив восстановления системы. Резерв для случая неверно созданного основного архива (например по ошибке создали с нерабочей системы)
- recover/disk\_prep.sh — вспомогательный скрипт. Выполняет разбиение и форматирование разделов целевого жёсткого диска
- recover/grub\_init.sh — вспомогательный скрипт. Выполняет установку загрузчика целевой системы
- recover/sys\_init.sh — вспомогательный скрипт. Выполняет минимально необходимую предварительную настройку распакованной системы.
- recover/mountsrc.sh - вспомогательный скрипт. Выполняет подключение целевых файловых систем, как для создания архива так и для восстановления из него
- recover/write\_img.sh — вспомогательный скрипт. Выполняет распаковку архива на целевую систему

Остальные файлы на диске относятся к системе установки Arch Linux, но необходимы для работы загрузочного диска.

## Создание и использование загрузочного диска

Использование диска восстановления:

- подключить flash-диск восстановления к целевому компьютеру
- настроить целевой компьютер для загрузки с USB
- дождаться появления меню загрузчика системы восстановления.
- в меню загрузчика выбрать соответствующий пункт, Backup — для создания резервной копии системы, Recover — для восстановления из архива на этом диске. Пункт Backup использовать только на заведомо работающей системе, поскольку текущая копия системы, для пункта Recover, будет заменена.
- Дождаться завершения операции. По завершению операции компьютер будет автоматически выключен.
- отключить flash-диск восстановления
- настроить целевой компьютер для загрузки первого жёсткого диска системы.
- Загрузить систему и проверить работоспособность.

Для создания дополнительной копии загрузочного диска восстановления, или переноса на другой flash-диск, необходимо:

- отформатировать подходящий flash-диск с указанием метки диска «SYSRESQ». Или установить данную метку на чистом диске. Данная метка **обязательна**, если изменена, то необходимо редактировать конфигурационный файл загрузчика.
- скопировать на него содержимое папки SYSRESQ дистрибутивного комплекта системы, или содержимое уже имеющегося диска восстановления.
- запустить с диска (**обязательно**) arch/boot/bootinst.bat или arch/boot/bootinst.sh (в зависимости от операционной системы, на которой производится данная операция, \*.bat для Windows, \*.sh для Linux) для установки загрузчика на flash-диск. Скрипты заменяют загрузчик диска, с которого запущены, поэтому выполнять исключительно на flash-диске.

Для ручного использования скриптов резервирования и восстановления:

- загрузиться с диска восстановления выбрав пункт меню загрузчика Boot Arch Linux
- В случае запроса имени пользователя «**login:**» ввести **root** (без пароля)
- выполнить переход в директорию скриптов восстановления командой **cd /run/archiso/bootmnt/recover**
- для создания архива восстановления выполнить команду **./makebackup** (точка и дробь перед именем команды необходимы для запуска из текущей директории). Команда создаст (или перезапишет) в текущей директории файл архива arch.tgz По завершении можно перезагрузить или выключить компьютер командами **reboot** или **halt** соответственно.
- для восстановления системы из архива выполнить команду **./recover** (точка и дробь перед именем команды необходимы для запуска из текущей директории). Команда создаст на первом жёстком диске целевого компьютера необходимые разделы, отформатирует их, распакует в них архив восстановления и установит загрузчик. По завершении перезагружаем компьютер командой **reboot**
- восстанавливаем на целевом компьютере загрузку с первого жёсткого диска
- проверяем работоспособность.

После восстановления необходимо загрузить актуальные данные из резервного SQL файла (резервной копии базы данных) одним из следующих способов:

- используя интерфейс импорта программы phpMyadmin — открыть в браузере страницу

[http://Адрес\\_сервера/phpmyadmin/](http://Адрес_сервера/phpmyadmin/) и авторизоваться как пользователь **root** базы MySQL, выбрать необходимую базу, импорт, выбор файла резервной копии данной базы, применить

- используя SSH, административную консоль — выполнить команду **mysql -D DBASE -u root -p <BACKUP.SQL** где DBASE имя базы данных, а BACKUP.SQL файл резервной копии. Команда запросит пароль пользователя **root** базы данных MySQL.
- Либо воспользоваться имеющимся MySQL клиентом для импорта резервной копии

Резервирование данных настраивается системным администратором по необходимости

## Пароли и их замена

В системе используется парольный доступ на следующие компоненты:

- административная консоль (как локальная так и по SSH). Пользователь **root** пароль по умолчанию **asdzxc**. Для смены обратитесь к системному администратору или документации к команде `passwd`.
- MySQL сервер. Пользователь для администрирования базы — **root** пароль по умолчанию **asdzxc** (совпадение с консольным совершенно случайно! Могут различаться). Программа обращается к базе ограниченным в правах пользователем **user** с паролем по умолчанию **uplift**. Для смены обратитесь к системному администратору или документации к MySQL в целом. Либо воспользуйтесь возможностями phpMyadmin, открыв браузером страницу [http://Адрес\\_сервера/phpmyadmin/](http://Адрес_сервера/phpmyadmin/)