



**EVIKA**

АВТОМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ



KNXLM

LogicMachine. Контроллер для шины KNX, ModBus, интерфейс Ethernet, SCADA

---

## **Руководство по эксплуатации и применению**

Программируемый контроллер с устройством  
отображения данных Logic Machine

---

## Авторские права

Авторские права принадлежат компании Embedded Systems SIA© 2011. Все права защищены.

## Уведомление

Embedded Systems SIA сохраняет за собой право при необходимости вносить изменения в приведенную ниже информацию. Компания не несет ответственности за ошибки и неточности, допущенные в настоящем руководстве. Руководство предназначено исключительно для содействия операторам в использовании аппаратного и программного обеспечения программируемого контроллера Logic Machine.

## Товарные знаки

Товарный знак genOHM принадлежит компании Embedded Systems SIA. Настоящим подтверждается, что все прочие наименования и товарные знаки являются собственностью их владельцев.

## Вводная часть

Контроллер Logic Machine предназначен для программирования сложной логики работы устройств в сети KNX/EIB. Программируемый контроллер позволяет эффективно выстраивать процесс автоматизации зданий в соответствии с предъявляемыми требованиями, обеспечивая конечным пользователям неограниченные возможности с минимальными затратами.

Контроллер Logic Machine представляет собой встраиваемую платформу с интерфейсом FT1.2 или Ethernet, что позволяет расширять возможности KNX/EIB, управляя логикой работы устройств сети посредством клиентских скриптов, и определять отношения между объектами. Встроенный веб-сервер, а также репозиторий скриптов и шаблонов скриптов, которые предоставляются с исходным кодом, обеспечивают удобный, пользовательско-ориентированный интерфейс управления. Используя клиентские скрипты, Logic Machine может имитировать действия термостата, панели управления системой безопасности, управления освещением и т. п.

Для создания скриптов используется язык программирования Lua, который был расширен модулем KNX и работает на базе промышленной платформы. Следует отметить и ряд других отличительных характеристик Logic Machine: запуск скриптов по расписанию и на основе событий, а также как резидентных программ; автоматическое обнаружение объектов в сети KNX; встроенные автономные часы реального времени. Logic Machine обеспечивает бесперебойную работу и может использоваться для решения различных задач, в том числе при неблагоприятных условиях окружающей среды.

## Техническая поддержка

Неисправные устройства подлежат возврату EVIKA.

В случае возникновения каких-либо вопросов в части технического обслуживания устройства необходимо обратиться в службу технической поддержки. Часы работы: 10:00–18:00 (понедельник – пятница)

Контактная информация:

Адрес электронной почты: [support@evika.ru](mailto:support@evika.ru)

Тел.: +7 (495) 988-09-91

Обновления для встроенного программного обеспечения: [www.evika.ru](http://www.evika.ru)

---

## Технические характеристики

### Аппаратное обеспечение

#### ► **На основе надежной аппаратной платформы PowerPC**

Поддержка промышленных температурных регуляторов; устройство контроля показателей аппаратного обеспечения; устройство защиты от перенапряжения разъема для подключения к источнику питания и LAN гарантирует бесперебойную и надежную работу. Допускается установка в агрессивных промышленных средах. Не требует технического обслуживания.

#### ► **Встроенный интерфейс FT1.2**

Logic Machine подсоединяется напрямую к шине KNX/EIB. Отсутствует необходимость в использовании внешнего интерфейса TPUART или FT1.2.

#### ► **Встроенные часы реального времени**

Часы реального времени обеспечивают надежный контроль времени для бесперебойной работы устройства и своевременное выполнение всех запланированных скриптов. В случае перебоев в питании может использоваться резервный аккумулятор.

### Программное обеспечение и скрипты

#### ► **Интерфейс пользователя AJAX**

Подключается к любому персональному компьютеру на базе любой платформы, не требует установки дополнительного программного обеспечения. В случае присвоения реального IP-адреса обеспечивается возможность удаленного доступа.

#### ► **Автоматическое обнаружение всех объектов**

Logic Machine добавляет новые объекты в базу данных в автоматическом режиме по мере выполнения групповых событий; необходимость в ручном вводе данных отсутствует. Для обеспечения поддержки новым объектом тех или иных скриптов следует задать соответствующий тип данных.

#### ► **Язык скриптов Lua**

Для создания скриптов Logic Machine применяется высокоэффективный, облегченный и высокопроизводительный язык программирования Lua. Расширяемые функции позволяют не только обеспечивать взаимосвязь объектов в рамках сети KNX, но и выходить за границы стандартной шины и соединять их с другими сетями и службами.

#### ► **Репозиторий скриптов. Справка**

Программа для создания скриптов использует заранее заданные фрагменты кодов, что существенно облегчает процесс программирования. Кроме того, все пользователи получают доступ к репозиторию скриптов, где приведены примеры и инструкции по применению наиболее используемых скриптов.

#### ► **Присвоение имен и маркеров**

Для маркировки объектов могут использоваться уникальные имена или набор тегов. Таким образом, обеспечена возможность изменения групповых адресов без риска нарушения целостности скриптов. Система тегов позволяет одновременно работать с несколькими похожими объектами (например установленными в одном месте).

#### ► **Запланированные скрипты. Скрипты на основе событий. Резидентные скрипты**

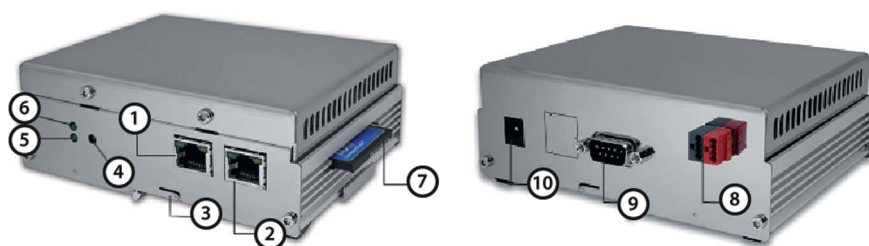
Logic Machine поддерживает скрипты трех видов.

*Основанные на событиях* — скрипты, выполняемые в случае обнаружения группового события на шине. Как правило, применяются при необходимости получения ответа максимально приближенно к реальному режиму времени.

*Запланированные* — скрипты, выполняемые в определенное время. Используются в сочетании с различными системами безопасности и для имитации присутствия.

*Резидентные* — скрипты, использующие упорядоченный опрос для проверки статуса объектов. Как правило, применяются в сочетании с системами вентиляции и отопления, когда данные принимаются от нескольких групповых адресов.

## СПЕЦИФИКАЦИЯ УСТРОЙСТВА



- 1 - Ethernet 1
- 2 - Ethernet 2
- 3 - DIN rail
- 4 - Programmable RESET button
- 5 - Power/Status LED

- 6 - Programmable LED
- 7 - 128MB Compact Flash
- 8 - FT1.2 connectors
- 9 - DB9 serial interface
- 10 - Power port

DIN rail	DIN-рейка
Programmable RESET button	Программируемая кнопка RESET
Power/Status LED	Индикатор питания (состояния)
Programmable LED	Программируемый индикатор
128MB Compact Flash	Карта памяти 128 Мб
FT1.2 connector	Разъем FT1.2
D89 serial interface	Последовательный интерфейс D89
Power port	Разъем для подсоединения к источнику питания

### В комплект поставки Logic Machine входят:

- ▶ контроллер с установленным программным обеспечением
- ▶ фиксаторы DIN-рейки
- ▶ блок питания 24 В
- ▶ интерфейс FT1.2
- ▶ карта памяти Compact Flash 128 Мб

<b>Контроллер</b>	ЦП	PPC405 266 МГц
	ЗУ с произвольной выборкой	32–64 Мб, 32-битная шина данных
	Флэш-память NOR	8 Мб, 32-битная шина данных
	Разъемы Ethernet	2 шт. 10 Base T / 100 Base TX (RJ-45), с защитой от статического электричества
	Операционная система	OpenWRT
	Часы реального времени	Да
	Разъемы MiniPCI	2 шт.
	Последовательные порты	2 x RS232 (1 x DB9, 1 x FT1.2)
	Карта памяти	128 Мб
	Интерфейс FT1.2	Да
<b>Источник питания</b>	Номинальное напряжение	7–36 В постоянного тока; один источник питания (не более 6 А) с защитой от статического электричества. Поддерживается функция питания через Ethernet
	Потребляемая мощность	1,5 Вт
<b>Соединение</b>	KNX/EIB	Разъем KNX
	RS232	DB9
<b>Устройства индикации и отображения данных</b>	Индикатор	1 индикатор состояния (питания); 1 программируемый индикатор (с использованием универсального устройства ввода-вывода)
<b>Тип защиты</b>	IP20	В соответствии с требованиями DIN EN 60529
	Защита и устойчивость	Защита питания и LAN-портов от перенапряжения, встроенный watchdog для контроля состояния устройства
<b>Температура</b>	Рабочая	–30...+85 °C
	Хранение	–30...+85 °C
	Транспортировка	–30...+85 °C
<b>Физические свойства</b>	Конструкция	DIN-рейка для установки на монтажной шине 35 мм
	Материал корпуса (цвет)	Алюминий (серый)
	Габаритные размеры (Ш x В x Д)	135 x 105 x 50 мм
	Вес	0,3 кг
<b>Аттестация</b>	Совет Европы (СЕ), «Правила ограничения содержания вредных веществ» (RoHS)	В соответствии с требованиями к электромагнитной совместимости и низкому напряжению
	РОСТ	Сертификат
<b>Гарантийный срок</b>	2 года	

## БЫСТРАЯ УСТАНОВКА И ЗАПУСК

Устройство должно использоваться в сухих помещениях.

Перед установкой Logic Machine посредством ETS задать настройки для всех объектов KNX.

- 1) Монтаж (демонтаж) устройства на DIN-рейке (допускается использование в качестве настольного устройства) (3)
- 2) Подсоединение (отсоединение) к шине KNX (8)
- 3) Подсоединение блока питания 24 В к устройству (10)
- 4) Подсоединение персонального компьютера через кабель Ethernet (1)

## Автоматическая настройка

Протокол DHCP Logic Machine по умолчанию подключен к интерфейсу Ethernet 1.

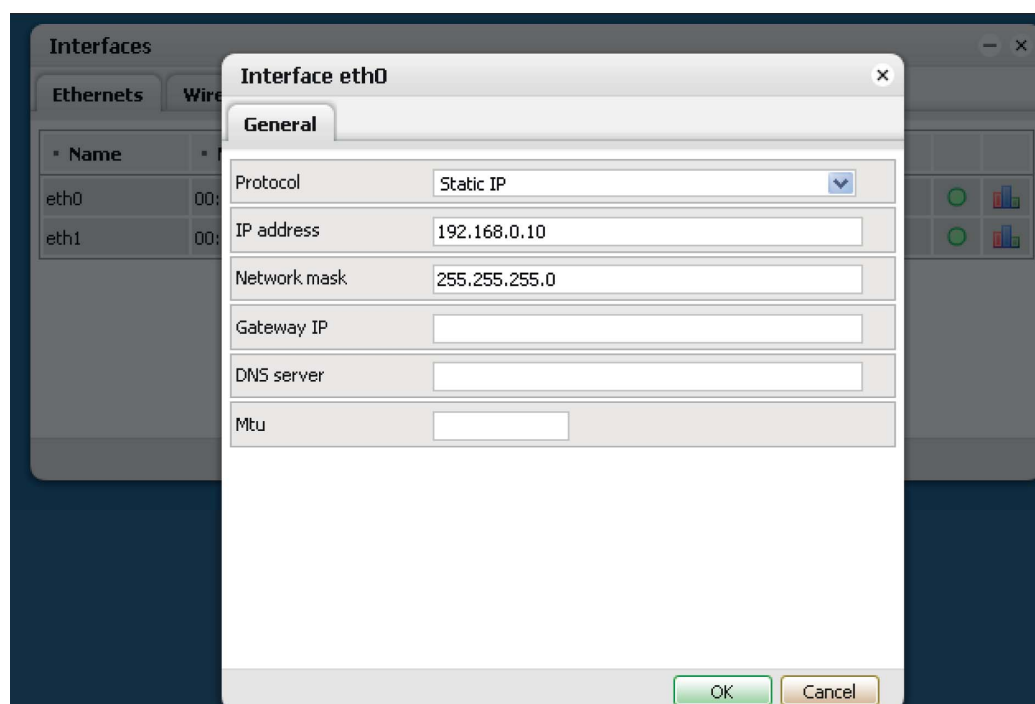
IP-адрес, присваиваемый интерфейсу Ethernet сервером DHCP, может быть считан при помощи программного обеспечения ETS.

## IP-адреса (по умолчанию)

Logic Machine / Network Configuration Имя пользователя	admin
Logic Machine / Network Configuration Пароль	admin
User mode visualization / Touch visualization Имя пользователя	Только для чтения: visview Редактирование: viscontrol Редактирование + администрирование: visadmin
User mode visualization / Touch visualization Пароль	Только для чтения: visview Редактирование: viscontrol Редактирование + администрирование: visadmin
IP-адрес для LAN1 (1)	192.168.0.10
Сетевая маска для LAN1 (1)	255.255.255.0

## Смена IP-адреса

В окне Network Interfaces выбрать интерфейс для изменения IP-адреса.



---

*Protocol* – протокол, используемый для присвоения адресов.

*None* – протокол не используется.

*Static IP* – статический IP-адрес. По умолчанию: 192.168.0.10.

*DHCP* – протокол DHCP используется для задания IP-адреса.

*Current IP* – IP-адрес, присвоенный сервером DHCP. Поле активно только в том случае, если присвоен IP-адрес. В противном случае данное поле недоступно.

*PPPoE* – использовать протокол PPP.

*Username* – имя пользователя для соединения с сервером PPPoE.

*Password* – пароль.

*Keepalive* – время активного подключения.

*Dial on Demand* – подключение по требованию.

*PPTP server IP* – IP-адрес сервера PPPoE для соединения.

*Network mask* – сетевая маска. По умолчанию: 255.255.255.0 (/24).

*Gateway IP* – IP-адрес шлюза.

*DNS server* – IP-адрес сервера DNS.

*MTU* – максимальный размер пакета, который может быть передан по протоколу обмена данными. По умолчанию: 1500.

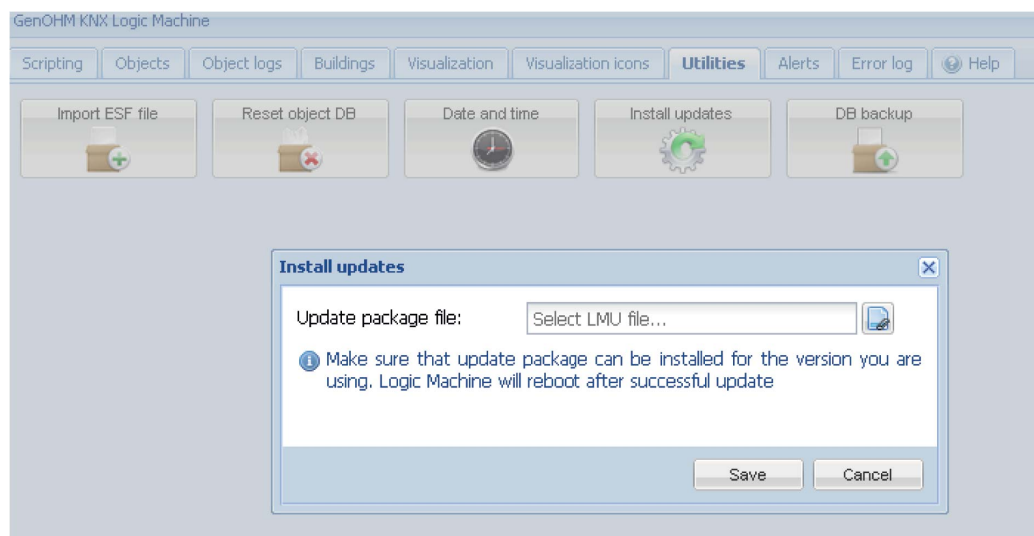
После внесения изменений в верхнем правом углу дисплея появляется приведенная ниже пиктограмма.

Нажать для подтверждения внесенных изменений.

 Apply changes

## ОБНОВЛЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для обновления программного обеспечения Logic Machine используется графический интерфейс пользователя. Обновления отображаются в виде графических объектов на странице службы технической поддержки [evika.ru](http://evika.ru).



### 1. Обновление программного обеспечения в автоматическом режиме

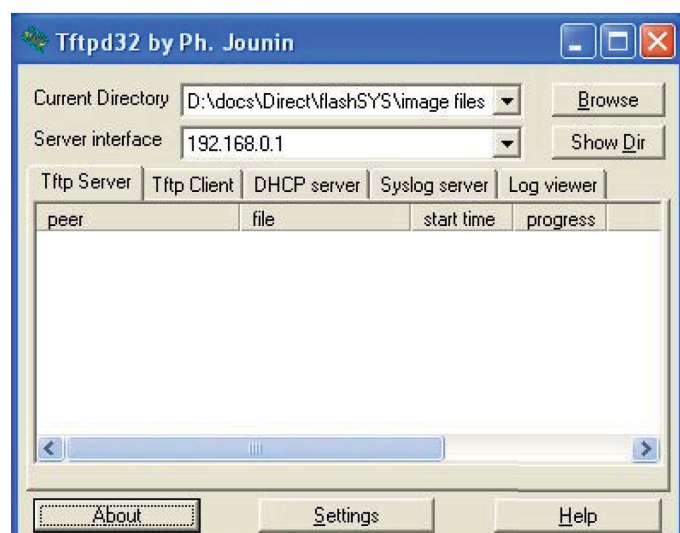
После входа в меню Logic Machine выбрать вкладку *Utilities* и нажать *Install updates*. После выбора \*.LMU file в соответствующем каталоге нажать *Save*. Перезагрузка устройства производится по истечении 5 секунд, после чего выполняется установка обновлений для программного обеспечения.

### 2. Обновление программного обеспечения в ручном режиме

Обновление в ручном режиме выполняется путем очистки флэш-памяти NAND и карты памяти. Очистка карты памяти производится путем ее подключения к персональному компьютеру и удаления всех файлов в ручном режиме.

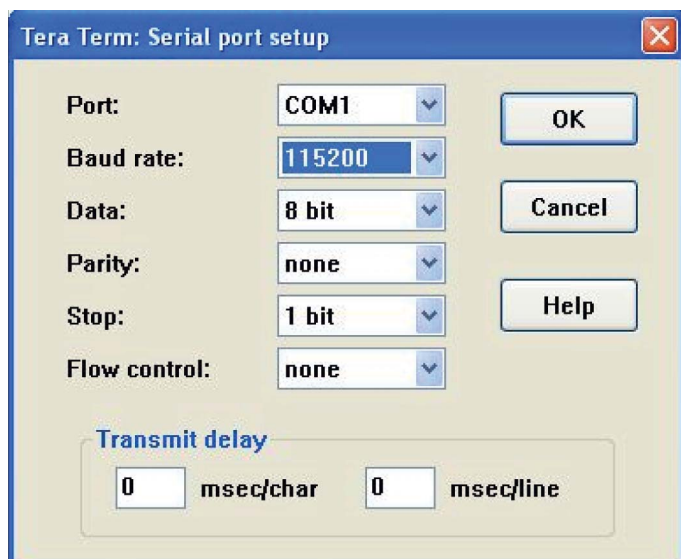
Для очистки флэш-памяти NAND в ручном режиме и записи новых файлов необходимо:

- ▶ Загрузить сервер TFTP (например, TFTP32 по адресу: [http://tftpd32.jounin.net/tftpd32\\_download.html](http://tftpd32.jounin.net/tftpd32_download.html)).
- ▶ Обеспечить наличие терминальной программы, например, TeraTerm PRO (<http://sourceforge.jp/projects/ttssh2/files/>).
- ▶ Загрузить файл изображений (\*.img).
- ▶ Обеспечить соединение карты Ethernet компьютера с разъемом Ethernet Logic Machine с использованием любого типа кабеля (А и В).
- ▶ Для LAN компьютера задать IP 192.168.0.1.
- ▶ Запустить программу сервера TFTP и указать место расположения файла.

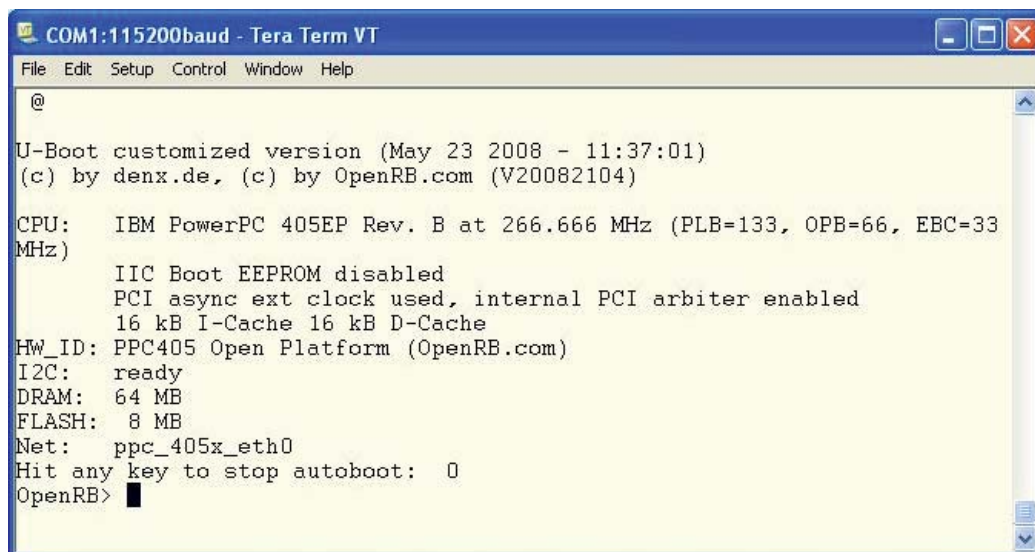




- ▶ Запустить терминальную программу TeraTerm PRO; задать *Baud Rate* и *Port* в меню *Setup > Serial port*. Параметр порта представляет собой имя последовательного порта персонального компьютера (COM1, COM2, ..., COMX). Для *Baud Rate* задать 115200.



- ▶ Подсоединить источник питания (5–40 В) к соответствующему разъему Logic Machine. Нажать любую кнопку и удерживать в течение 3 секунд.



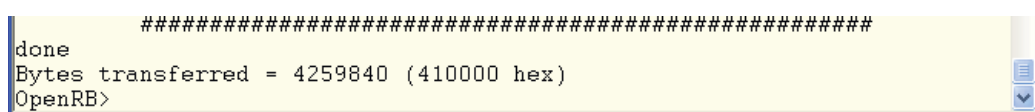
- ▶ Ввести следующие команды (синим цветом):

`erase 0xff800000 fffbffff` – форматирование флэш-памяти.

`tftp 100000 lm-19.03.2010.img` – копирование загруженных графических данных через TFTP на 100000-й адрес памяти.

`cp.b 0x100000 0xff800000 0x410000` – копирование загруженного файла изображений из памяти во флэш-память.

*Внимание! Адрес памяти, выделенный жирным шрифтом, в обязательном порядке меняется на адрес, полученный после выполнения команды №2. На приведенном ниже примере – 0x410000.*



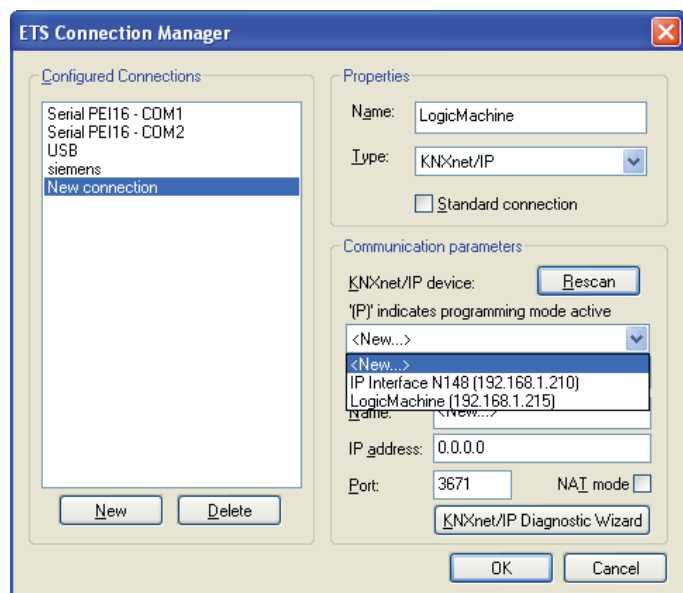
`reset` – перезагрузка.

- ▶ Дождаться перезагрузки Logic Machine. Это займет не более 60 секунд.

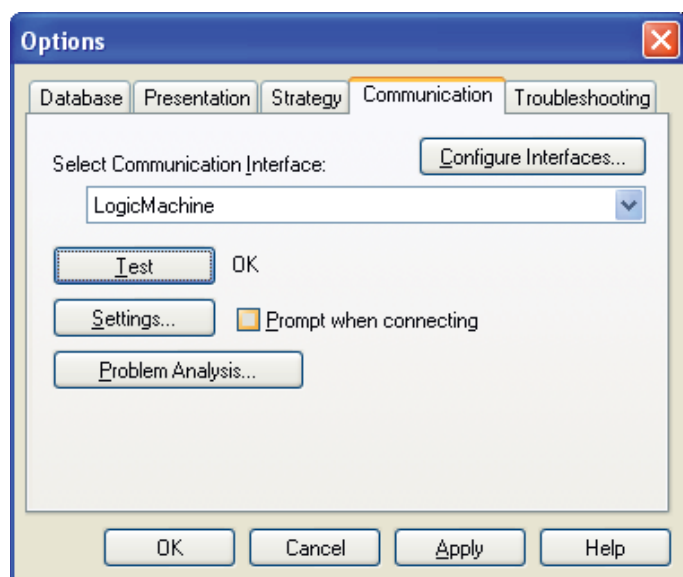
## LOGIC MACHINE ДЛЯ НАСТРОЙКИ КОНФИГУРАЦИИ СЕТИ KNX/EIB ЧЕРЕЗ ПРОТОКОЛ TCP/IP

Для использования Logic Machine в рамках KNXnet/IP следует добавить устройство в ETS Connection Manager.

- Выбрать *Extras > Options > Communication > Configure interfaces*.



- Задать *Name* для соединения (по выбору).
- Выбрать *Type* = KNXnet/IP.
- Нажать кнопку *Rescan*; из выпадающего меню выбрать Logic Machine
- Нажать OK.
- В окне *Options > Communication* выбрать вновь созданное соединение; в выпадающем меню установить его в качестве *Communication Interface*.
- Для проверки соединения при помощи ETS нажать кнопку *Test*.



- Убедиться, что шине присвоен статус *Online* – в ETS нажать кнопку.

## НАСТРОЙКА КОНФИГУРАЦИИ KNX В NETWORK CONFIGURATION

Настройка KNX производится в окне *System > KNX connection*.

### Вкладка General

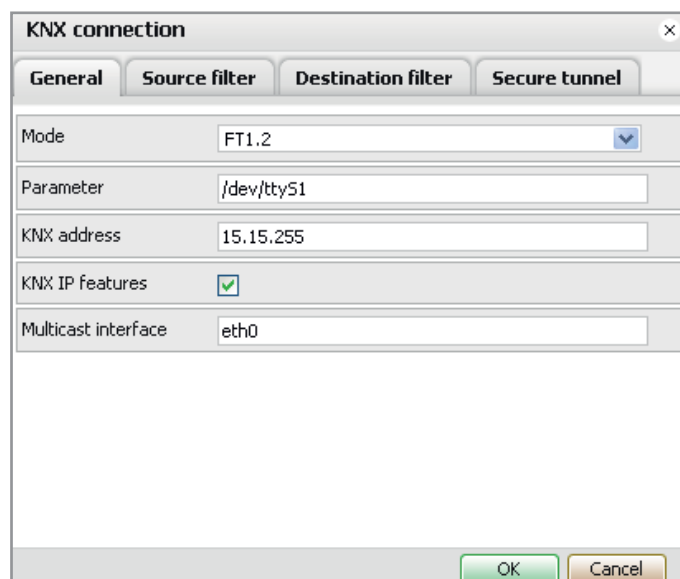
*Mode [FT1.2 / TP-UART / EIBnet IP Tunneling]* — режим соединения с KNX. По умолчанию используется встроенный интерфейс FT1.2.

*Parameter* — последовательный объект в системе OpenWRT, отвечающий за соединение с KNX.

*KNX address* — физический адрес устройства в KNX.

*KNX IP features* — устройство с KNX IP; используется, в частности, для настройки конфигурации сети KNXnet/IP.

*Multicast interface* — групповой интерфейс; используется для отправки телеграмм KNX в другие сети KNX через протокол TCP/IP.

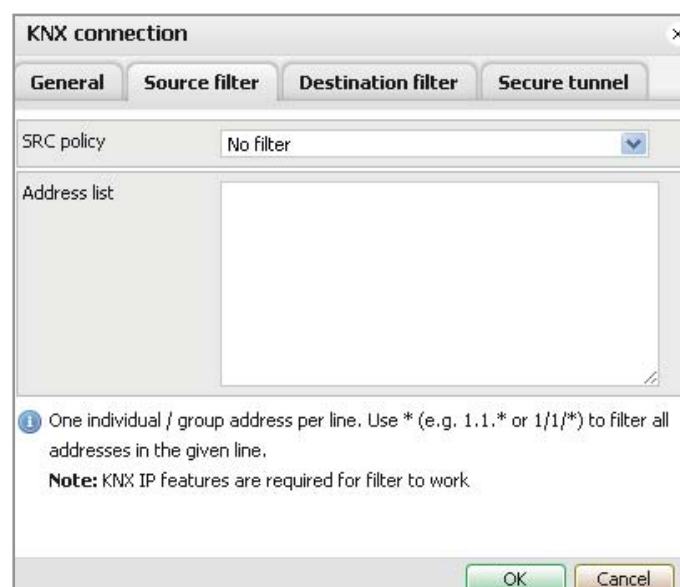


### Вкладка Source filter

*SRC policy [No filter / Accept selected individual addresses / Drop selected individual addresses]* (Без фильтра / Разрешенные адреса / Запрещенные адреса) — правило, применяемое к списку адресов, посылающих пакеты.

*Address list* — перечень индивидуальных и групповых адресов. В каждой строке указывается один адрес. Использовать \* (пример: 1.1.\* or 1/1/\*) для фильтрации адресов в соответствующей строке.

**Внимание!** Для использования режима фильтрации включить *KNX IP features*.

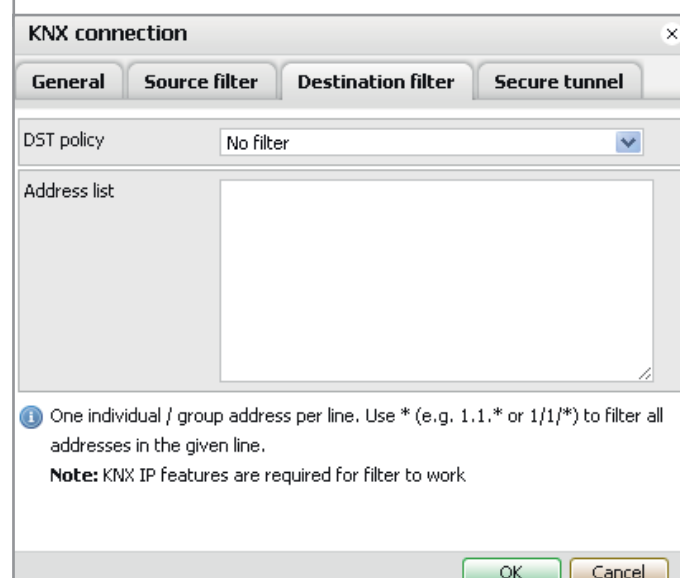


### Вкладка Destination filter

*DST policy [No filter / Accept selected individual addresses / Drop selected individual addresses]* (Без фильтра / Разрешенные адреса / Запрещенные адреса) — правило, применяемое к списку адресов, принимающих пакеты.

*Address list* — перечень индивидуальных и групповых адресов. В каждой строке указывается один адрес. Использовать \* (пример: 1.1.\* or 1/1/\*) для фильтрации адресов в соответствующей строке.

**Внимание!** Для использования режима фильтрации включить *KNX IP features*.



## Вкладка Secure tunnel

Существует возможность создания безопасного туннеля между двумя сетями KNX. В сравнении со стандартными туннелями, использующими протокол UDP, в данном случае применяется протокол TCP. Благодаря функции подтверждения доставки пакетов данных обеспечивается высокая надежность соединения. Таким образом, отправитель получает уведомление о доставке (неудачной доставке) пакета получателю.

*Secure tunnel [Disabled / Client / Server] (Отключен/ Клиент / Сервер)* – режим безопасного туннеля.

*Server IP* – указать IP-сервер в случае организации связи по защищенному каналу.

*Local IP* – локальный IP-адрес.

*Network mask* – сетевая маска.

*Password* – пароль.

The screenshot shows a software window titled "KNX connection" with a close button (X) in the top right corner. It contains four tabs: "General", "Source filter", "Destination filter", and "Secure tunnel", with the last one being active. The "Secure tunnel" tab has a dropdown menu labeled "Secure tunnel" with "Client" selected. Below this are five input fields: "Server IP", "Local IP", "Network mask", and "Password". An information icon (i) is located to the left of a text block that reads: "Secure tunnel creates an encrypted network between several KNX nodes. **Password** must match for every node in a single network. **Local IP** - IP address of the node on the secure tunnel network. **Server IP** - real IP address of the server node. **Local IP** and **Server IP** must be on different subnetworks." At the bottom right are "OK" and "Cancel" buttons.

## РУКОВОДСТВО ПО ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

### 1. Вход в графический интерфейс пользователя

По умолчанию для Logic Machine в интерфейсе LAN1 задан следующий IP-адрес: 192.168.0.10. Ввести вышеупомянутый IP-адрес в адресной строке браузера.

*Внимание! Убедиться, что используются IP-настройки соединения компьютера и Logic Machine одной подсети.*

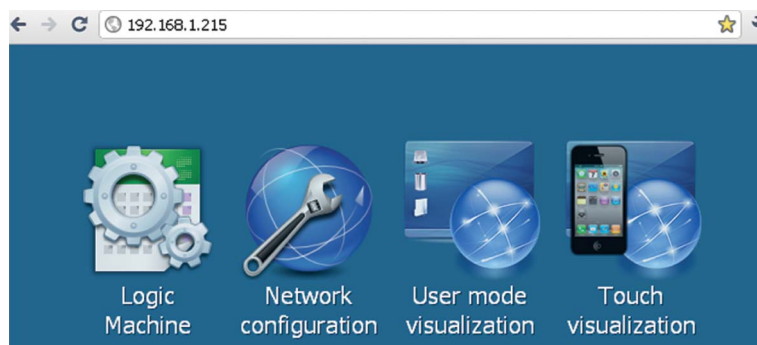
После входа выводится страница по умолчанию.

**Logic Machine** – настройка параметров KNX: скрипты, связи между объектами, предупреждения и объекты KNX, визуализация управления зданием.

**Network configuration** – настройка IP и KNXnet/IP.

**User mode visualization** – определение отображения объектов управления.

**Touch visualization** – система отображения данных для сенсорных устройств iPhone/iPod/iPad/Android.



### 2. Окно Logic Machine

Исходный каталог для настройки конфигурации Logic Machine. Главное меню включает следующие подменю:

*Scripting* – управление архивом скриптов.

*Objects* – перечень сетевых объектов KNX.

*Object logs* – журналов событий объектов шины KNX.

*Building* – загрузка сведений о здании и файла изображений.

*Visualization* – управление и контроль графического отображения данных.

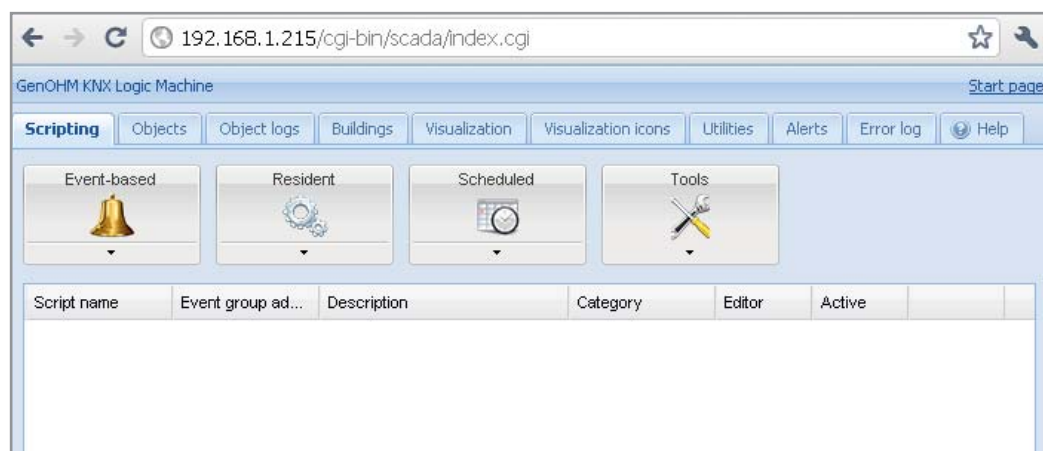
*Visualization icons* – управление пиктограммами.

*Utilities* – сервисные программы (в т. ч. импорт из ETS, сброс базы данных объектов, резервное копирование, обновление системы).

*Alerts* – предупредительные сообщения, заданные с использованием функции *alert*.

*Error log* – сообщения об ошибках на шине KNX.

*Help* – справка по синтаксису скриптов.



Имя пользователя	Пароль
admin	admin

## 2.1. Меню Scripting

Настоящее меню позволяет добавлять и управлять скриптами различных типов. Для создания пользовательских скриптов используется язык программирования Lua. Основные моменты, связанные с применением данного языка программирования, описаны в первом издании руководства «Программирование на языке Lua», которое находится в свободном доступе по адресу: <http://lua.org/pil/>.

*Внимание! Формат данных: в большинстве случаев данные хранятся и передаются друг другу компонентами Logic Machine с использованием закодированных в шестнадцатеричном формате строк (2 байта /1 байт данных).*


Сценарии бывают трех основных видов:

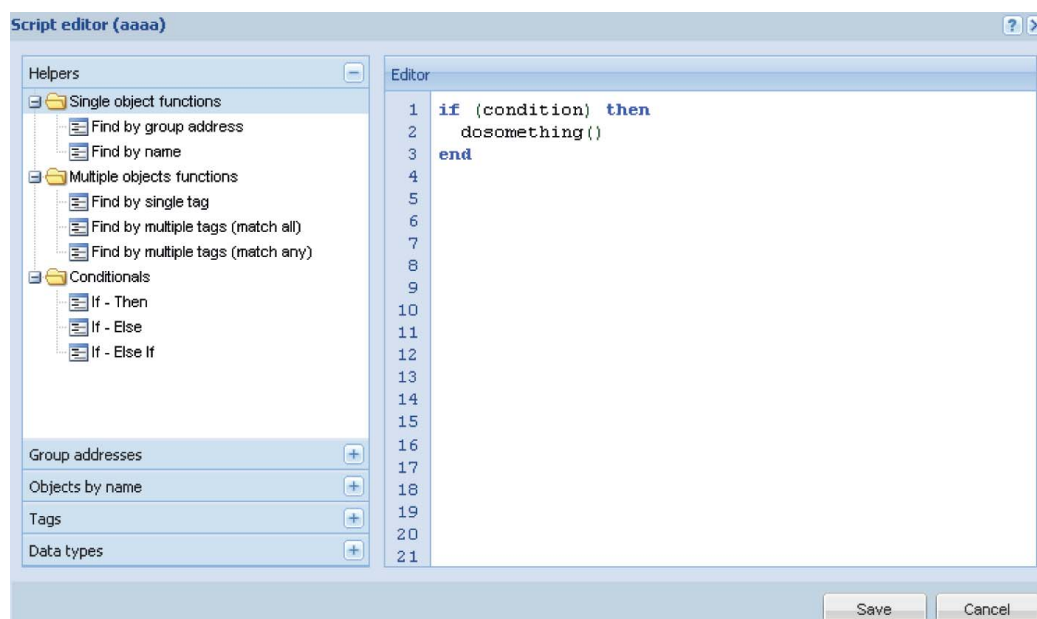
**Основанные на событиях** – скрипты, выполняемые при наступлении определенного группового события на шине данных. Как правило, применяются при необходимости получения ответа максимально приближенно к реальному режиму времени.



**Резидентные** – скрипты, использующие упорядоченный опрос для проверки статуса объектов. Как правило, применяются в сочетании с системами вентиляции и отопления, когда данные принимаются от нескольких групповых адресов.

**Запланированные** – скрипты, выполняемые в определенное время. Используются в сочетании с различными системами безопасности и для имитации присутствия.

### 2.1.1. Редактор скриптов. Синтаксис

После добавления скрипта в графе *Editor* появляется пиктограмма ; это означает, что пользователь может открыть скрипт в режиме редактирования и внести необходимые изменения с использованием заранее заданных фрагментов кода.



В случае если пользователь не знаком с синтаксисом, выводится подсказка, где описан порядок создания скриптов. Фрагменты кода экономят время и облегчают процесс программирования (по сравнению с окном *Add script*). После выбора соответствующего фрагмента кода программа автоматически добавляет его в поле редактирования. В правом верхнем углу отображается ярлык справки и закрытия окна ( ).

Пять основных разделов редактора скриптов:

- ▶ *Helpers* – заранее заданные фрагменты кода (в частности, функции «если, то»). Данная группа включает три подгруппы:
  - ▶ *Single object functions* – функции одного объекта.
  - ▶ *Multiple object functions* – функции нескольких объектов.
  - ▶ *Conditional* – условные функции объектов.
- ▶ *Group addresses* – существующие групповые адреса шины KNX.
- ▶ *Objects by name* – выбор объекта по имени.
- ▶ *Tags* – выбор объекта по тегу.
- ▶ *Data types* – выбор объекта по типу данных.

---

## ФУНКЦИИ ОБЪЕКТОВ

*Knxobject* обеспечивает простой доступ к базе данных объектов KNX.

Функция *knxobject.get* предназначена для направления запросов в базу данных объектов KNX с указанием соответствующих параметров. Значение объекта декодируется автоматически только в том случае, если в меню *Objects* для этого объекта указан тип данных.

### **knxobject.get ('address', address)**

Получение одиночного объекта с указанным групповым адресом (второй параметр функции).

### **knxobject.get ('name', name)**

Получение одиночного объекта с указанным именем (меню *Objects*).

Функция возвращает *Lua nil* в случае невозможности обнаружить объект; в противном случае возвращается *Lua table* и следующие параметры:

- ▶ *address* – групповой адрес объекта;
- ▶ *updatetime* – дата последнего обновления в формате времени UNIX. Для преобразования в доступный для чтения формат использовать *Lua os.date()*;
- ▶ *datahex* – шестнадцатеричный код объекта;
- ▶ *dataraw* – не декодированное двоичное значение объекта.

Если в поле *Objects* указан тип данных объекта, доступны следующие поля:

- ▶ *name* – уникальное имя объекта;
- ▶ *datatype* – тип данных объекта, указанный пользователем;
- ▶ *decoded* – установлено true при наличии возможности декодирования значения;
- ▶ *data* – декодированное значение объекта.

### **knxobject.get ('tag', tags, mode)**

Возвращает массив объектов с указанным тегом (определяется в меню *Objects*). Второй параметр (тег) – может быть массивом или строкой. Третий параметр (режим) – *all* (возвращает объекты, имеющие все указанные теги) и *any* (значение по умолчанию, возвращаются объекты, имеющие хотя бы один из указанных тегов).

*Внимание!* Приведенные ниже функции используются исключительно для доступа к объектам напрямую по групповому адресу; в противном случае рекомендуется использовать функции одного или нескольких объектов.

### **knxobject.write (groupaddress, data, datatype)**

Запись данных по указанному групповому адресу. Если тип данных не указан, данные должны быть в шестнадцатеричном коде. В противном случае данные кодируются исходя из указанного типа данных. В случае невозможности закодировать данные или отсутствия соединения с сетью KNX функция возвращает FALSE. В любом другом случае возвращается TRUE.

### **knxobject.response (groupaddress, data, datatype)**

См. *knxobject.write*. Запрос группе по указанному групповому адресу.

### **knxobject.read (groupaddress)**

Запись данных по указанному групповому адресу.

*Внимание!* Настоящая функция отображает данные непосредственно после запроса и не может быть использована для вывода результатов запроса на считывание.



---

## ФУНКЦИИ ОДНОГО ОБЪЕКТА

Объекты, полученные с использованием *knxobject.get('address', ...)* или *knxobject.get('name', ...)*, имеют приведенные ниже функции.

*Внимание! В обязательном порядке убедиться в том, что искомый объект был найден. В противном случае обращение к данным функциям приведет к ошибке. Пример приведен ниже.*

### **object:write (data, datatype)**

Запись данных в групповой адрес, ассоциированный с объектом. Параметр *datatype* опционален и требуется только в том случае, если данные вводятся в шестнадцатеричном формате. Как правило, данные задаются как есть и затем преобразуются в нужный формат в автоматическом режиме. Необходимо убедиться, что в меню *Objects* веб-интерфейса Logic Machine указан тип данных.

### **object:response (data, datatype)**

См. *object:write*. Запрос на ответ по ассоциированному с объектом групповому адресу.

### **object:read ()**

Запрос на считывание данных по ассоциированному с объектом групповому адресу.

### **Примеры**

```
myobject = knxobject.get('name', 'My object name')
```

```
-- значение knxobject.get будет false, если объект не обнаружен
```

```
if myobject then
```

```
    myobject:write(1) -- указать 1 для параметра объекта
```

```
end
```

```
myobject = knxobject.get('address', '1/1/15')
```

```
-- убедиться, что запрашиваемый объект был найден
```

```
If myobject then
```

```
    myobject:write(52.12, dt.float16) -- прямое присвоение типа данных
```

```
    dt.float16 (2-байтовая плавающая точка)
```

```
end
```

## ФУНКЦИИ НЕСКОЛЬКИХ ОБЪЕКТОВ

Функции одного объекта могут использоваться для массивов данных, полученных при помощи *knxobject.get('tag', ...)*.

*Внимание! Убедиться, что маркированные объекты имеют один тип данных; в противном случае групповые запросы на запись данных могут привести к возникновению ошибок.*

## ФУНКЦИИ ТИПА ДАННЫХ

Объект *knxdatatype* обеспечивает кодирование и расшифровку данных в формате Lua и KNX.

### **knxdatatype.decode (value, datatype)**

Преобразование шестнадцатеричных кодированных данных в переменные Lua исходя из указанного типа данных. Тип данных указывается как основной (целое число от 1 до 16) или вспомогательный тип данных KNX (целое число от 1000 до 16000).

Возвращаемые значения:



- 
- ▶ успешное выполнение — расшифрованные данные в формате Lua (тип зависит от типа данных), длина значения в байтах;
  - ▶ ошибка — ноль, строка ошибки.

### **knxdatatype.encode (value, datatype)**

Преобразование значений Lua в шестнадцатеричные кодированные данные исходя из указанного типа данных.

Возвращаемые значения:

- ▶ успешное выполнение — таблица Lua (2 колонки):
  - ▶ `dataraw` (строка) — несформированные двоичные данные (не для всех типов данных);
  - ▶ `datahex` (строка) — шестнадцатеричные кодированные данные;
- ▶ ошибка — ноль, строка ошибки.

## **ТИПЫ ДАННЫХ**

Для кодирования и расшифровки данных KNX могут использоваться приведенные ниже типы данных. Ниже указаны данные на языке Lua и предварительно заданные константы (жирным шрифтом).

- ▶ 1. 1 bit (*boolean*) — **dt.bool** — логический.
- ▶ 2. 2 bit (*1 bit controlled*) — **dt.bit2** — числовой.
- ▶ 3. 4 bit (*3 bit controlled*) — **dt.bit4** — числовой.
- ▶ 4. 1 byte ASCII character — **dt.char** — строковый.
- ▶ 5. 1 byte unsigned integer — **dt.uint8** — числовой.
- ▶ 6. 1 byte signed integer — **dt.int8** — числовой.
- ▶ 7. 2 byte unsigned integer — **dt.uint16** — числовой.
- ▶ 8. 2 byte signed integer — **dt.int16** — числовой.
- ▶ 9. 2 byte floating point — **dt.float16** — числовой.
- ▶ 10. 3 byte time/day — **dt.time** — таблица со следующими данными:
  - ▶ *day* — числовое значение (0–7);
  - ▶ *hour* — числовое значение (0–23);
  - ▶ *minute* — числовое значение (0–59);
  - ▶ *second* — числовое значение (0–59).
- ▶ 11. 3 byte date — **dt.date** — таблица со следующими данными:
  - ▶ *day* — числовое значение (1–31);
  - ▶ *month* — числовое значение (1–12);
  - ▶ *year* — числовое значение (1990–2089).
- ▶ 12. 4 byte unsigned integer — **dt.uint32** — числовой.
- ▶ 13. 4 byte signed integer — **dt.int32** — числовой.
- ▶ 14. 4 byte floating point — **dt.float32** — числовой.
- ▶ 15. 4 byte access control — **dt.access** — числовой (в настоящее время поддерживается не в полном объеме).
- ▶ 16. 14 bit ASCII string — **dt.string** — строковые и нулевые символы ('\\0') при расшифровке не учитываются.

## **ОБЩИЕ ФУНКЦИИ. ХРАНЕНИЕ ДАННЫХ. ПРЕДУПРЕДИТЕЛЬНЫЕ СООБЩЕНИЯ**

### **Хранение данных**

Объект *storage* обеспечивает долгосрочное хранение основных данных для скриптов пользователя. Поддерживаются исключительно приведенные ниже типы данных Lua:

- ▶ логические
- ▶ числовые
- ▶ строковые
- ▶ табличные

---

### **storage.set (key, value)**

Задание нового значения для выбранного ключа. Предыдущее значение перезаписывается. Возвращается результат операции *boolean* и в случае *false* строка с описанием ошибки.

### **storage.get (key, default)**

Получение значения ключу; если не найдено значение, то возвращается значение по умолчанию — *default (nil)*, если значение не указано).

Примечание: все пользовательские скрипты хранят данные в одном хранилище. Убедиться, что один и тот же ключ не используется для сохранения различных типов данных.

### **Примеры**

На приведенных ниже примерах показан базовый синтаксис для *storage.set*. Логическая константа *true* возвращается в том случае, если все параметры заданы верно:

```
result = storage.set('my_stored_value_1', 12.21)
```

В этом примере будет возвращен *false*, потому что мы пытаемся сохранить функцию.

```
testfn = function(t)
    return t * t
end
```

```
result = storage.set('my_stored_value_2', testfn) — ошибка
```

На приведенных ниже примерах показан базовый синтаксис для *storage.get*. Если значение ключа не обнаружено, после первого запроса возвращается *nil*, после второго — 0 (задано по умолчанию).

```
result = storage.get('my_stored_value_3') -- nil если значение не обнаружено
```

```
result = storage.get('my_stored_value_3', 0) — 0, если значение не обнаружено
```

При сохранении таблиц проверить тип возвращаемых данных. Пример: мы создаем сохраняемый элемент с ключом *test\_object\_data*:

```
objectdata = {}
objectdata.temperature = 23.1
objectdata.scene = 'default'
```

```
result = storage.set('test_object_data', objectdata) — сохранить переменную objectdata
```

как  
'test\_object\_data'

Теперь мы выбираем данные из хранилища. Проверяем тип данных на корректность.

```
objectdata = storage.get('test_object_data')
```

```
if type(objectdata) == 'table' then
    if objectdata.temperature > 24 then
        -- принять меры при повышенной температуре
    end
end
end
```

---

## Функция формирования предупредительных сообщений

### **Alert (message, [var1, [var2, [var3]]])**

Сохранение предупредительного сообщения и текущего системного времени в основной базе данных. Все предупредительные сообщения хранятся в меню *Alerts*. Параметры настоящей функции идентичны *Lua string.format*.

Пример:

температура = 25.3

```
if температура > 24 then
    -- отображаемое сообщение: 'Temperature levels are too high: 25.3'
    alert ('Temperature level is too high: %.1f', temperature)
end
```

## Библиотека функций

Большая часть функций используется системой, однако они также могут применяться при создании пользовательских скриптов для кодирования и расшифровки данных.

### **common.decode\_ia (indaddressa, indaddressb)**

Преобразование индивидуального адреса в двоичном формате в строку Lua. Данная функция принимает один или два параметра (интерпретируются как два отдельных байта).

### **common.decode\_ga(groupaddressa, groupaddressb)**

Преобразование группового адреса в строку Lua. Данная функция принимает один или два параметра (интерпретируются как два отдельных байта).

### **common.encode\_ga(groupaddress, separate)**

Преобразование строки Lua в двоичный групповой адрес. Групповой адрес возвращается как одно значение Lua, если второй параметр — *false*; в противном случае выводятся два отдельных байта.

### **common.hextoint(hexvalue, bytes)**

Преобразование шестнадцатеричной строки в целое число установленной длины в байтах.

### **common.inttohex(intvalue, bytes)**

Преобразование целого числа в шестнадцатеричную строку установленной длины в байтах.

### **common.strtohex(str)**

Преобразование двоичной строки в шестнадцатеричную кодированную строку.

### **common.hextostr(hexstr)**

Преобразование шестнадцатеричной кодированной строки в двоичную строку.

## Формат запланированного времени выполнения скрипта

Для планирования выполнения скриптов используется стандартный (*cron*) формат ввода даты и времени. Допустимые значения:

- ▶ \* — выполнять скрипт каждую минуту, час или день;
- ▶ \*/N — выполнять скрипт через каждые N минут, часов и дней. N — целое число; скрипт выполняется, если в результате деления текущего значения на N в остатке 0. Например, скрипт, где для часов задано \*/8, будет выполняться при текущем значении 0, 8 и 16;
- ▶ N — выполнять скрипт в точно указанную минуту, час и день N;
- ▶ N–K — выполнять скрипт, если значение минут, часа и дня от N до K (включительно);
- ▶ N, K — можно указать несколько параметров N и N–K (через запятую). Например, если в скрипты задано 15, 50–52 минуты, он выполняется при текущем значении 15, 50, 51 и 52.

---

### 2.1.2. Сценарии, основанные на событиях

Сценарии, основанные на событиях, могут использоваться для задания логики обработки событий, связанных с групповым адресом. Функция, установленная пользователем, выполняется при наступлении события *group write* для указанного группового адреса. Данные о событии хранятся в глобальной переменной типа *event*. Значения переменной:

- ▶ *dstraw (integer)* – закодированный групповой адрес назначения;
- ▶ *srcraw (integer)* – закодированный индивидуальный адрес источника;
- ▶ *dst (string)* – расшифрованный групповой адрес назначения (пример: 1/1/4);
- ▶ *src (string)* – расшифрованный индивидуальный адрес источника (пример: 1.1.2);
- ▶ *type (string)* – тип события: *groupwrite*, *groupread* или *groupresponse*. В настоящее время пользовательские скрипты привязаны исключительно к событиям *groupwrite*;
- ▶ *dataraw (integer/string)* – закодированные двоичные данные;
- ▶ *datahex (string)* – данные в виде шестнадцатеричной кодированной строки; используются для преобразования значения в переменную Lua.

*Внимание! Переменная event доступна только для функций, основанных на событиях, и недоступна для резидентных и запланированных функций.*

*Внимание! Все основанные на событиях скрипты выполняются в рамках единой очереди. Убедиться, что скрипты, основанные на событиях, не содержат бесконечные циклы, команды на переход в ждущий режим и иные блокирующие команды.*

**Текстовое поле включает приведенные ниже колонки, которые могут быть отсортированы в автоматическом режиме по соответствующему признаку (при выборе):**

*Script name* – имя скрипта.

*Event group address* – значение группового адреса.

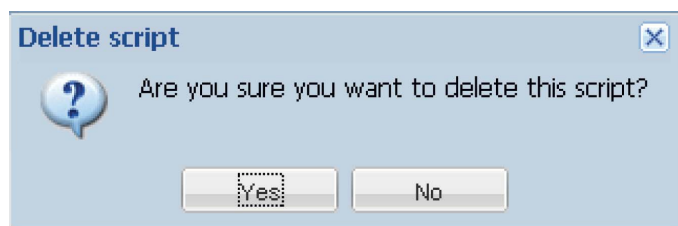
*Description* – описание скрипта.

*Category* – имя группы, к которой относится скрипт.

*Editor*  – возможность вносить изменения в скрипт с фрагментами кода в редакторе скриптов.

*Active* – включенный (зеленый) / выключенный (красный) скрипт.

 – пиктограмма появляется справа от колонки *Active* после добавления скрипта. При нажатии на пиктограмму скрипт будет удален. При этом выводится окно с требованием подтвердить удаление.



---

### 2.1.3. Резидентные скрипты

Резидентные скрипты выполняются бесконечное число раз. Скрипты выключаются после каждого вызова; выполнение скриптов возобновляется после истечения установленного времени задержки.

*Внимание! Хотя резидентные скрипты выполняются одновременно, они не должны содержать бесконечные циклы; при этом возможность повторной загрузки скриптов после редактирования отсутствует.*

**Текстовое поле включает приведенные ниже колонки, которые могут быть отсортированы в автоматическом режиме по соответствующему признаку (при выборе):**

*Script name* – имя скрипта.


*Script sleep interval* – период времени, по истечении которого скрипт выполняется повторно.

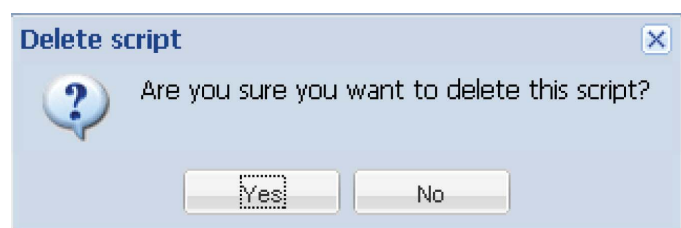
*Description* – описание скрипта.

*Category* – группа, к которой относится скрипт.

*Editor*  – возможность вносить изменения в скрипт с фрагментами кода в редакторе скриптов.

*Active* – включенный (зеленый) / выключенный (красный) скрипт.

 – пиктограмма появляется справа от колонки *Active* после добавления скрипта. При нажатии на пиктограмму скрипт будет удален. При этом выводится окно с требованием подтвердить удаление.



### 2.1.4. Запланированные скрипты

Запланированные скрипты выполняются, когда системное время совпадает с заданным временем запуска скрипта.

Запланированные скрипты выполняются только раз после каждого срабатывания таймера.

**Текстовое поле включает приведенные ниже колонки, которые могут быть отсортированы в автоматическом режиме по соответствующему признаку (при выборе):**

*Script name* – имя скрипта.


*Start at (cron format)* – период времени, по истечении которого скрипт выполняется повторно.

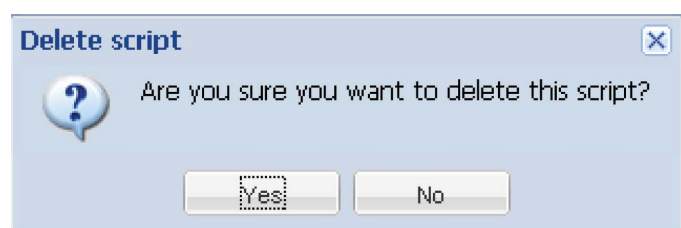
*Description* – описание скрипта.

*Category* – группа, к которой относится скрипт.

*Editor*  – возможность вносить изменения в скрипт с фрагментами кода в редакторе скриптов.

*Active* – включенный (зеленый) / выключенный (красный) скрипт.

 – пиктограмма появляется справа от колонки *Active* после добавления скрипта. При нажатии на пиктограмму скрипт будет удален. При этом выводится окно с требованием подтвердить удаление.



### 2.1.5. Добавление нового скрипта

После нажатия на стрелку в нижней части кнопки *Event-based*, *Resident* или *Scheduled* выводятся два пункта меню:

*List view* – отобразить список скриптов.

*Add new script* – добавить в список новый скрипт



#### Добавление нового скрипта

При добавлении нового скрипта следует заполнить следующие поля:

*Script name* – имя скрипта.

*Event group address* – использовать только цифровые значения от 0 до 9 и / в качестве разделителя.

❗ в правой части текстового поля означает, что используется неверный адрес. Верный групповой адрес (пример): 1/1/1.

*Active* – включенный (зеленый) / выключенный (красный) скрипт.

*Category* – новая или существующая группа скриптов для сохранения нового скрипта. Данный параметр не влияет на выполнение скрипта и помогает сортировать скрипты, а также просматривать их по группам на странице *Tools > Print script listings*.

*Description* – описание скрипта.

*Script source* – исходный код скрипта. Данное поле может быть заполнено вручную или оставлено незаполненным. Исходный код скрипта можно написать позже с использованием *Script Editor*.

Resident script

Script name: Presence simulator

Script sleep interval: 20

Active: ☒

Category: Presence

Description: Synchronizes 0/0/2 value with 0/0/1

Script source:

```
-- if object exists "presence" variable will be a table, nil otherwise
presence = knxobject.get('address', '0/0/1')

-- check that object exists and data has been decoded
if presence and presence.decoded then
  -- result will be either "value = true" or value = "false"
  alert('value = ' .. tostring(presence.data))

  -- update 0/0/2 with the same data
  knxobject.write('0/0/2', presence.data, dt.bool)
else
  alert('read error')
end
```

Save Cancel

## 2.1.6. Инструменты

*User function library* – библиотека пользовательских функций.

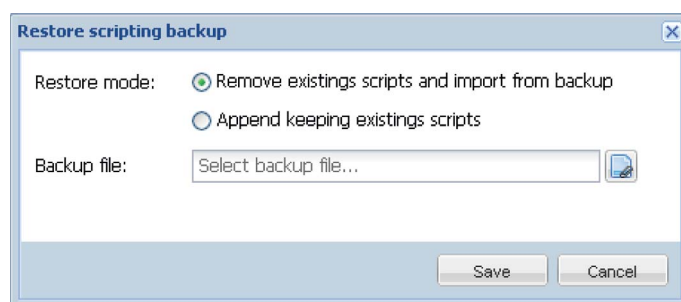
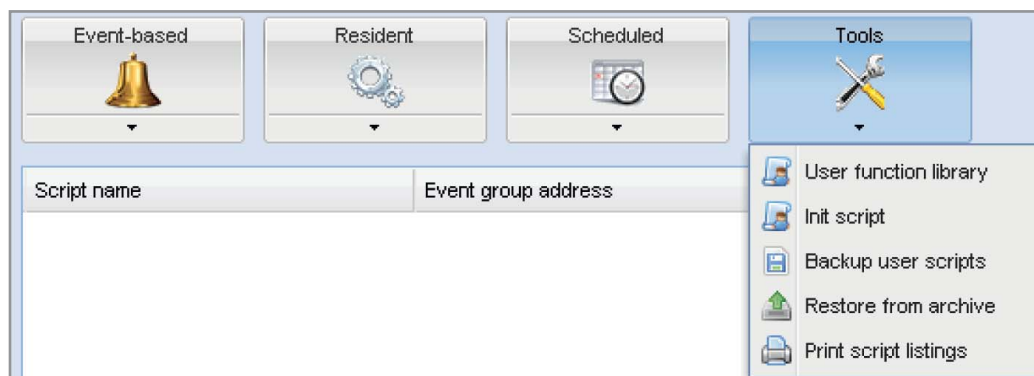
*Init script* – запустить инициализирующий скрипт.

*Backup user scripts* – резервное копирование всех скриптов в файле \*.gz.

*Restore from archive* – копирование скрипта из архива (\*.gz); доступно две функции:

- ▶ удалить все скрипты и импортировать из резервной копии;
- ▶ добавить к существующим скриптам (s).

*Print script listings* – отображаются все группы скриптов с кодами (в виде списка).



### Category: Presence

#### Presence simulator (id: 1)

Type: Resident

Active: Yes

Script sleep interval: 20

Synchronizes 0/0/2 value with 0/0/1

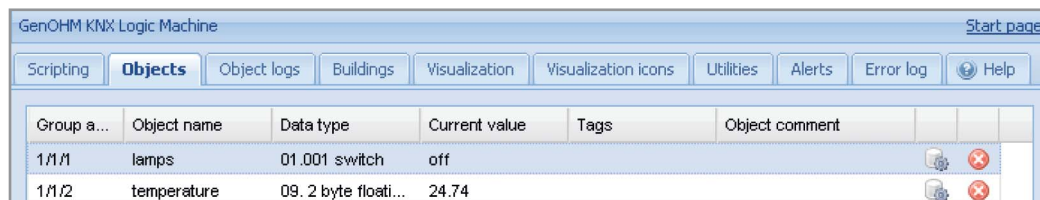
```
-- if object exists "presence" variable will be a table, nil otherwise
presence = knxobject.get('address', '0/0/1')

-- check that object exists and data has been decoded
if presence and presence.decoded then
  -- result will be either "value = true" or value = "false"
  alert('value = %s', tostring(presence.data))

  -- update 0/0/2 with the same data
  knxobject.write('0/0/2', presence.data, dt.bool)
else
  alert('read error')
end
```

## 2.2. Объекты

Перечень объектов сети KNX отображается в меню *Objects*. Объекты попадают в перечень после перехвата телеграмм с шины от неизвестных групповых адресов. В случае перехвата телеграммы объект отображается с неопределенным типом данных и может использоваться при визуализации.

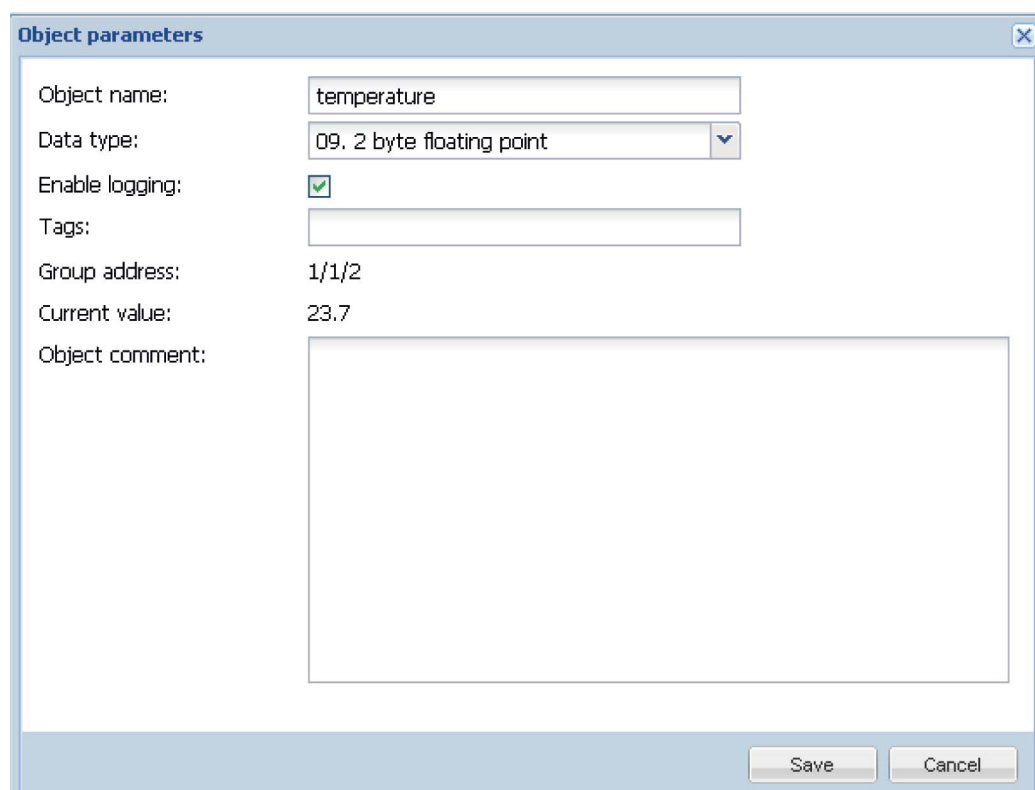


The screenshot shows the 'Objects' menu in the GenOHM KNX Logic Machine. It contains a table with the following data:

Group a...	Object name	Data type	Current value	Tags	Object comment
1/1/1	lamps	01.001 switch	off		
1/1/2	temperature	09. 2 byte floati...	24.74		

### 2.2.1. Параметры объектов

Для изменения настроек существующих и новых объектов выбрать соответствующий объект из перечня.



The 'Object parameters' dialog box contains the following fields and controls:

- Object name:
- Data type: - Enable logging: ☒
- Tags:
- Group address:
- Current value:
- Object comment:
- Buttons: Save, Cancel

*Object name* – имя объекта.

*Data type* – тип данных KNX объекта. Задается после распознавания Logic Machine нового объекта и подтверждения его параметров.

*Enable logging* – регистрация событий для объекта. Журналы отображаются в меню *Objects logs*.

*Tags* – присвоить тэг объекту для последующего создания скриптов (пример: *All\_lights\_first\_floor*).

*Group address* – групповой адрес объекта.

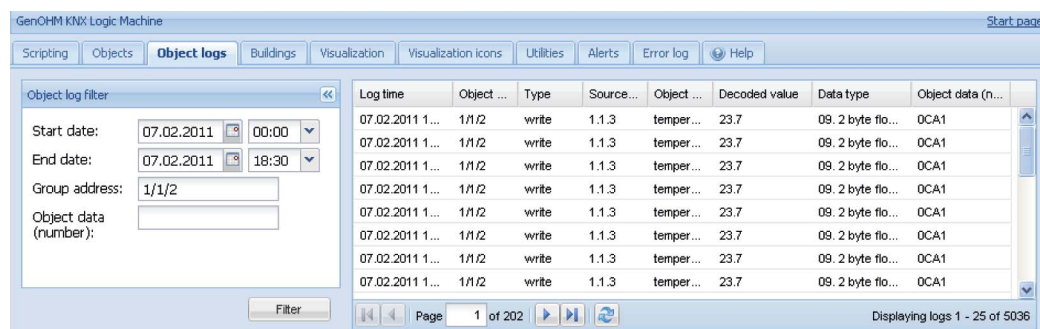
*Current value* – текущее значение объекта.

*Object comment* – примечания для объекта.



## 2.3. Журналы объекта

История телеграмм для соответствующего объекта отображается в меню *Object logs*. После включения данной функции в журнале регистрируются все события.



При необходимости поиска информации за определенный период времени задается фильтр.

*Start date* – дата и время начала фильтрации событий в журнале.

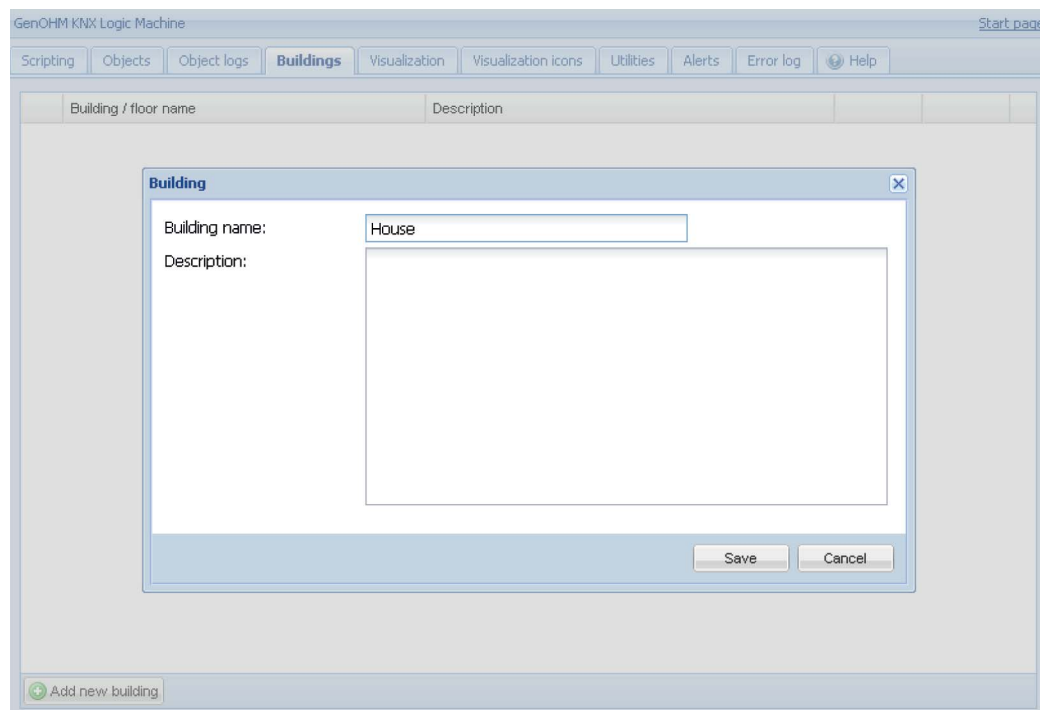
*End date* – дата и время окончания фильтрации событий в журнале.

*Group address* – групповой адрес.



*Object data (number)* – данные объекта.

## 2.4. Здания

В меню Building задается порядок отображения данных, загружаются фоновые изображения.



Нажать *Add new building* для добавления нового здания.

После добавления здания можно определить этажи и загрузить изображения этажей здания. Для добавления этажа нажать на зеленую пиктограмму ; для удаления записи нажать на красную пиктограмму .

**Floor**

Building name: House

Floor name: Stage 1

Floor number: 1

Admin only access: ☒

*i* Background image can be uploaded later by clicking "Upload floor plan" icon in building grid view.

Save Cancel

*Floor name* – название этажа.

*Floor number* – номер этажа.

*Admin access only* – установить допуск к данным этажа только для пользователей с правами администратора.

GenOHM KNX Logic Machine

Scripting Objects Object logs **Buildings** Visualization Visualization icons Utilities Alerts Error log Help

Building / floor name	Description		
House		+	-
• Stage 1	Floor number 1	+	-
• Stage 2	Floor number 2	+	-

Add new building

Version: 20101207 © Embedded Systems 2010

Нажать на , чтобы добавить новое фоновое изображение для этажа; выводится следующее окно:

**Upload floor plan image**

Current floor plan: —

New floor plan:  No file chosen

*i* Leave floor plan file upload field empty if you want to delete the old floor plan without uploading new one.

*i* Supported image types are: BMP, GIF, JPEG, PNG.

Save Cancel

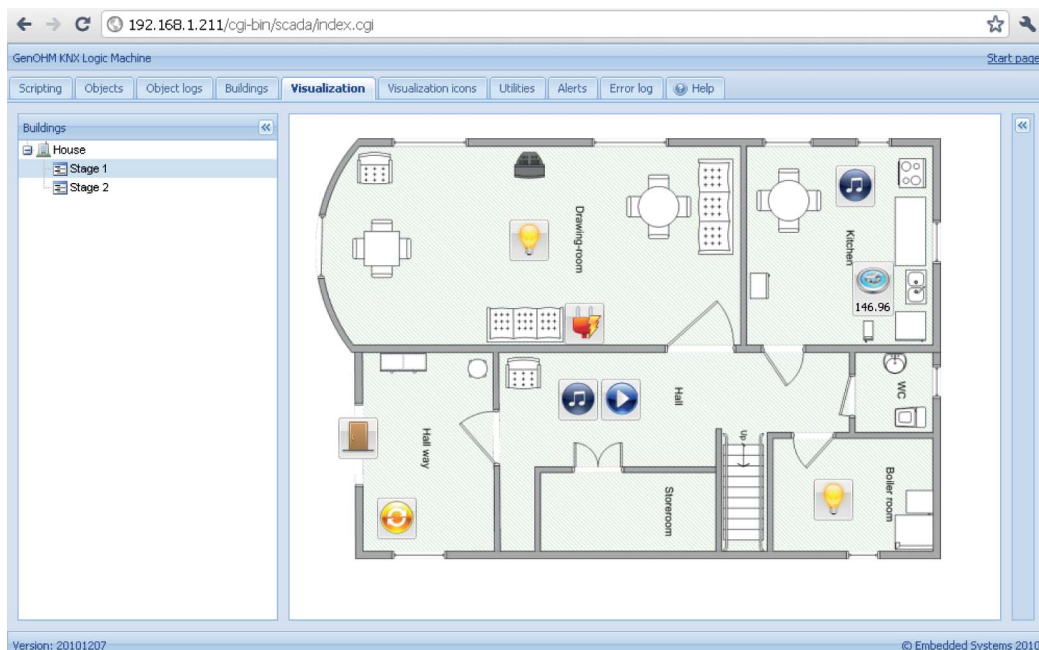
Для загрузки можно использовать изображения в следующих форматах: BMP, GIF, JPEG и PNG.


Для удаления изображения, загруженного для соответствующего этажа, не заполнять поле загрузки.


Для удаления выбрать сохраненный этаж и нажать .


## 2.5. Графическое отображение


После ввода параметров здания и этажей во вкладке *Buildings* в меню *Visualization* выводится графическое изображение данных. На данном этапе имеется возможность добавлять и управлять контролируруемыми объектами.




Нажать , чтобы убрать боковые вкладки; масштаб изображения увеличивается. Данная функция в особенности полезна при просмотре изображения на небольших экранах.

**Floor plan editor** 

Object 


Main object:  


Status object:  


Custom name:


Read-only: ☐

Hide in touch: ☐

Icon:  

Floor link 

Camera 

Graph 

### 2.5.1. Редактор плана этажа здания

Справа от плана этажа появляется окно *Floor plan editor*. Нажать на *Unlock current floor plan for editing* для вывода четырех основных меню и ввода параметров конфигурации:

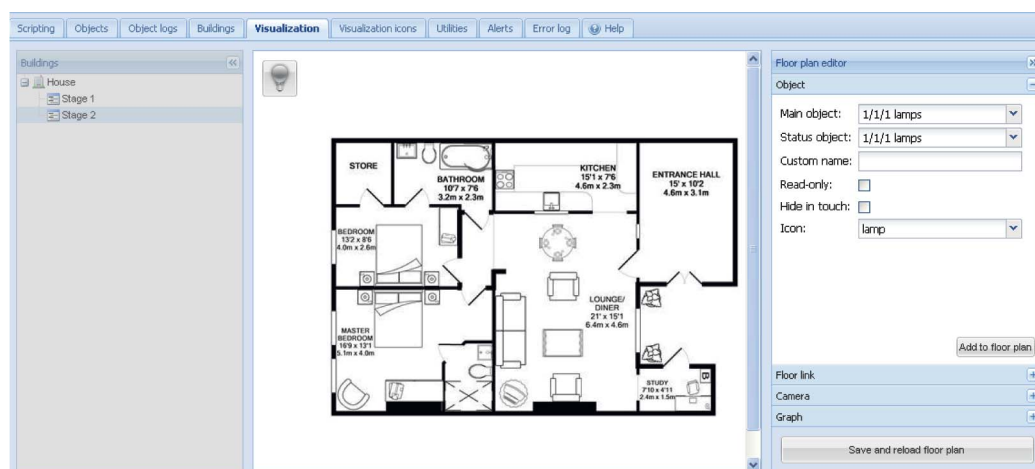
*Object* – новый объект, добавляемый на карту.

*Floor link* – связывание нескольких этажей (используется специальная пиктограмма).

*Camera* – использование сетевой веб-камеры для отображения параметров на плане.

*Graph* – график (в реальном времени) контроля значений масштабированных объектов.

### 2.5.2. Объект



*Main object* – список существующих групповых адресов шины KNX/EIB; данные адреса доступны для редактирования во вкладке *Objects*.

*Status object* – список статусных объектов KNX/EIB.

*Custom name* – имя объекта.

*Read-only* – объект доступен только для чтения; функция редактирования недоступна.

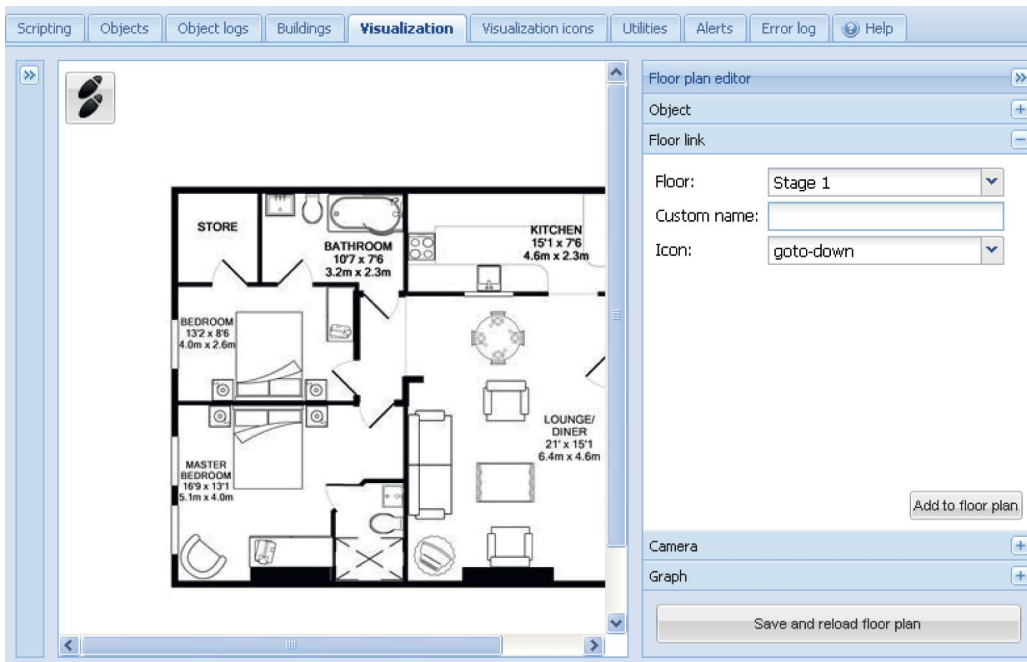
*Hide in touch* – не отображать объект в меню *Touch Visualization*.

*Icon* – перечень доступных пиктограмм (вкладка *Visualization icons*).

После задания параметров объектов нажать *Add to floor plan*; появляется вновь созданный объект. Пользователь может переместить объект в заданное место. Следует помнить, что в режиме редактирования объект выключен. После добавления всех требуемых объектов нажать *Save and reload floor plan*; объекты будут включены.

### 2.5.3. Связывание этажей

Для более наглядного представления плана используется функция связывания этажей. Пользователь имеет возможность добавить на план определенные пиктограммы, представляющие собой ссылки на другие этажи.



*Floor* – название этажа, к которому относится ссылка.

*Custom name* – имя ссылки.

*Icon* – отображаемая на карте пиктограмма.

После ввода параметров для связывания этажей нажать *Add to floor plan*; появляется вновь созданный объект. Пользователь может переместить объект в заданное место. Следует помнить, что в режиме редактирования объект выключен. После добавления всех требуемых объектов нажать *Save and reload floor plan*; объекты будут включены.

### 2.5.4. Камера

Logic Machine поддерживает сетевые IP веб-камеры любых производителей, которые могут использоваться для создания плана этажа.



*Source url* – адрес источника потока видеоданных.

*Width* – ширина подокна для просмотра изображения.

*Height* – высота подокна для просмотра изображения.

*Custom name* – имя объекта.

После ввода параметров работы камеры нажать *Add to floor plan*; объект появится на плане этажа. Пользователь может переместить объект в заданное место. Следует помнить, что в режиме редактирования объект выключен. После добавления всех требуемых объектов нажать *Save and reload floor plan*; объекты будут включены. Нажать на видеокамеру для вывода подокна с картинкой, фиксируемой с использованием сетевой IP веб-камеры. Окно можно установить таким образом, чтобы не загромождать другие доступные функции.

### 2.5.5. График

Для контроля текущих и предыдущих значений масштабированных объектов в рамках плана этажа используются планы реального времени.

*Data object* – групповой адрес объекта.

*Custom name* – имя объекта.

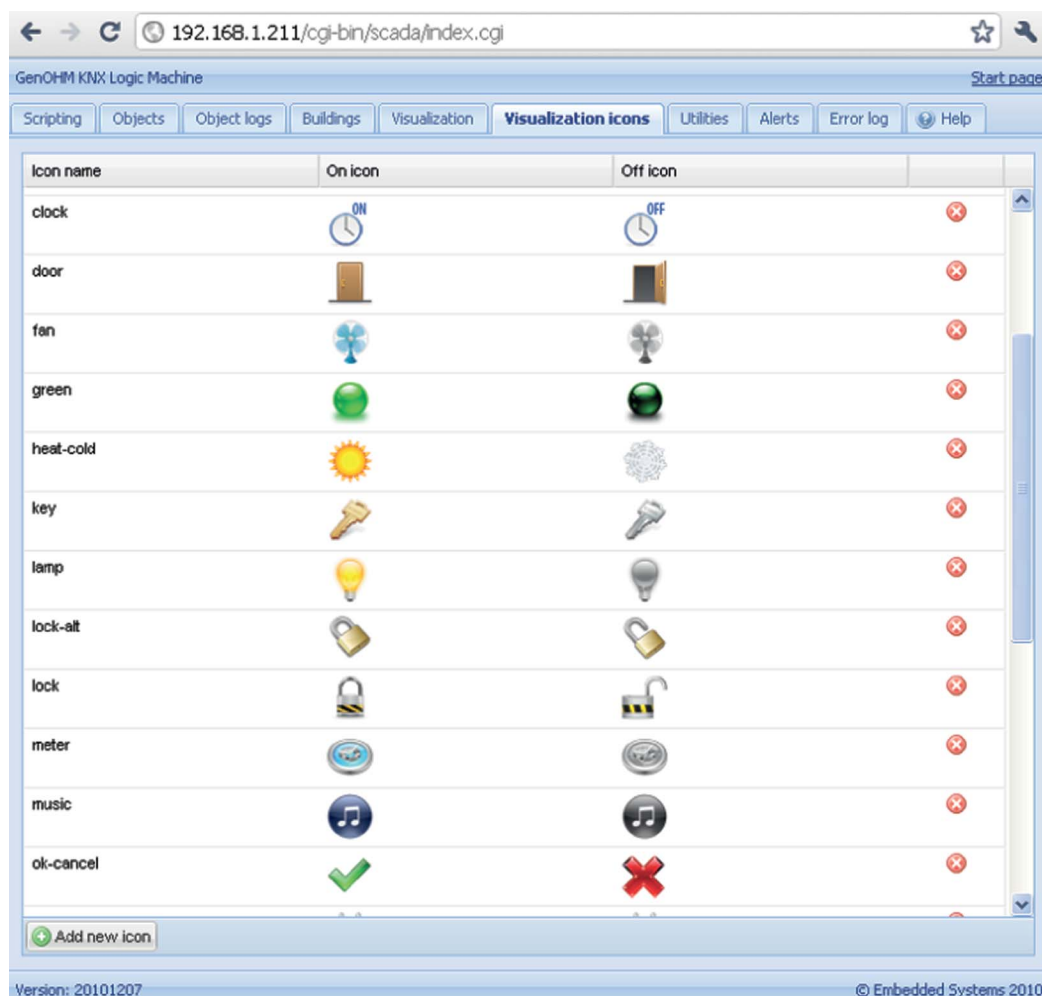
*Icon* – пиктограмма для включения функции отображения плана.

*Width* – ширина подокна для отображения плана.

*Height* – высота подокна для отображения плана.

### 2.6. Пиктограммы, используемые при создании плана этажа

Для просмотра перечня заранее установленных пиктограмм открыть вкладку *Visualization icons*.



Нажать *Add new icon* и добавить новую запись. По умолчанию используются следующие размеры: 32 x 32 пикселей.

Add new icon

Icon name:

On icon:

Select PNG icon file...

Off icon:

Select PNG icon file...

Only PNG icons can be used. The default size is 32x32px. "On" icon will be used for non-binary objects. Name can contain letters, numbers, underscore and minus sign.

Save

Cancel

*Icon name* – название пиктограммы, отображаемое в перечне после добавления нового объекта. Допускается использовать буквенные символы, цифровые значения, нижнее подчеркивание и знак минус.

*On icon* – пиктограмма для отображения статуса 'On' для объектов, имеющих два состояния; стандартная пиктограмма для не двоичных объектов.

*Off icon* – означает, что двоичный объект выключен.

## 2.7. Утилиты

В данной вкладке доступны следующие утилиты:

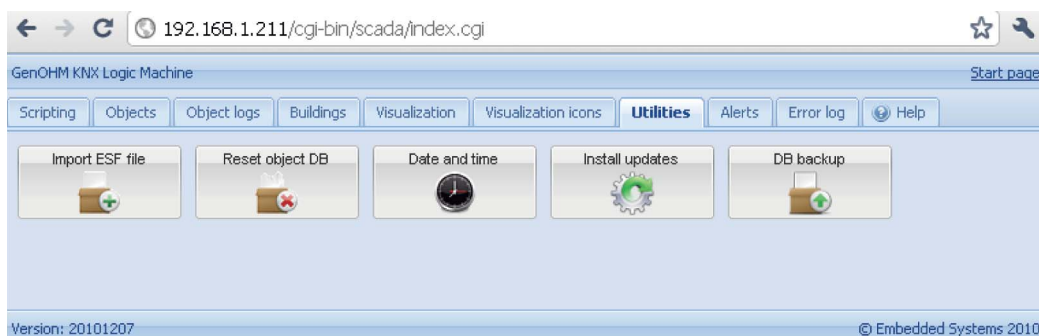
*Import ESF file* – импорт файла объектов ETS. Следует задавать точные типы данных для импортируемых объектов. Существующие объекты не перезаписываются. Объекты, имеющие одно имя, считаются копиями и не импортируются.

*Reset object DB* – удалить все объекты из памяти Logic Machine; объекты также удаляются и с графической карты.

*Date and time* – настройки даты и времени.

*Install updates* – загрузить файл обновления Logic Machine \*.Imu. Logic Machine выполняет перезагрузку после успешного обновления.

*DB backup* – резервное копирование всех объектов, журналов и скриптов. Графические карты резервному копированию не подлежат.





## 2.8. Предупредительные сообщения

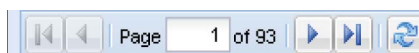
Во вкладке *Alert* содержится перечень заданных с использованием функции *alert* предупредительных сообщений для соответствующих скриптов. Сообщения хранятся на карте памяти.

Alert time	Message
01.01.1970 10:20:42	read error
01.01.1970 10:20:22	read error
01.01.1970 10:20:02	read error
01.01.1970 10:12:58	read error

Page 1 of 93

Displaying alerts 1 - 25 of 2317

Используя панель управления, пользователь имеет возможность перемещаться и перезагружать страницы.



Предупредительное сообщение (пример):

температура = 25.3

if температура > 24 then

-- отображаемое сообщение: 'Temperature levels are too high: 25.3'

alert ('Temperature level is too high: %.1f', temperature)

end

## 2.9. Журнал ошибок

Сообщения об ошибках, обнаруженных в скриптах, отображаются во вкладке *Error log*.

GenOHM KNX Logic Machine

Start page

Scripting Objects Object logs Buildings Visualization Visualization icons Utilities Alerts **Error log** Help

Error time	Script name	Error description

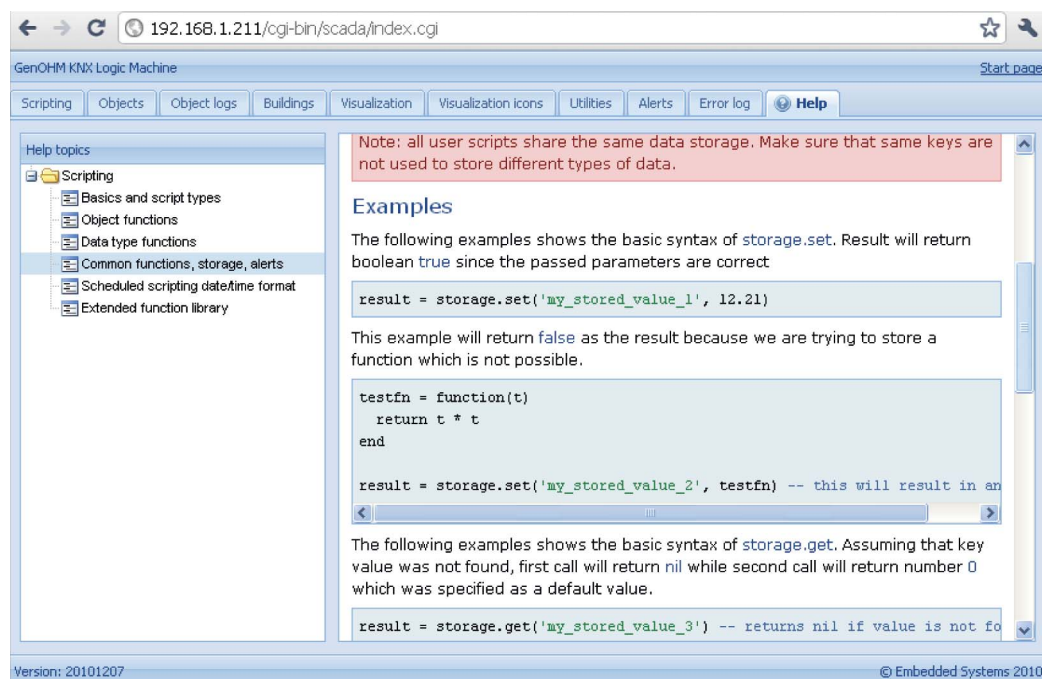
Page 1 of 1

No errors found



## 2.10. Справка

Во вкладке *Help* содержится справочная информация по синтаксису скриптов.

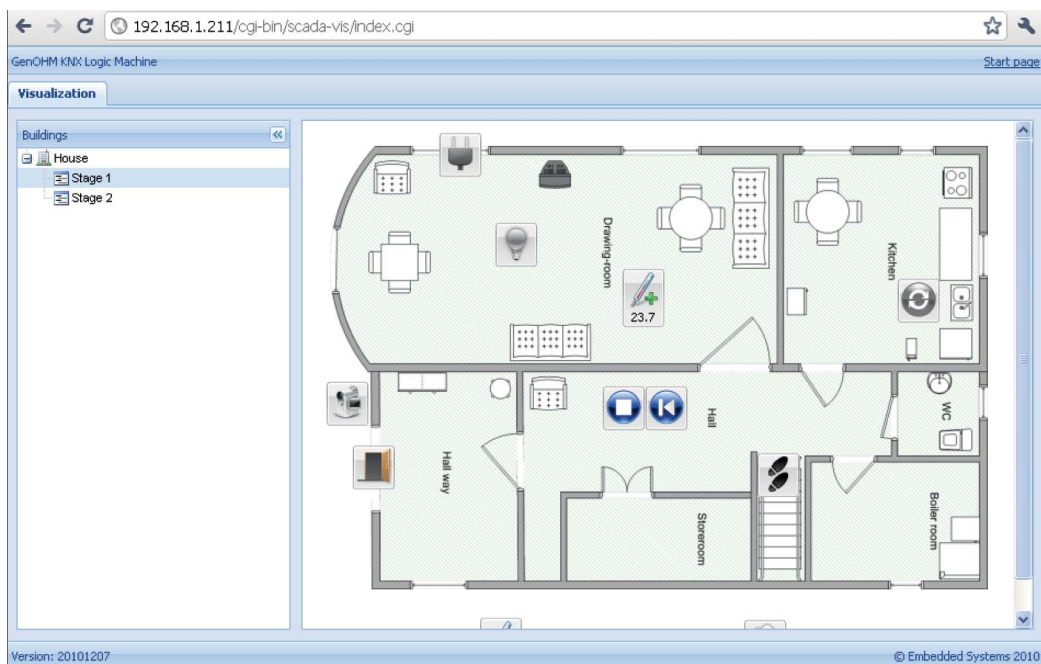


## 3. Графическое отображение параметров для различных уровней доступа

Отображение *User mode visualization*, главным образом, используется главным образом для ограничения доступа пользователей к графическим картам. В данном случае доступ ограничивается исключительно к созданным графическим картам, а не скриптам, журналам и службам.

Logic Machine поддерживает три уровня доступа: чтение, редактирование и администрирование.

Уровень доступа	Имя пользователя	Пароль
Только чтение	visview	visview
Редактирование	viscontrol	viscontrol
Редактирование с правами администратора	visadmin	visadmin



#### 4. Touch visualization – режим графического представления данных с использованием сенсорных устройств

Данный режим предназначен для сенсорных устройств (iPhone/iPod/iPad/Android). Все объекты конфигурации Logic Machine по умолчанию доступны в данном режиме (если не включена функция *Hide in touch*).

Logic Machine поддерживает три уровня доступа: чтение, редактирование и администрирование.

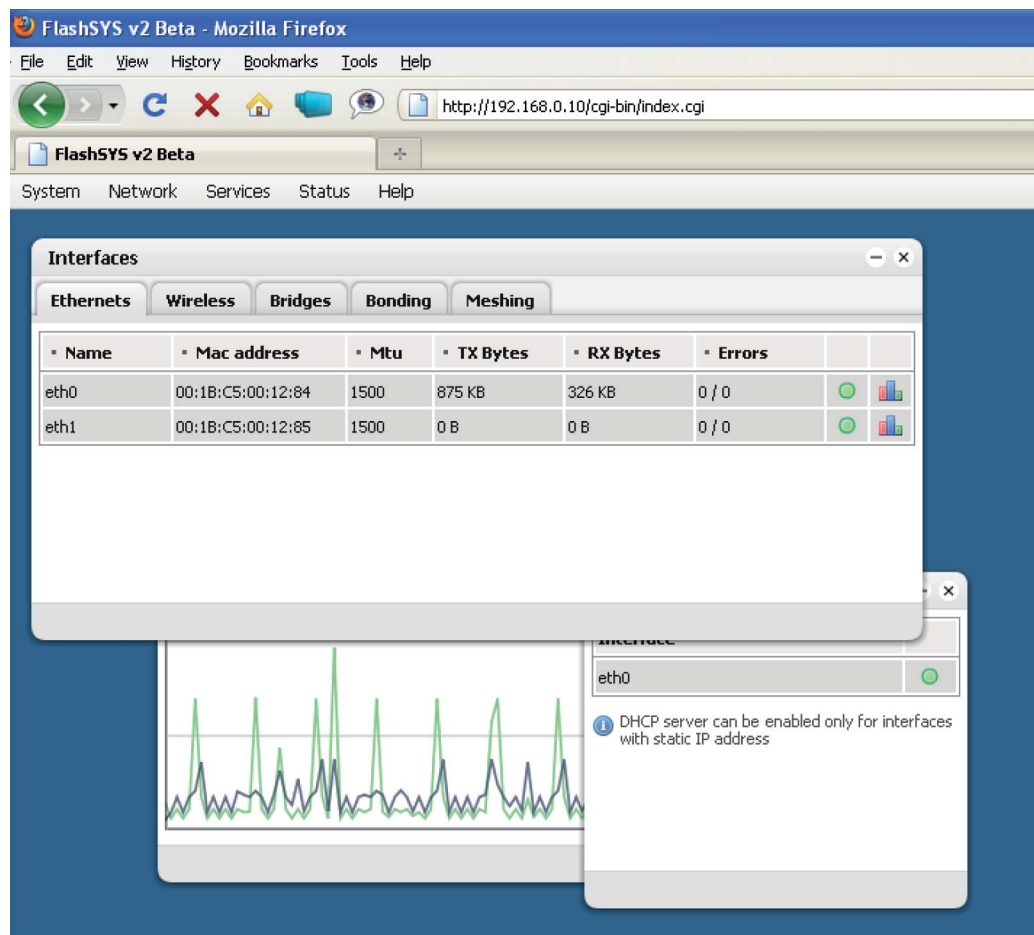
Уровень доступа	Имя пользователя	Пароль
Только чтение	visview	visview
Редактирование	viscontrol	viscontrol
Редактирование с правами администратора	visadmin	visadmin

Главное окно – *Building*; пользователь выбирает этаж и здание для внесения изменений (просмотра). После этого на экран выводится перечень всех объектов, связанных с выбранным этажом; управление осуществляется при помощи круглой пиктограммы справа или слева (для двоичных объектов) или направляющей (для масштабированных объектов).



## 5. Конфигурация сети

Настройка конфигурации сети производится с целью управления маршрутизатором Logic Machine. Большинство функций, описанных в настоящем руководстве, связаны с IP-конфигурацией, соответствующей шине KNX. Другие функции маршрутизатора описаны в руководстве пользователя графического интерфейса FlashSYS ([www.openrb.com](http://www.openrb.com)).



Имя пользователя	Пароль
admin	admin

## 5.1. Смена пароля

Открыть *System > GUI Login* для изменения имени пользователя и пароля. Пользователь имеет возможность сменить пароль для *admin (Logic Machine)* и *Network configuration*, а также для уровней доступа в режиме *Touch* и *User mode visualization* (только чтение; редактирование; редактирование с правами администратора).

The screenshot shows a 'GUI login' dialog box with a close button (X) in the top right corner. It contains four sets of login fields, each with a 'Login' label and a corresponding 'Password' and 'Repeat password' field. The users listed are 'admin', 'visadmin', 'viscontrol', and 'visview'. The password fields are masked with dots. At the bottom, there are 'OK' and 'Cancel' buttons.

Login	Password	Repeat password
admin	.....	.....
visadmin	.....	.....
viscontrol	.....	.....
visview	.....	.....

OK Cancel

## 5.2. Конфигурация Ethernet

На первой вкладке указаны оба интерфейса Ethernet. Пользователь имеет возможность включить (выключить) и контролировать потоки данных при помощи специальных пиктограмм (в правой части окна)

The screenshot shows the 'Interfaces' configuration window with the 'Ethernet' tab selected. It displays a table with columns for Name, Mac address, Mtu, TX Bytes, RX Bytes, and Errors. There are also status icons (green circle and red/blue bar chart) for each interface.

Name	Mac address	Mtu	TX Bytes	RX Bytes	Errors	Status
eth0	00:50:C2:1E:AF:FE	1500	41 MB	29 MB	0 / 0	Green circle, Red/Blue bar chart
eth1	00:50:C2:1E:AF:FD	1500	0 B	0 B	0 / 0	Green circle, Red/Blue bar chart

Для доступа в окно настройки IP-конфигурации выбрать интерфейс, в параметры которого необходимо внести изменения. На примере ниже выбран *eth0*; при этом выводится следующее окно:

The screenshot shows a window titled "Interface eth0" with a "General" tab. The configuration fields are as follows:

Field	Value
Protocol	Static IP
IP address	192.168.1.211
Network mask	255.255.255.0
Gateway IP	192.168.1.1
DNS server	192.168.1.1
Mtu	

*Protocol* – имя протокола для данного интерфейса. Доступные протоколы: статический, DHCP и PPP. По умолчанию для *eth0* установлен протокол DHCP.

## СТАТИЧЕСКИЕ НАСТРОЙКИ IP

*IP address* – статический IP-адрес.

*Network mask* – сетевая маска; по умолчанию: 255.255.255.0 (/24).

*Gateway IP* – IP-адрес шлюза по умолчанию. При наличии данного параметра нет необходимости дублировать значение в окне *Routes*.

*DNS Server* – адрес сервера DNS.

*MTU* – максимальный размер пакета, передаваемого по протоколу обмена данными.

This is another screenshot of the same "Interface eth0" configuration window, showing the same settings as the first image.

Field	Value
Protocol	Static IP
IP address	192.168.1.211
Network mask	255.255.255.0
Gateway IP	192.168.1.1
DNS server	192.168.1.1
Mtu	

## НАСТРОЙКИ ПРОТОКОЛА DHCP

*Current IP* – IP-адрес, присвоенный сервером DHCP. Данное поле доступно только в случае наличия IP-адреса. В противном случае поле недоступно.

*MTU* – максимальный размер пакета, передаваемого по протоколу обмена данными.

*DNS Server* – адрес сервера DNS.

The screenshot shows the 'Interface eth0' configuration window with the 'General' tab selected. The 'Protocol' dropdown is set to 'DHCP'. The 'Current IP' field displays '192.168.1.211'. The 'DNS server' field displays '192.168.1.1'. The 'Mtu' field is empty. At the bottom right, there are 'OK' and 'Cancel' buttons.

## НАСТРОЙКИ ТУННЕЛЯ PPPoE

При необходимости использовать протокол PPPoE для выбранного интерфейса Ethernet задаются соответствующие настройки протокола PPP.

*Username* – имя пользователя для соединения с сервером PPPoE.

*Password* – пароль.

*Keepalive* – время активного подключения.

*Dial on Demand* – подключение по требованию.


*Radius server* – адрес сервера RADIUS для авторизации.

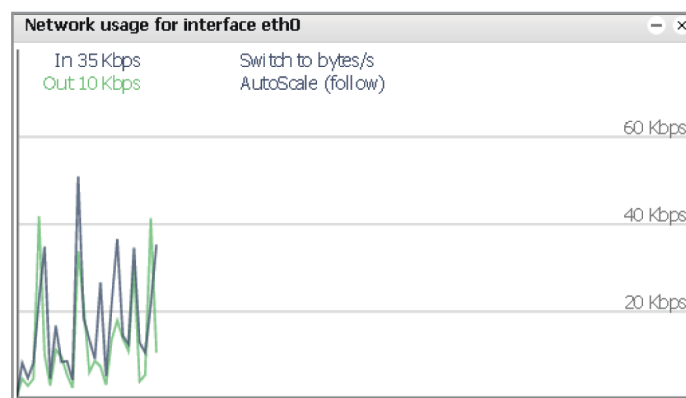
*DNS Server* – адрес сервера DNS.

*MTU* – максимальный размер пакета, передаваемого по протоколу обмена данными.

The screenshot shows the 'Interface eth0' configuration window with the 'General' tab selected. The 'Protocol' dropdown is set to 'PPPoE'. The 'Username' and 'Password' fields are empty. The 'Keepalive' field is empty. The 'Dial on Demand' field is empty. The 'Radius server' field is empty. The 'DNS server' field displays '192.168.1.1'. The 'Mtu' field is empty. At the bottom right, there are 'OK' and 'Cancel' buttons.

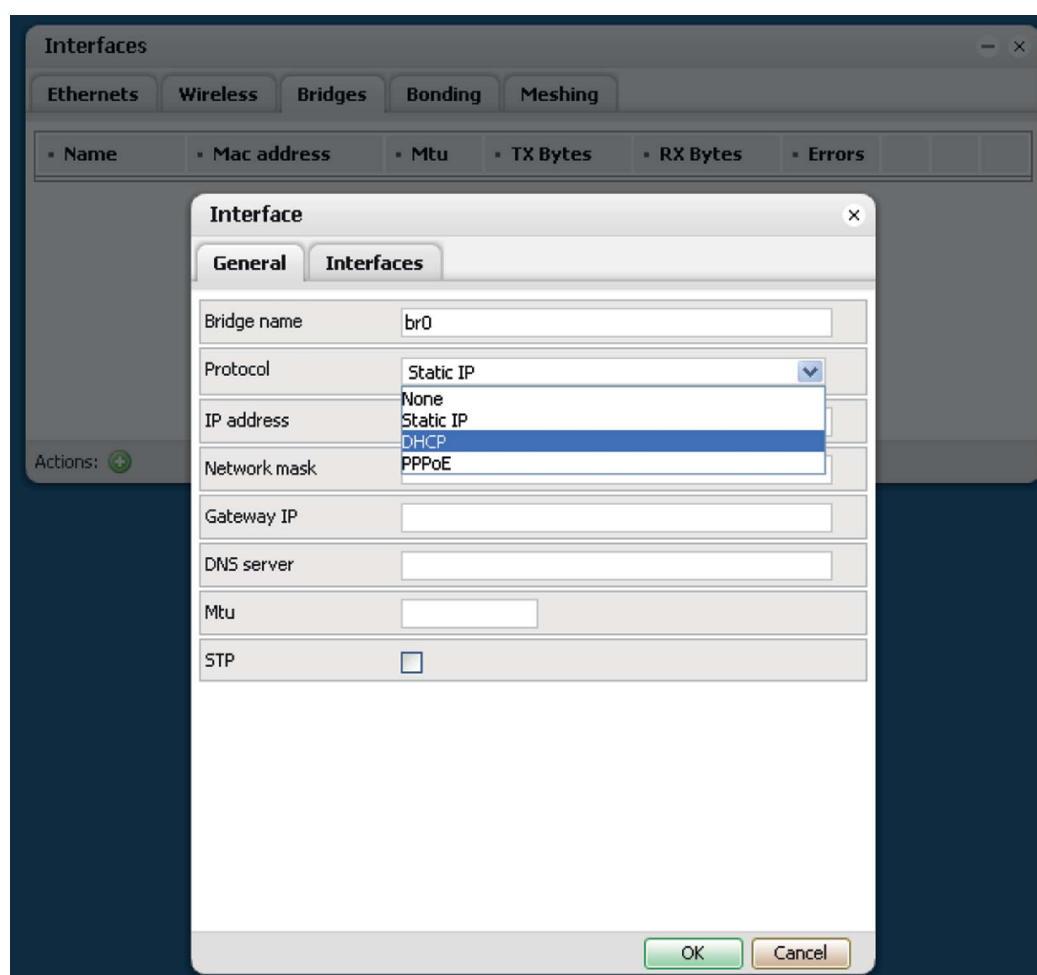
## ГРАФИК НАГРУЗКИ НА ИНТЕРФЕЙС ETHERNET

В главном окне вкладки Ethernets после нажатия на  выводится новое окно, где в режиме реального времени выстраивается график нагрузки на интерфейс (ввод/вывод). Доступно две единицы измерения: кбайт/сек. и байт/сек.



### 5.3. Мосты

Несколько интерфейсов могут быть объединены при помощи одного моста. При подключении интерфейса к мосту в данном меню задаются настройки протокола.



*Bridge name* – название мостового интерфейса.

*Protocol [None / Static IP / DHCP / PPPoE]* – используемый протокол.

*IP address* – статический IP-адрес.

*Network mask* – сетевая маска.

*Gateway IP* – IP-адрес шлюза по умолчанию. Если адрес задан, дополнительные параметры в окне *Routes* не указываются.

*DNS Server* – адрес сервера DNS.

*MTU* – максимальный размер пакета, передаваемого по протоколу обмена данными.

*STP* – позволяет протоколу связующего дерева избегать замкнутых циклов в мостовой схеме.

Во вкладке *Interfaces* содержится перечень интерфейсов; пользователь выбирает интерфейсы, объединяемые в мостовую схему.

## 5.4. Таблица маршрутизации

Таблица маршрутизации системы находится в меню *Network > Routes*. Окно разделено на две части, где указаны статические и динамические маршруты.

Routes				
Dynamic Static				
Interface	Destination	Gateway	Network mask	Flags
ath0	192.168.2.0	*	255.255.255.0	U
eth0	192.168.1.0	*	255.255.255.0	U
eth0	224.0.0.0	*	224.0.0.0	U
eth0	default	192.168.1.1	0.0.0.0	UG


*Interface* – название интерфейса.

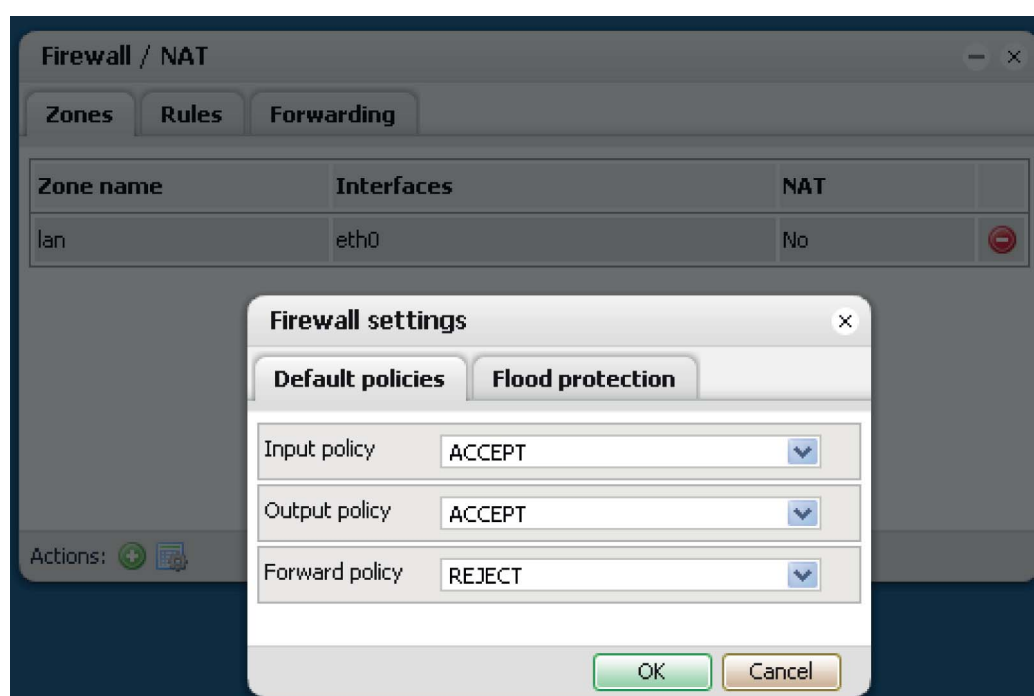
*Destination* – IP-адрес назначения.

*Network mask* – сетевая маска.

*Gateway* – IP-адрес шлюза.

## 5.5. Настройки системы защиты доступа (Firewall rules)

Настройка системы защиты доступа выполняется в окне *Network > Firewall*. Пользователь имеет возможность задавать настройки, обеспечивающие прием, отклонение и переадресацию тех или иных пакетов данных. Нажать на  для задания основных настроек системы защиты доступа.



В дальнейшем пользователь имеет возможность изменять значения по умолчанию и настройки защиты от избыточного трафика.



## Зоны

Пользователь может задавать несколько зон системы защиты доступа и предусматривать для них разные настройки. Вышеупомянутые зоны используются системой защиты доступа, в том числе при переадресации (NAT).

*Zone name* – название зоны.

*Input policy* – порядок обработки пакетов данных, принимаемых в данной зоне.

*Output policy* – порядок обработки пакетов данных, отправляемых из данной зоны.

*Forward policy* – порядок обработки пакетов данных, проходящих через данную зону.

*NAT* – включить/выключить функцию преобразования сетевых адресов для данной зоны.

The 'Firewall zone' dialog box has two tabs: 'Zone settings' and 'Interfaces'. The 'Zone settings' tab is active. It contains the following fields:

- Zone name:
- Input policy:  (dropdown arrow)
- Output policy:  (dropdown arrow)
- Forward policy:  (dropdown arrow)
- NAT: ☐

At the bottom are 'OK' and 'Cancel' buttons.

The 'Firewall zone' dialog box has two tabs: 'Zone settings' and 'Interfaces'. The 'Interfaces' tab is active. It contains a table with the following data:

Interface	Checkbox
ath0	<input type="checkbox"/>
eth1	<input type="checkbox"/>
eth0	<input type="checkbox"/>

At the bottom are 'OK' and 'Cancel' buttons.

## НАСТРОЙКИ

The 'Firewall / NAT' configuration window has three tabs: 'Zones', 'Rules', and 'Forwarding'. The 'Rules' tab is active. It shows a table with columns: 'Nr', 'Rule type', 'Source zone', and 'Dest zone'. A 'Firewall rule' dialog box is open over the table. The dialog box contains the following fields:

- Rule type:  (dropdown arrow)
- Source zone:  (dropdown arrow)
- Source IP:
- Source ports:
- Dest zone:  (dropdown arrow)
- Dest IP:
- Dest ports:
- Protocol:  (dropdown arrow)

At the bottom of the dialog box are 'OK' and 'Cancel' buttons.

*Rule type* – тип настроек.

*Forward* – данные переадресовываются на указанный IP-адрес/порт.

*Accept* – прием данных.

*Drop* – отказ в приеме данных.

*Source zone* – название зоны источника.

*Source IP* – IP-адрес источника.

*Source ports* – порты источника. Допускается ввод одной строкой через запятую.

*Destination zone* – название зоны назначения.

*Dest. IP* – IP-адрес назначения.

*Dest. Ports* – порты назначения. Допускается ввод одной строкой через запятую.

*Protocol* – используемый протокол передачи данных.

*Both* – используются протоколы TCP и UDP.

*TCP* – используется протокол TCP.

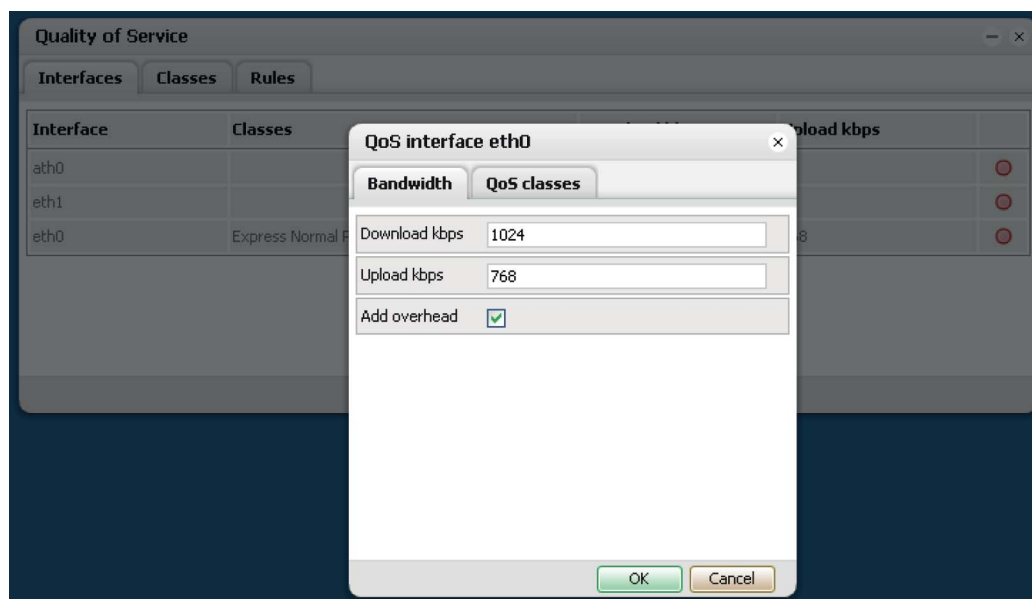
*UDP* – используется протокол UDP.

## ПЕРЕАДРЕСАЦИЯ

В случае использования нескольких зон пользователь может включать переадресацию с одной зоны на другую.

## 5.6. Параметры резервирования каналов

Параметры резервирования каналов задаются в окне *Network > Quality of Service*. Пользователь имеет возможность ограничить ширину пропускания канала для того или иного интерфейса, IP-адреса или приложения, задать приоритеты для данных отдельных типов и т. п.



## ИНТЕРФЕЙСЫ

Выбрать интерфейс для ограничения ширины пропускания канала и присвоения класса резервирования.

*Download kbps* – скорость загрузки, заданная для интерфейса.

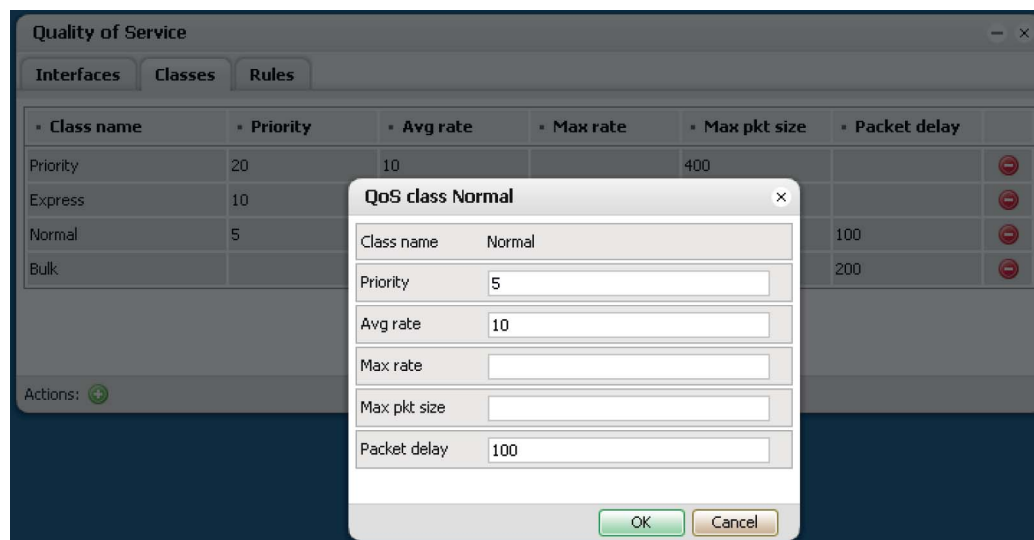
*Upload kbps* – скорость выгрузки, заданная для интерфейса.

*Add overhead* – дополнительные данные.

Во вкладке *QoS classes* пользователь задает классы резервирования и присваивает их интерфейсам.

## КЛАСС РЕЗЕРВИРОВАНИЯ

По умолчанию для Logic Machine задано несколько классов резервирования с предварительно установленными параметрами: *Priority*, *Express*, *Normal* и *Bulk*. Пользователь может добавлять другие классы на свое усмотрение.



*Class name* – название класса.

*Priority* – указатель приоритета.

*Avg rate* – средняя пропускная способность.

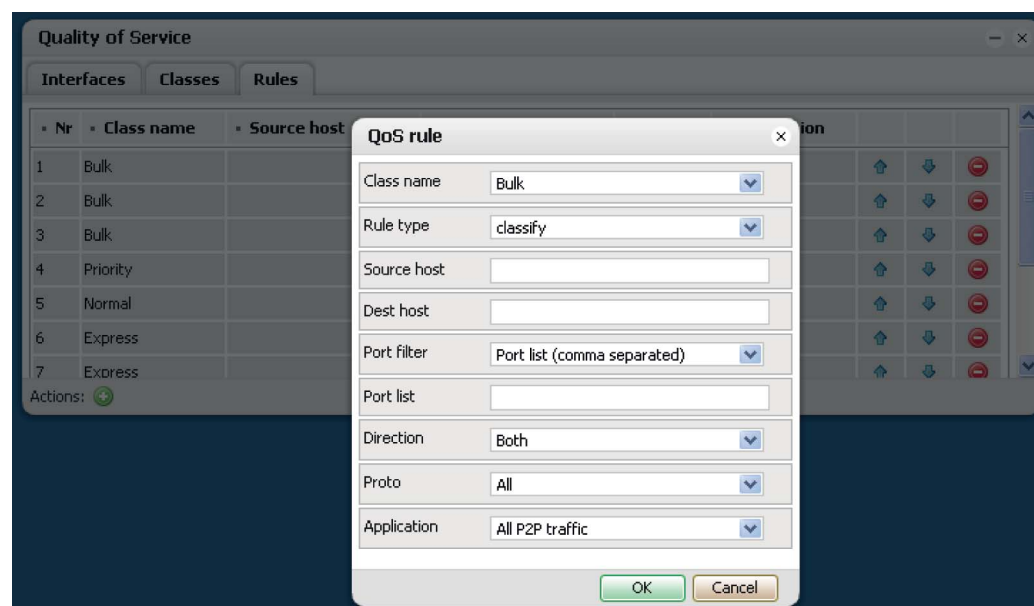
*Max rate* – максимальная пропускная способность.

*Max pkt size* – максимальный размер пакета.

*Packet delay* – задержка пакета.

## НАСТРОЙКИ

Настройки задаются во вкладке *Rules* в зависимости от IP-адреса, приложения, порта и класса.



*Class name* – название предварительно заданного класса.

*Rule type* [classify/reclassify/default] – тип настройки.

*Source host* – IP-адрес источника.

*Dest host* – IP-адрес назначения.

*Port filter [Port list / Port range]* – тип фильтра для порта.

*Port list* – перечень портов через запятую.



*Port range* – диапазон портов.

*Direction [Both/Inbound/Outbound]* – направление.

*Proto [All/TCP/UDP/ICMP]* – протокол.

*Application [All P2P traffic / No filter]* – специальный фильтр для приложения.

## 5.7. Сервер DHCP

Пользователь может добавлять несколько серверов DHCP (по одному для каждого интерфейса). Во вкладке Interfaces отображаются доступные интерфейсы для привязывания и настройки сервера DHCP. Доступными считаются интерфейсы с заданными статическими IP-настройками.  и  используются для включения и выключения службы (соответственно).

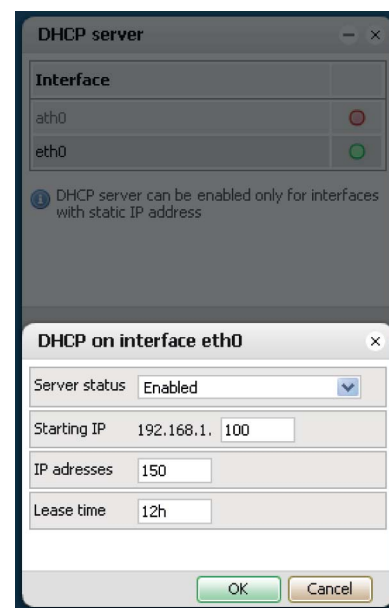
Для доступа в окно настройки сервера выбрать интерфейс из списка.

*Server status* – включен/выключен.

*Starting IP* – IP-адрес, с которого начинается пул IP-адресов сервера DHCP.

*IP addresses* – количество IP-адресов в пуле сервера DHCP.

*Lease time* – время, в течение которого клиент может пользоваться заданным IP-адресом.






## 5.8. Состояние системы

В окне *System status* отображаются общие параметры работы Logic Machine, коэффициент загрузки ЦП и размер неиспользованной памяти.

System status	
General Memory usage Partitions	
Parameter	Value
CPU clock speed	266.666400MHz
CPU BogoMips	533.33
Linux kernel version	2.6.32.10
System uptime	0d 3h 12m
Load averages	0.04 0.02 0.00

## 5.9. Состояние сети

В окне *Network status* отображаются общие параметры сети, IP-настройки и информация о переданных данных.

Network status							
Name	Mac address	IP address	Mtu	TX Bytes	RX Bytes	Errors	
ath0	00:80:48:52:D3:BB	192.168.2.211	1500	0 B	0 B	0 / 0	
eth0	00:50:C2:1E:AF:FE	192.168.1.211	1500	2 MB	4 MB	0 / 0	
eth1	00:50:C2:1E:AF:FD		1500	0 B	0 B	0 / 0	

## 5.10. Сетевые службы

Службы *ping* и *traceroute* доступны в окне *Network utilities*. Допускается ввод IP-адреса или имени сервера DNS.

Network utilities

Ping

Traceroute

Network performance

IP / Hostname192.168.1.1

PING 192.168.1.1 (192.168.1.1): 56 data bytes  
64 bytes from 192.168.1.1: seq=0 ttl=64 time=3.545 ms  
64 bytes from 192.168.1.1: seq=1 ttl=64 time=0.312 ms  
64 bytes from 192.168.1.1: seq=2 ttl=64 time=0.352 ms  
64 bytes from 192.168.1.1: seq=3 ttl=64 time=0.333 ms  
64 bytes from 192.168.1.1: seq=4 ttl=64 time=0.325 ms  
  
--- 192.168.1.1 ping statistics ---  
5 packets transmitted, 5 packets received, 0% packet loss  
round-trip min/avg/max = 0.312/0.973/3.545 ms

OK

Cancel

## 5.11. Перезагрузка Logic Machine

Выбрать *System > Reboot system*. Выводится следующее окно:

The page at 192.168.1.211 says:

Reboot system now?

OK

Cancel