

# Руководство пользователя WebНМІ

# Содержание

## Статьи

<b>Назначение и применение</b>	<b>1</b>
Назначение и применение	1
<b>Подготовка к первому включению</b>	<b>4</b>
Описание внешних разъемов	4
Подключение внешних устройств	5
Первое включение	8
<b>Настройка сетевых подключений</b>	<b>12</b>
Настройка сетевых соединений	12
Подключение к Reorle.net	25
Подключение к МТС Коннект	29
<b>Системные настройки</b>	<b>34</b>
Системные настройки и сервис	34
Синхронизация времени	42
Сброс настроек	43
<b>Взаимодействие с другими устройствами и сбор данных</b>	<b>44</b>
Соединения	44
Поддерживаемые протоколы	47
Работа с регистрами	47
События	58
<b>Отображение данных</b>	<b>62</b>
Приборные панели	62
Тренды	68
Исторические графики	70
<b>Дополнительные функции</b>	<b>71</b>
Настройка виртуального UART	71
Функция Modbus/TCP сервер	76
<b>Взаимодействие через API</b>	<b>78</b>
Описание API	78

API - Список соединений	79
API - Список регистров	82
API - Получение лога регистров	93
API - Получение текущих значений регистров	98
API - Запись нового значения в регистр	100
API - Список словарей	101
API - Список трендов	103
API - Список графиков	105
API - Список изображений	107
API - Получение данных для графика	108
API - Список событий	113
API - Получение данных для события	117
API - Получение данных о локальном времени	124
API - Список панелей	125
API - Список блоков панелей	127
Пример доступа к данным из Excel	130
Пример доступа к данным из C/C++	131
<b>Типовые решения</b>	<b>134</b>
Подсчет количества упаковок	134
<b>Примечания</b>	
Источники и основные авторы	138
Источники, лицензии и редакторы изображений	139

---

# Назначение и применение

---

## Назначение и применение

---

### Назначение

WEBHMI представляет собой устройство, позволяющие пользователю эффективно решить задачу мониторинга состояния различных инженерно-технических систем и объектов, а также удаленного управления ими. С одной стороны, WEBHMI имеет встроенную поддержку популярных протоколов (таких, как ModBus), принятых производителями промышленного оборудования, с другой – прямой интерфейс с пользователем через встроенный web-сервер или другими приложениями через API.

С помощью WEBHMI можно решать как простые задачи «пульта управления» оборудованием, другими словами - человеко-машинного интерфейса (HMI – Human Machine Interface), так и создавать системы мониторинга (диспетчеризации) распределенных объектов. Устройства могут обмениваться данными как между собой, так и с центральным диспетчерским сервером, передавая данные о состоянии контролируемых систем на верхний уровень. WEBHMI имеет достаточно хорошие средства обеспечения собственной надежности, но ориентирован, прежде всего, на управление небольшими (несколько десятков сигналов) обособленными или территориально распределенными системами, а не на решение критически важных задач управления на ответственных промышленных объектах.

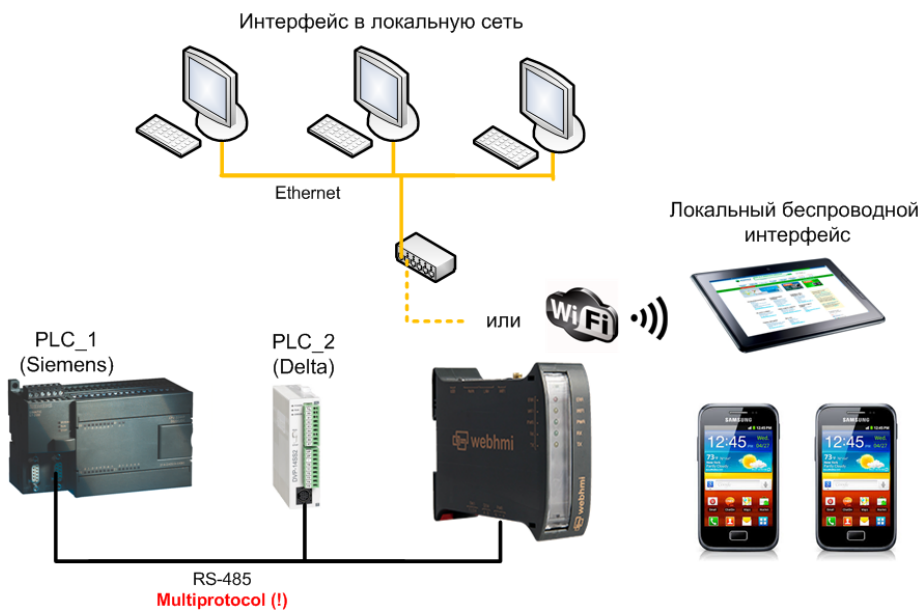
Концепция системы максимально проста и прозрачна. Устройство поставляется “из коробки”, без каких либо искусственных ограничений и дополнительных лицензий, в единственной конфигурации, обеспечивающей всю полноту его функциональных возможностей. WEBHMI является готовым решением, сочетая в себе ультракомпактную аппаратную платформу, отвечающую требованиям промышленных стандартов, с предустановленным ПО, обеспечивающим **быстрое развертывание проектов WEB-мониторинга различных объектов, без специальных знаний в области сетевых технологий и навыков программирования.**

Процесс создания проекта WEBHMI очень прост и интуитивно понятен, он в целом аналогичен принятому в большинстве современных SCADA систем и ПО операторских панелей. Однако, его уникальной особенностью является то, что средства разработки интегрированы прямо в пользовательский интерфейс. Таким образом, разработчику не требуется специфическое программное обеспечение, он может работать в любой удобной для него ОС, проект не требует компиляции и загрузки - все действия в процессе его конфигурирования сразу отражаются в системе, а благодаря веб-интерфейсу работать с проектом можно удаленно.

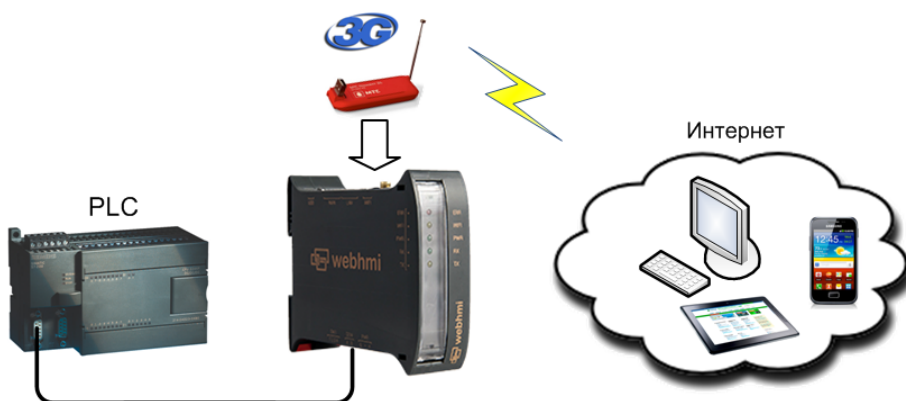
### Применение

Наиболее простое применение - локальный интерфейс оператора для мониторинга систем промышленной автоматизации и оборудования. Возможность управления различными устройствами с любого компьютера, планшета или смартфона, через веб-браузер, из любой точки сети или по WiFi. При этом WEBHMI может интегрироваться в существующую сеть или выступать в качестве самостоятельной точки доступа.

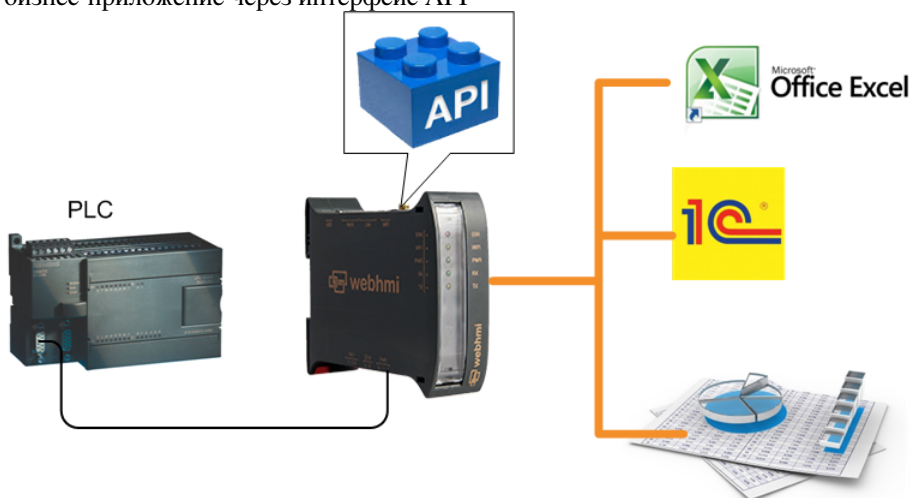
---



Вариант аналогичный первому, обеспечивающий доступ к интерфейсу через интернет, для мониторинга удаленных объектов. В случае отсутствия каналов связи наиболее рациональным решением будет подключить WEBHMI к мобильному 3G интернету, просто установив модем в USB разъем устройства. Для доступа к сайту потребуется получить у провайдера статический IP адрес или подключиться к какой-либо внешней сети с помощью VPN.



Следующий вариант предполагает возможность получения данных от техпроцесса непосредственно в целевое бизнес-приложение через интерфейс API



И наконец, вариант построения системы диспетчеризации распределенных объектов, предполагающей двухуровневую топологию, в которой WEBHMI выступают в качестве шлюзов данных нижнего уровня,

передавая информацию на выделенный сервер, обеспечивающий её дальнейшую обработку и хранение. Подробнее об этой возможности можно узнать у поставщика устройства.



# Подготовка к первому включению

## Описание внешних разъемов

**10-контактный разъем "под винт"** питание, реле, RS-485

RS-485	B	Data-
	A	Data+
	SG	Логический (сигнальный) "общий"
	NC	Нет подключения
Твердотельные реле	R2	Нормально открытый контакт реле №2
	COM	«Общий» реле №1 и 2
	R1	Нормально открытый контакт реле №1
"Земля"	GND	Изолированная «земля» для экранов разъемов LAN, WAN
Питание	+24V	Питание «+»
	-24V	Питание «-»

Подключение RS-485 - описано здесь. [1]

Нагрузочная способность твердотельных реле - макс. ток 100мА ~/=, макс. напряжение до 350В.

При использовании экранированного кабеля Ethernet его экран будет соединяться с GND, который для защиты от помех рекомендуется садить на "землю".

Напряжение питания 24В +/- 10% (15%).

Разъем подключения **внешней антенны WiFi** (тип RP-SMA)

**USB** - для подключения дополнительных устройств (спецификация 2.0).

Ethernet разъемы **LAN** и **WAN** - назначение см.здесь [2].

# Подключение внешних устройств

## Варианты подключения внешних устройств

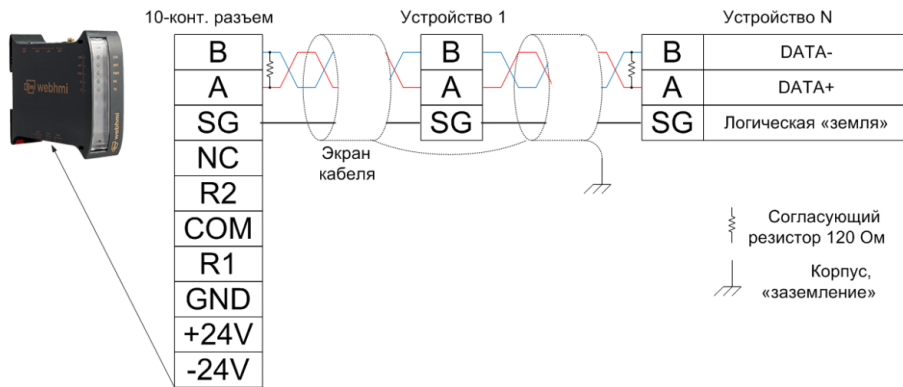
Устройства автоматики и другие интеллектуальные устройства, данные с которых необходимо получать можно подключать следующими способами:

- через встроенный порт RS-485, при этом на один и тот же порт **можно ставить устройства с разными протоколами**. это является отличительным преимуществом WEBHMI. Драйвер порта умеет динамически ("на лету") переконфигурировать приемопередатчик.
- устройства с другими интерфейсами, такими как RS-232,422 или дополнительные сети RS-485 можно подключать через переходники USB в COM порт, используя встроенный USB порт.
- устройства с протоколом Modbus/TCP можно подключать через порт Ethernet LAN или WAN.

Конфигурирование данных соединений описано здесь <sup>[1]</sup>.

## Рекомендации по подключению устройств с интерфейсом RS-485

Схема подключения WEBHMI к устройствам с интерфейсом RS-485.



Вероятно (если у Вас уже имеется опыт работы с данным интерфейсом), прочитав фразу RS-485 Вы сразу же решите использовать 2-х проводную схему, подключившись к стандартным выводам А и В, поскольку для передачи сигнала в нем действительно достаточно только 2 провода (этого же мнения придерживается довольно много специалистов). Однако это является **общепринятым заблуждением**. Стандарт EIA/RS-485 говорит о том, что **3-й провод нужен**:

*"Proper operation of the generator and receiver circuits requires the presence of a signal return path between the circuit grounds of the equipment at each end of the interconnection. The circuit reference may be established by a third conductor connecting the common leads of devices, or it may be established by connections in each using equipment to an earth reference."*

Источниками формирования подобного заблуждения могут быть:

- документация в интернете по теме:

### Подключение [\[ править | править вики-текст \]](#)

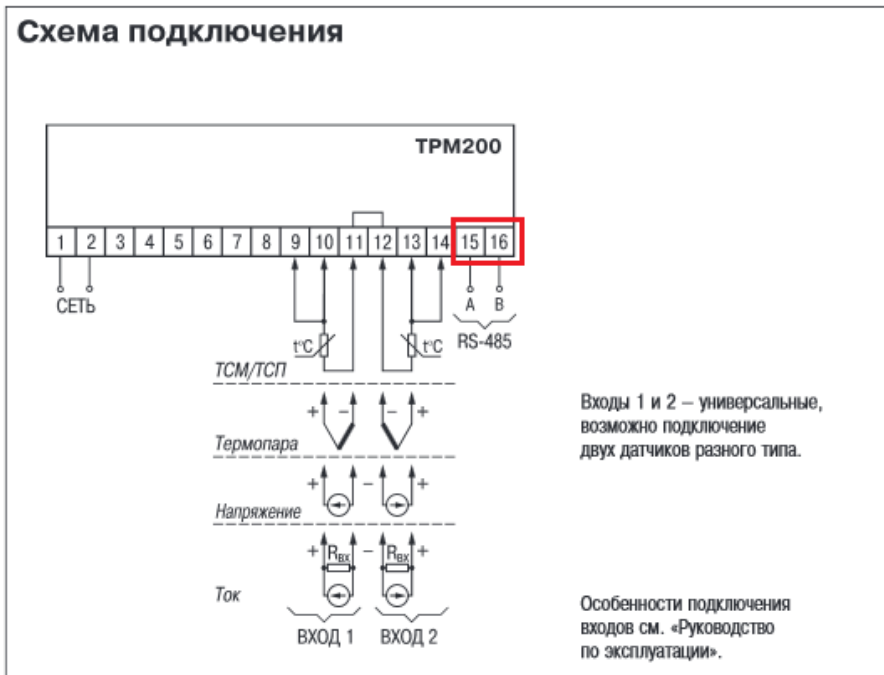
#### Контакты RS-485

Разъем состоит из двух или трех контактов:

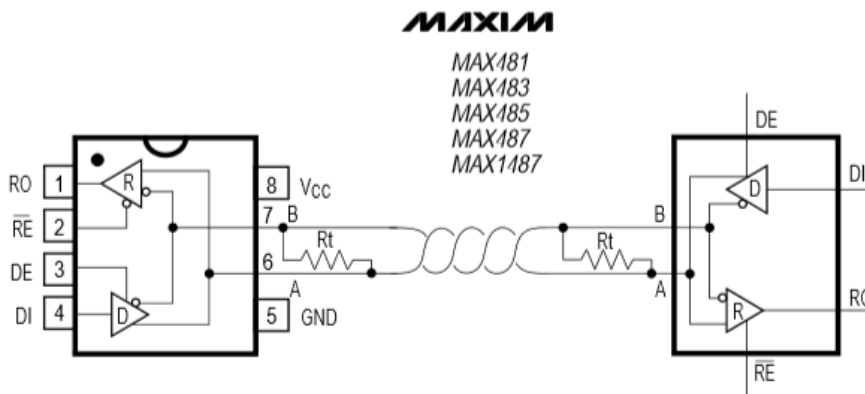
- А или «+» (TxD+/RxD+), неинvertированный<sup>[11]</sup>.
- В или «-» (TxD-/RxD-), инvertированный.
- Опциональный общий провод. Соединение общих шин устройств не обязательно, но улучшает устойчивость работы интерфейса **При наличии гальванической развязки не нужен.**



- некоторые устройства автоматизации, имеющие на борту всего две клеммы для подключения RS-485.



- широко распространенные кабели для RS-485, имеющие **всего два провода**.
- данные руководств производителей микросхем интерфейсов,

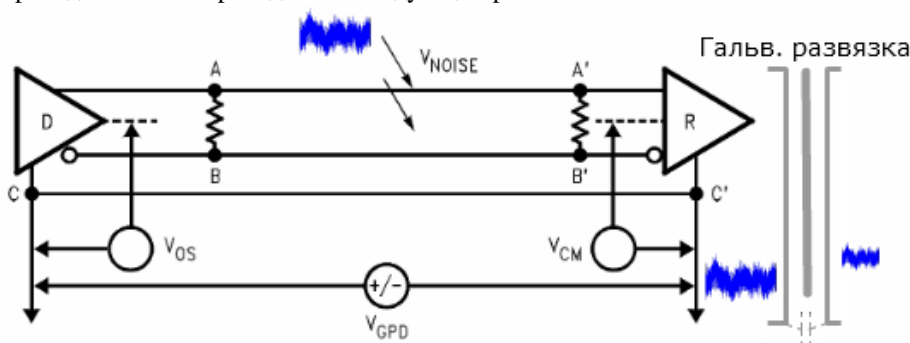


- положительный опыт использования 2-х проводной схемы
- и т.д.

В действительности оказывается, что:

- документация неофициальная
- устройство с 2 клеммами предполагает подключение программатора коротким проводом на низкой скорости обмена
- производитель кабеля не напоминает покупателю, что меры по обеспечению "выравнивания" общих точек приемопередатчиков никто не отменял и они прописаны в руководствах по эксплуатации на устройства (использованием отдельного провода или сажая "общие" на "чистую" землю)
- производитель микросхемы для упрощения восприятия совмещает принципиальную и функциональную схемы
- 2-х проводная схема работала на столе, на коротком проводе, на длинном проводе, но была благоприятная электромагнитная обстановка

Таким образом, для надежного соединения необходимо использовать 3-х проводную схему. Работа трех-проводной схемы приведена на следующем рис.



Здесь

$V_{os}$  - напряжение смещения передатчика (измеряется относительно общего и средней точки делителя напряжения с одинаковыми плечами, подключенного к выходам A,B),  $V_{gpd}$  - напряжение между "общими точками" передатчика и приемника,  $V_{noise}$  - напряжение электромагнитной "наводки" на линию,  $V_{cm}$  - напряжение помехи "общего вида" прикладываемое к обоим входам приемника, которое складывается в итоге как  $V_{gpd} + V_{os} + V_{noise}$ . Передатчик модулирует дифференциальное напряжение, формируя сигнал. Приемник должен распознать этот сигнал в условиях сильных помех, и делает это очень эффективно, но в определенных пределах. Величины напряжений для передатчика и приемника ограничены стандартом и составляют от -7 до +12В (+/- 7В относительно сигнала 0..5В). В реальности кроме помех, между общими точками приемника и передатчика может существовать разность потенциалов "земель"  $V_{gpd}$ , которая прикладывается через выходы формирователя ко входам приемника. В случае превышения  $V_{cm}$  указанных выше уровней, связь становится ненадежной либо вообще пропадает. Наводимую помеху эффективно подавляет экран, заземленный в одной точке. Помеху через разность потенциалов "общих" точек подавляет третий провод, замыкая на себя токи, порождаемые  $V_{gpd}$ , которые в противном случае стали бы входными токами линий A и B приемника.

Что касается 2-х проводной схемы **изолированного RS-485**, она также является нестабильной по следующей причине. Система может подвергаться наводкам и в этом случае, но токи помехи "общего вида" всегда стремятся вернуться к источнику, независимо от того в каком месте и как цепь захватила помеху, т.е. в том числе и вернуться через изолированный "общий" схемы. Эта точка может стать в свою очередь источником электромагнитных помех, например для другой части электронной схемы (например, через емкостные паразитные связи). Столь большое объяснение роли одного провода приводится только лишь с целью уберечь пользователя от потери драгоценного времени на наладку связи. Кроме этого, на длинных линиях и при высоких скоростях обмена рекомендуется использование согласованного кабеля (витая пара в экране, волновое сопротивление 120 Ом) и согласующих резисторов на концах сегмента сети.

## Диагностика подключения

В первую очередь после подключения устройства и создания соединения можно создать регистр находящийся в заведомо известном состоянии, например бит "всегда включен" и т.п. После создания регистра его значение (Value) будет отображаться с списке регистров:

ID	Title	PLC	Address	Type	State	Value	Actions
1	Идет упаковка	S7 200	M3.4	Bit	Enabled	0	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Clone</a>

Перечеркнутое соединение означает, что связи нет. В случае отсутствия связи в первую очередь еще раз стоит проверить кабель и соответствие настроек с обеих сторон.

Помочь разобраться с вопросами связи на встроенном порту RS-485 помогут диагностические светодиоды TX/RX на передней панели - при нормальной работе они должны мигать поочередно. При отсутствии RX - понятно что устройство не отвечает вообще, т.е. возможно неправильно настроен формат или скорость обмена. Если светодиод RX мигает аритмично, как бы "пробиваясь" через равномерное мигание TX скорее всего дело в наличии сильных электромагнитных помех, необходимо выполнить рекомендации под

подключению RS-485.

Если значение читается, но неправильно - задан не тот тип данных в описании регистра, поэтому например может читаться только байт параметра вместо целого слова.

## Первое включение

---

### Подготовка к работе

Перед первым включением WebHMI необходимо выполнить некоторые подготовительные шаги.

1. Вставить SD-карту. Рекомендуемый объем - от 512 MiB. Более надежны карты с памятью SLC типа.
2. По умолчанию у WebHMI IP-адрес интерфейса LAN - 192.168.1.1. Если планируется подключение к локальной сети то необходимо убедиться что в этой сети нет устройств с адресом 192.168.1.1 иначе произойдет конфликт адресов. Это же относится к ситуации когда в локальную сеть подключаются несколько устройств WebHMI с заводскими настройками. У всех устройств будет одинаковый IP адрес и сеть работать будет не корректно.
3. При подключении шины RS485 следует убедиться что подключены не только линии А и В но и их общий провод который выравнивает потенциалы между WebHMI и другими устройствами на шине. Это особенно актуально если линия длинная или устройства питаются от разных источников питания.
4. Если на вашем компьютере нет WiFi то необходимо подключить WebHMI к вашему компьютеру или сети с помощью Ethernet-кабеля через интерфейс LAN. Только на этом интерфейсе разрешены входящие соединения. Учтите, что у WebHMI на этом интерфейсе по умолчанию включен DHCP-сервер. Возможны конфликты с другими DHCP-серверами в локальной сети.

После выполнения этих шагов можно включать WebHMI.

В процессе загрузки система проверит наличие SD-карты и автоматически отформатирует ее если это необходимо.

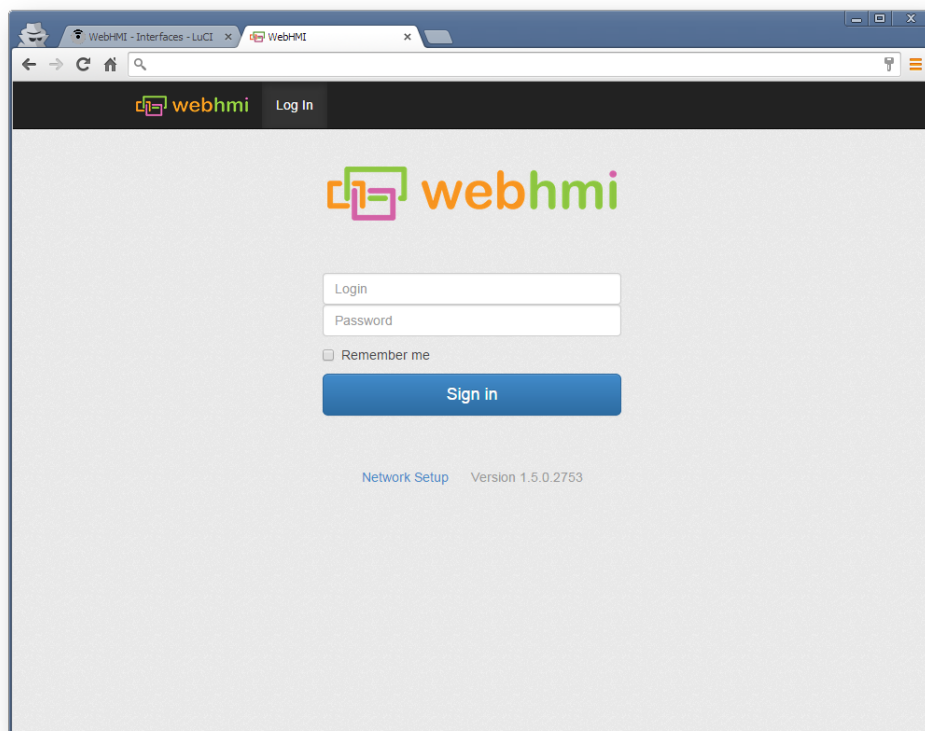
Если SD-карта не будет обнаружена то WebHMI начнет мигать тремя светодиодами Error, WWW, VPN. После 20 миганий система загрузится но не все службы будут запущены. В течении следующей минуты watchdog увидит что SD карты нет и перезагрузит систему. Таким образом, система будет постоянно перезагружаться пока не будет вставлена SD-карта.

### Доступ к веб-интерфейсу

Работа с WebHmi происходит по сети. Все настройки, включая конфигурирование проектов, делаются в браузере. Все что нужно сделать для начала работы, это подключиться к устройству и зайти к нему в веб-интерфейс.

#### Беспроводное подключение по Wi-Fi

По умолчанию беспроводной интерфейс WebHmi сконфигурирован в режиме точки доступа с включенным DHCP и отключенной безопасностью. Таким образом, Ваш ноутбук или планшет можно просто подключить к сети **WebHmi** (настройка параметров произойдет автоматически), после чего зайти на сайт с адресом: **192.168.1.1**

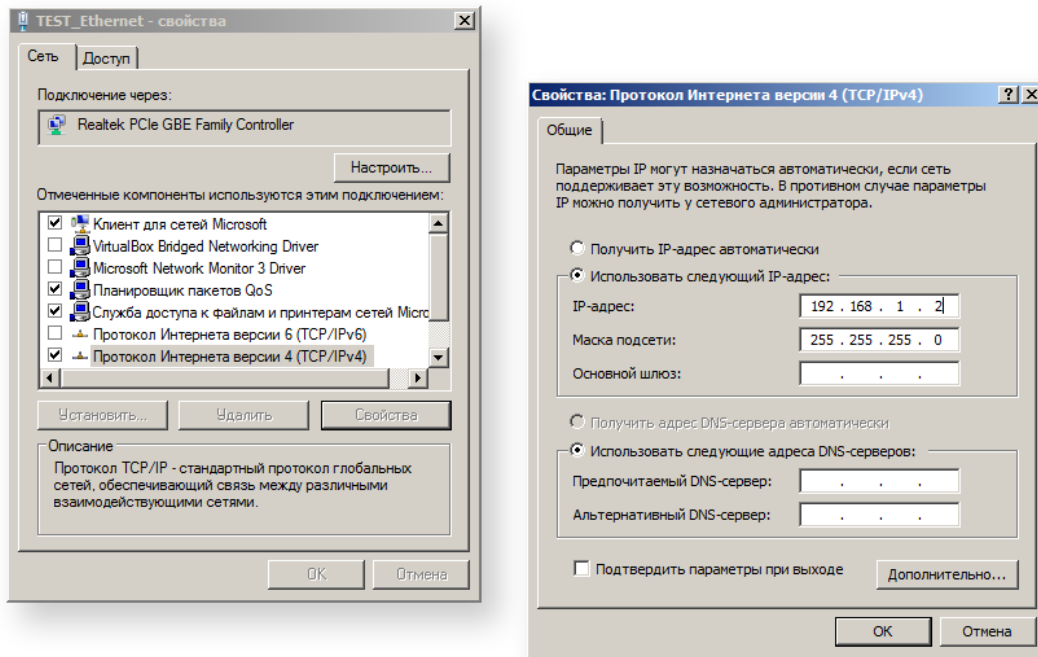


### Проводное подключение

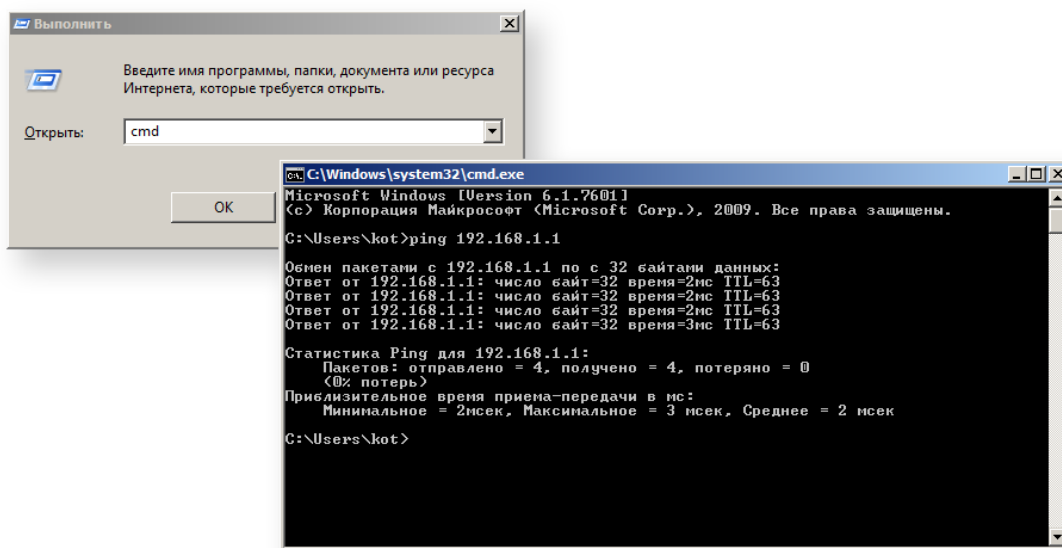
WebHmi имеет 2 порта Ethernet: LAN сконфигурированный как DHCP сервер и WAN настроенный как DHCP клиент, для которого по умолчанию в фаерволе запрещены входящие соединения. Таким образом, доступ к веб-интерфейсу системы можно получить только из LAN.

Дальнейшие действия могут отличаться в зависимости от ОС (пример с Windows 7)

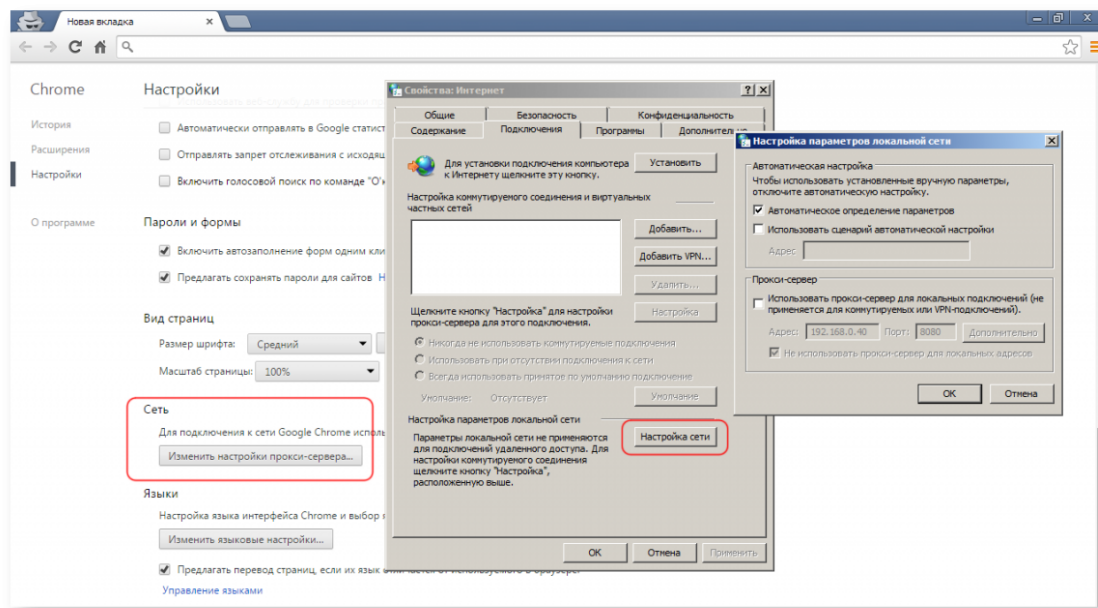
1. Подключить кабель Ethernet к порту LAN
2. Настроить в ПК свойства сетевого подключения - IP адрес [192.168.1.2] и маску подсети [255.255.255.0] или выбрать «Получить IP адрес автоматически» (Пуск / Панель управления / Центр управления сетями и общим доступом / Изменение параметров адаптеров / Свойства подключения / Протокол интернета версия 4 / Свойства). Ok.



3. Проверить связь. Вызвать окно «Выполнить» нажав сочетание клавиш Win+R и запустить от туда интерпретатор командной строки “CMD”. В открывшемся окне “CMD” выполнить команду **ping 192.168.1.1** Если пойдет обмен пакетами - то связь есть.



Если, связь есть, но по каким-то причинам не удастся зайти на сайт – проверьте, не используется ли **proxy server** в настройках браузера.



**СОВЕТ.** Для корректной работы веб-интерфейса системы, используйте браузер CHROME [1] последней версии.

---

# Настройка сетевых подключений

---

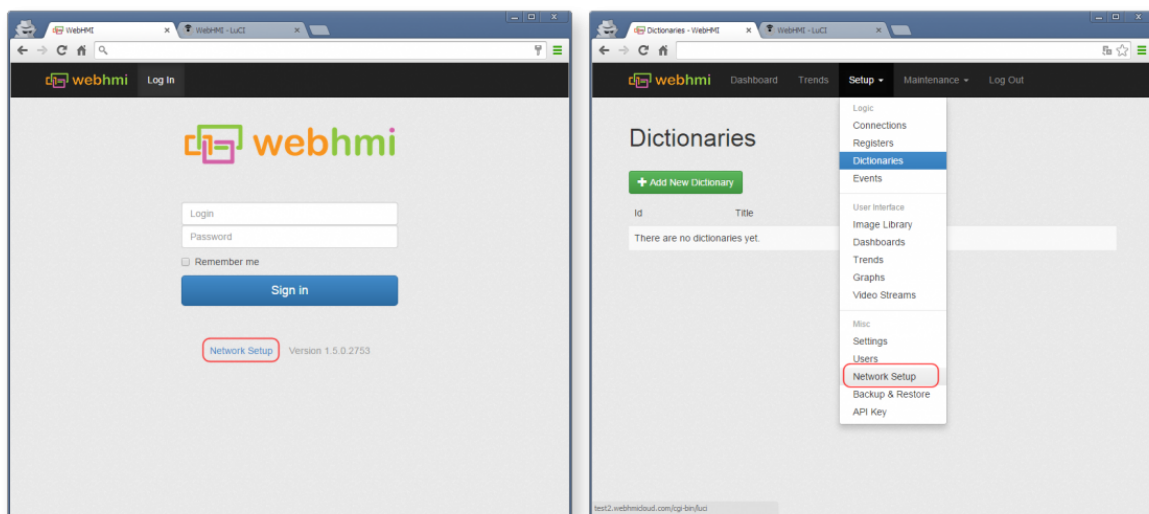
## Настройка сетевых соединений

---

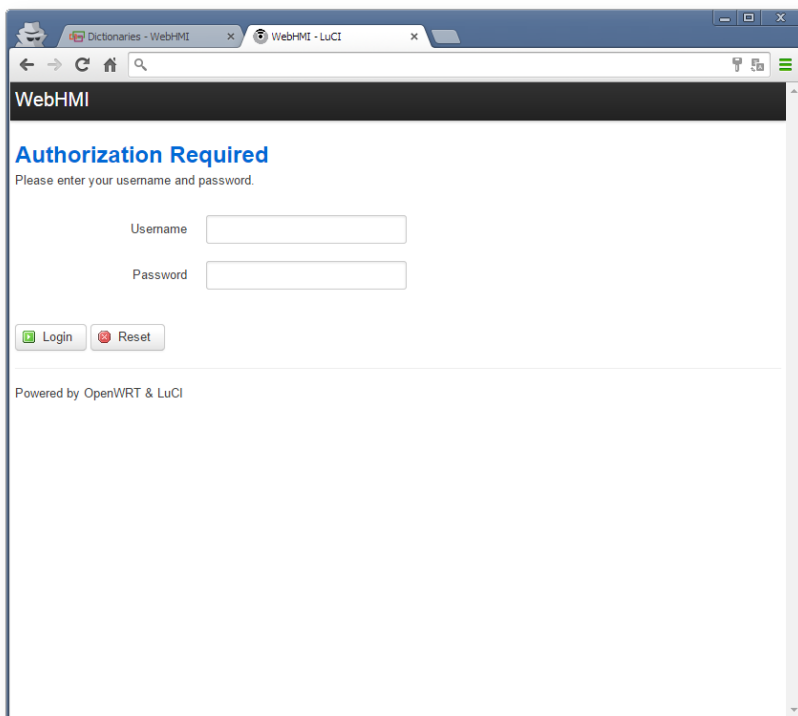
### Интерфейсы lan и wan

Независимо от того, какие конкретно задачи решаются с помощью WEBHMI, в любом случае, потребуется его подключение к локальной сети или сети Internet. Есть несколько вариантов как это можно сделать, самый надежный из них – проводной Ethernet.

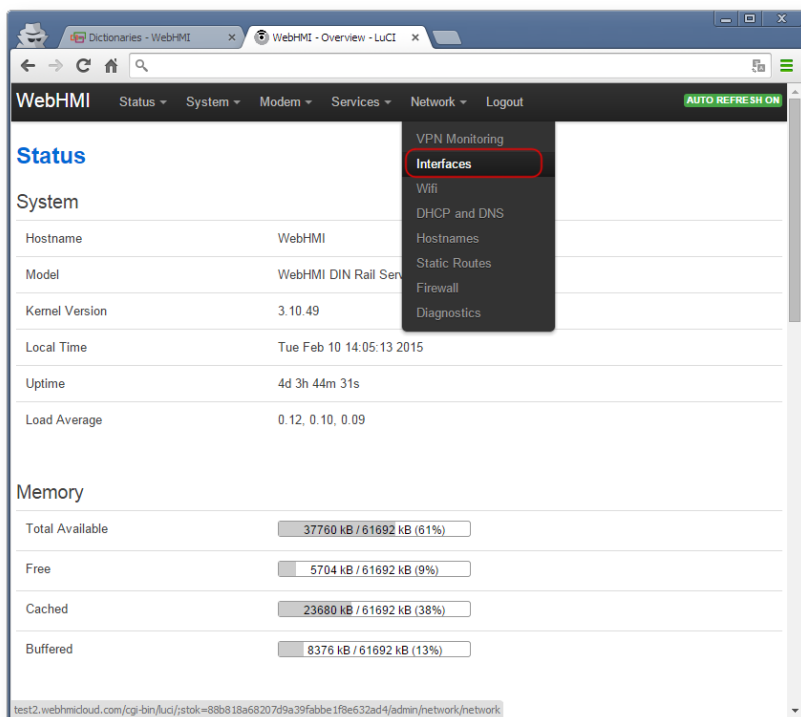
WEBHMI имеет 2 сетевых разъема RJ45 маркированных как LAN и WAN. По умолчанию LAN сконфигурирован как DHCP <sup>[1]</sup>-сервер, обеспечивающий автоматическую раздачу сетевых настроек подключаемым устройствам, а WAN как DHCP-клиент, ориентированный на подключение к сети, в которой уже есть какой-то DHCP-сервер, к примеру, роутер раздающий интернет. Кроме того, настройки Firewall <sup>[2]</sup> запрещают все входящие подключения из сети WAN, таким образом, доступ к Web-интерфейсу WEBHMI <sup>[3]</sup> возможен только со стороны LAN. Изменить текущие настройки можно с помощью панели **Network setup**, доступ к которой можно получить по ссылке на странице авторизации или из главного меню **Setup / Network setup** после входа в систему



Для входа в панель настроек потребуется авторизация. По умолчанию **Login: admin / Password: webhmi**  
**ВНИМАНИЕ! В целях безопасности, в процессе настройки проекта, пожалуйста, измените стандартный пароль**

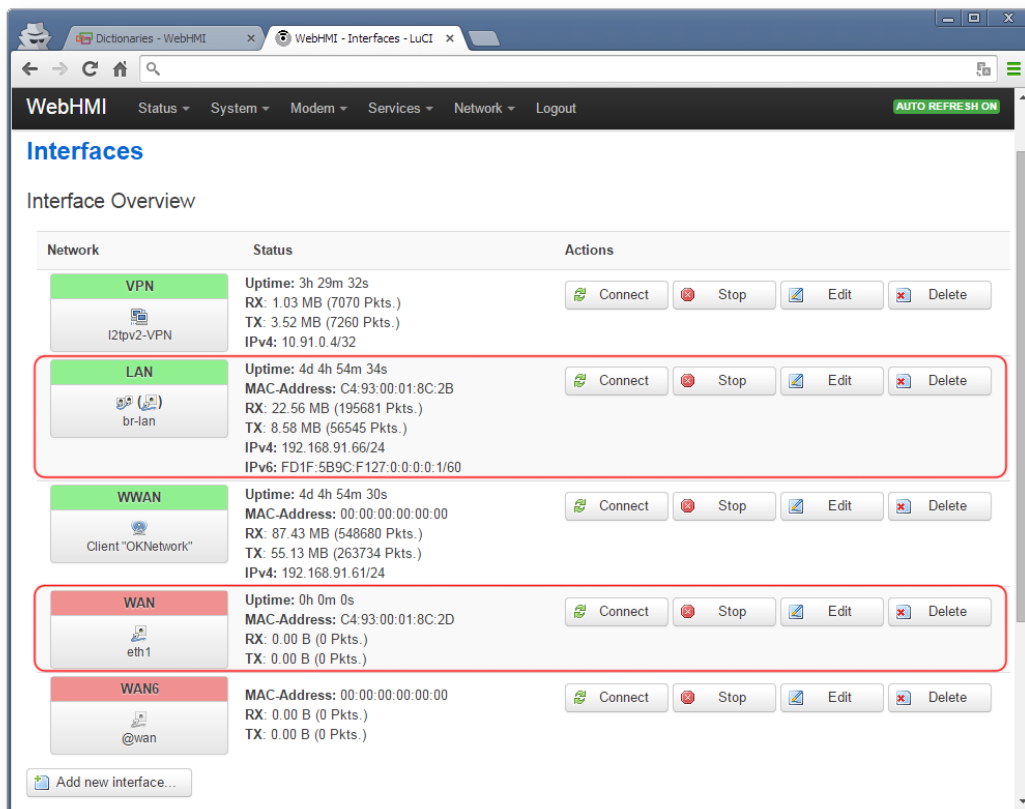


Все сетевые настройки сгруппированы во вкладке **Network**. Подменю **Interfaces** позволяет создавать, конфигурировать и управлять сетевыми подключениями.



На этой странице отображается общая информация, характеризующая текущее состояние сетевых подключений. Для изменения параметров интерфейса (LAN или WAN) нажмите **Edit**.





Все настройки для удобства разбиты на несколько отдельных групп по закладкам: **General**, **Advanced**, **Physical** и **Firewall settings**:

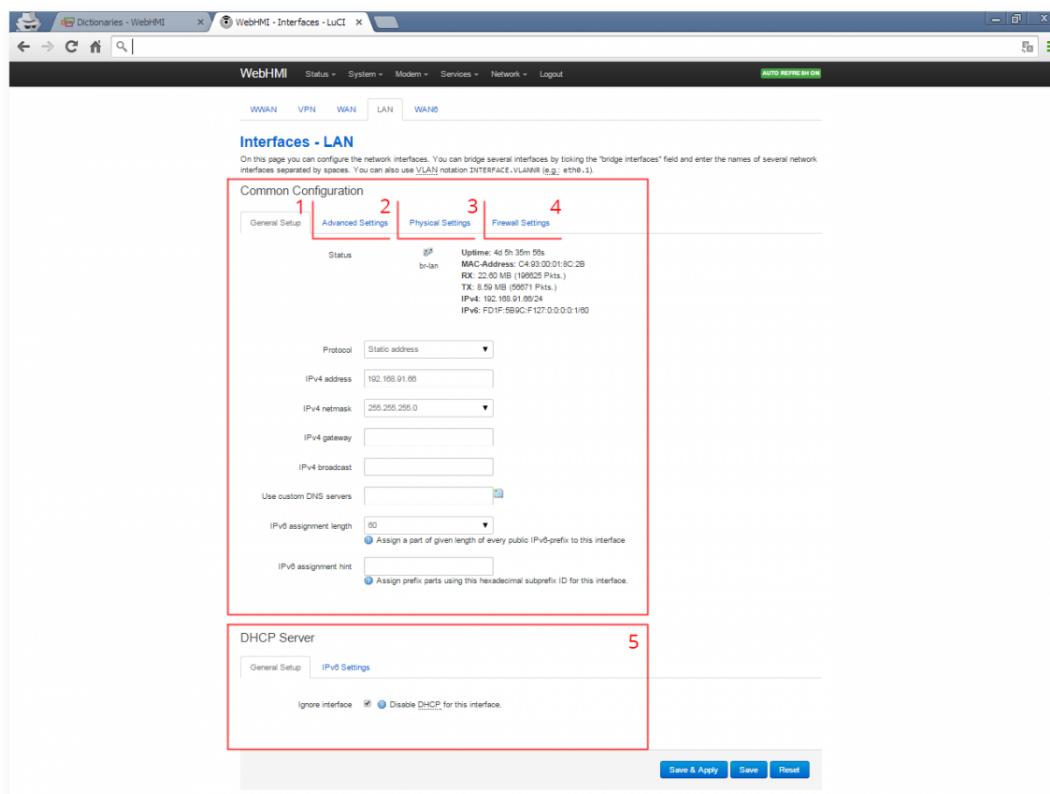
**General setup** позволяет задавать общие (основные) настройки, которые будут отличаться, в зависимости от выбранного типа протокола (способа подключения: Static IP, DHCP, PPPoE, L2TP и пр.). В случае статической адресации (наиболее широко используемый вариант) это будут: IP адрес <sup>[4]</sup> хоста, маска подсети, адреса шлюза и DNS <sup>[5]</sup> серверов.

В **Advanced settings** задаются такие параметры как: MAC <sup>[6]</sup> адрес порта (идентификатор устройства, используемый протоколами канального уровня, а также системами безопасности для управления доступом к сетевым ресурсам и фильтрации пакетов), MTU <sup>[7]</sup> размер кадра TCP и метрика шлюза.

**Physical settings** определяет привязки к коммуникационным интерфейсам устройства. В этой вкладке можно объединить несколько интерфейсов мостом, связав, тем самым, несколько сегментов сети, а также, используя STP, организовать отказоустойчивое резервирование каналов связи (STP <sup>[8]</sup> - Spanning Tree Protocol позволяет создавать избыточные связи между сегментами сети, избегав топологических петель и закливания пакетов).

Вкладка **Firewall settings** позволяет указать групповые политики, в соответствии с которыми будет проводиться обработка входящего и исходящего трафика для этого соединения.

**В самом общем случае, для подключения WEBNMI к локальной сети достаточно будет настроить лишь несколько важных параметров.**



Вкладка **General settings**:

1. **Protocol** – указывает на тип подключения. Несмотря на все видимое многообразие, для локальной сети возможно лишь два варианта: DHCP client и Static address. DHCP-client подразумевает получение настроек сети автоматически (возможно лишь в том случае, когда в сети есть DHCP-сервер), в случае Static address настройки вводятся вручную пользователем.

**СОВЕТ.** Как правило динамическая адресация не удобна для предоставления каких либо сервисов внутри сети т.к. каждый раз адреса ресурсов будут меняться и доступ к ним будет затруднен

2. **IP address** – уникальный адрес устройства (узла) в IP сети. В версии протокола IPv4 он имеет длину 4 байта и записывается через точку типа xx.xx.xx.xx IP-адрес устройства состоит из адреса сети и адреса узла в этой сети (определяется маской подсети). О правилах адресации в IP сетях можно прочитать здесь <sup>[9]</sup>.

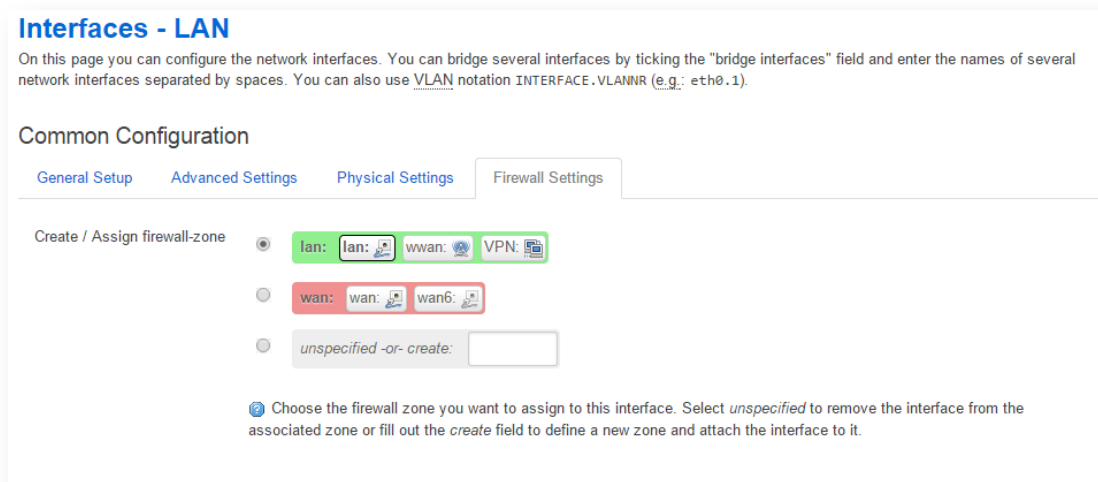
**ВНИМАНИЕ!** В случае подключения к одной сети нескольких устройств с одинаковым IP-адресом это вызовет конфликт адресов, что приведёт к отключению одного из устройств. Такое возможно, например, при одновременном подключении к сети двух устройств WebNmi с заводской конфигурацией

3. **IP netmask** (маска подсети) – указывает на то, какая часть IP-адреса (старшие биты) относится к адресу сети, а какая (младшие) к адресу хоста (узла) в этой сети. В IPv4 маска подсети имеет длину 4 байта и записывается побайтно, через точку, по аналогии с IP-адресом.

4. **IP gateway** (шлюз) – это адрес устройства в сети которое обрабатывает все пакеты направленные за пределы этой сети (устройствам с адресом из другой сети).

Вкладка **Firewall settings**:

5. Необходимо указать к какой зоне относится этот интерфейс. В данном случае LAN или WAN. Подробнее см. раздел Межсетевой экран (Firewall) <sup>[2]</sup>



**ВНИМАНИЕ!** В случае запрета всех входящих соединений для всех сетей, будет полностью потерян доступ к WEB-интерфейсу устройства. В этом случае, для восстановления придется сбрасывать настройки <sup>[10]</sup> и все данные, включая проект будут потеряны

6. **DHCP Server** (Dynamic Host Configuration Protocol) – протокол автоматической настройки узлов IP сети. Если устройство подключается как DHCP клиент, или в сети используется статическая адресация – отключите эту опцию.

После изменения настроек нажмите **Save & Apply** внизу экрана. Новые параметры вступят в силу через несколько секунд.

**ВНИМАНИЕ!** При изменении настроек сети, в случае **ОТСУТСТВИЯ ДОСТУПА К УСТРОЙСТВУ**, не забудьте проверить настройки сетевого подключения Вашего ПК

## Wi-Fi

Возможности работы WebNmi в беспроводных сетях Wi-Fi позволяют очень гибко решать вопросы организации связи с устройствами, там, где по какой либо причине не получается использовать проводное подключение к локальной сети.

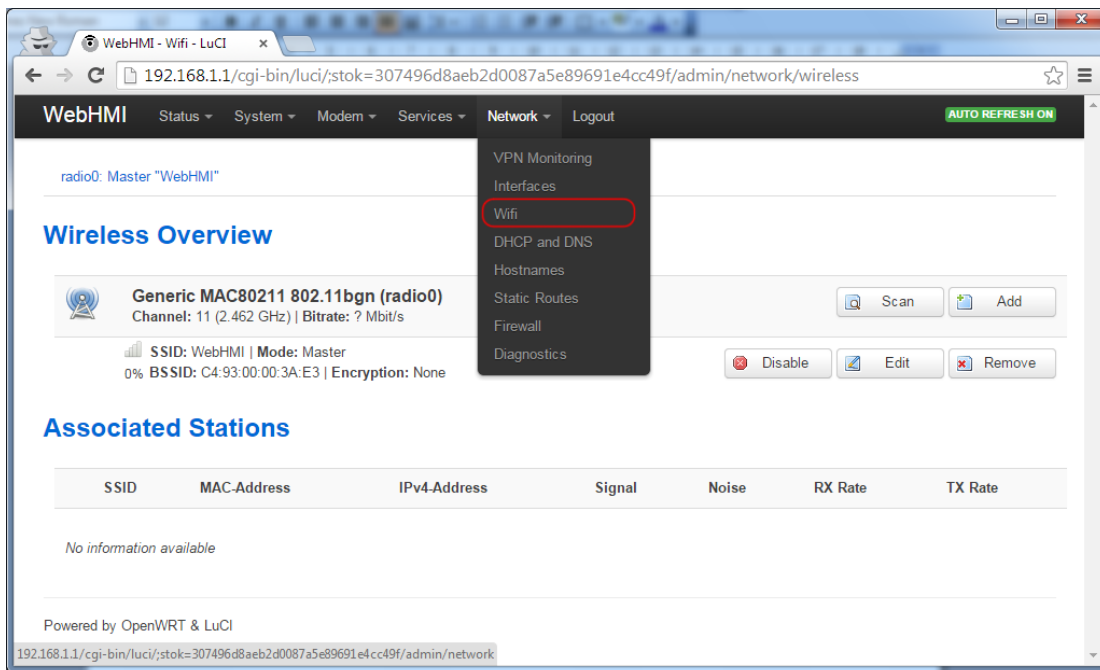
**СОВЕТ.** Если есть такая возможность, всегда следует отдавать предпочтение более надежному проводному подключению в сравнении с беспроводным.

WebNmi может быть как клиентом в уже существующей сети, так и работать в режиме точки доступа. **Интересной особенностью является возможность одновременной работы сразу в нескольких сетях!** (например, подключиться к интернету в одной сети и одновременно с этим, раздать доступ к своим ресурсам в другой).

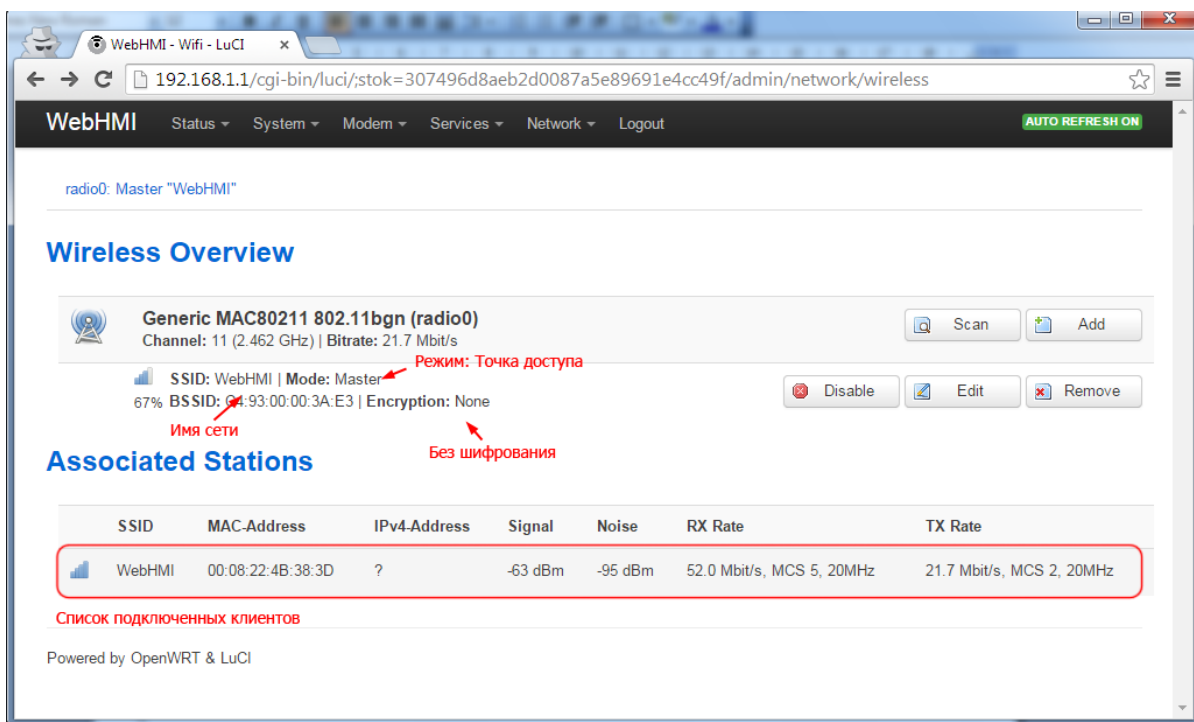
**СОВЕТ.** На передней панели устройства находится индикатор Wi-Fi, отображающий активность беспроводных подключений

### Точка доступа (Access point)

По умолчанию беспроводной интерфейс WebNmi сконфигурирован как точка доступа. Настройки параметров беспроводных сетей собраны в разделе **Network / Wi-Fi** главного меню панели **Network setup** <sup>[11]</sup>



В открывшемся окне **Wireless Overview** отображаются основные параметры существующих беспроводных сетей. Отсюда можно ими управлять: добавлять/удалять, включать/выключать, а также менять их настройки.



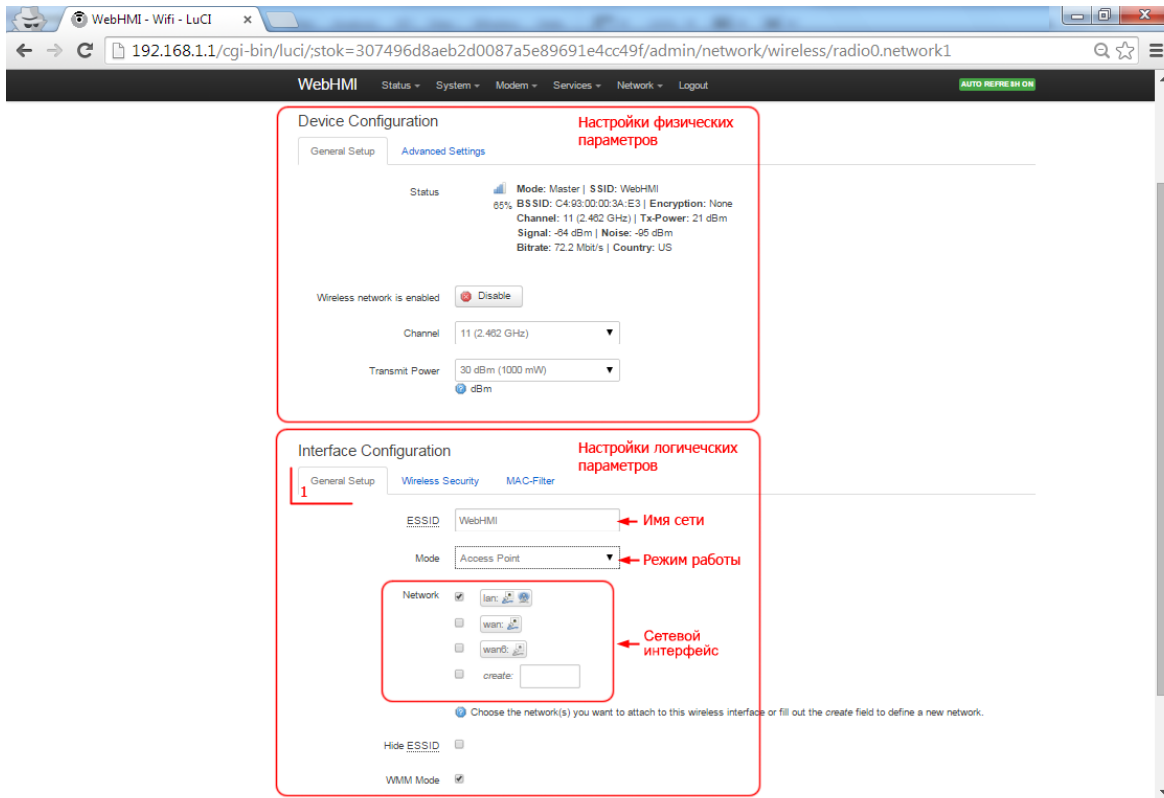
Для изменения настроек существующей сети нажмите **Edit**. Здесь все параметры по смыслу разделены на две группы: физические настройки, определяющие параметры радиоканала, которые в большинстве случаев можно не трогать и логические, задающие, собственно, свойства беспроводной сети – её идентификатор, режим работы и параметры безопасности.

**ВНИМАНИЕ!** Если вы подключены к WebHmi по Wi-Fi – манипуляции с настройками беспроводных сетей могут привести к нарушению связи с устройством.

**Interface configuration**, закладка **General setup**:

1. **ESSID** – имя сети
2. **Mode** – режим работы. В большинстве случаев он будет либо **Access point** (точка доступа) при создании новой беспроводной сети, либо **Client** – при подключении к уже существующей
3. **Network** – определяет физическую привязку этой сети либо к другим, уже существующим сетям (подключение типа мост), либо создание для нее нового интерфейса

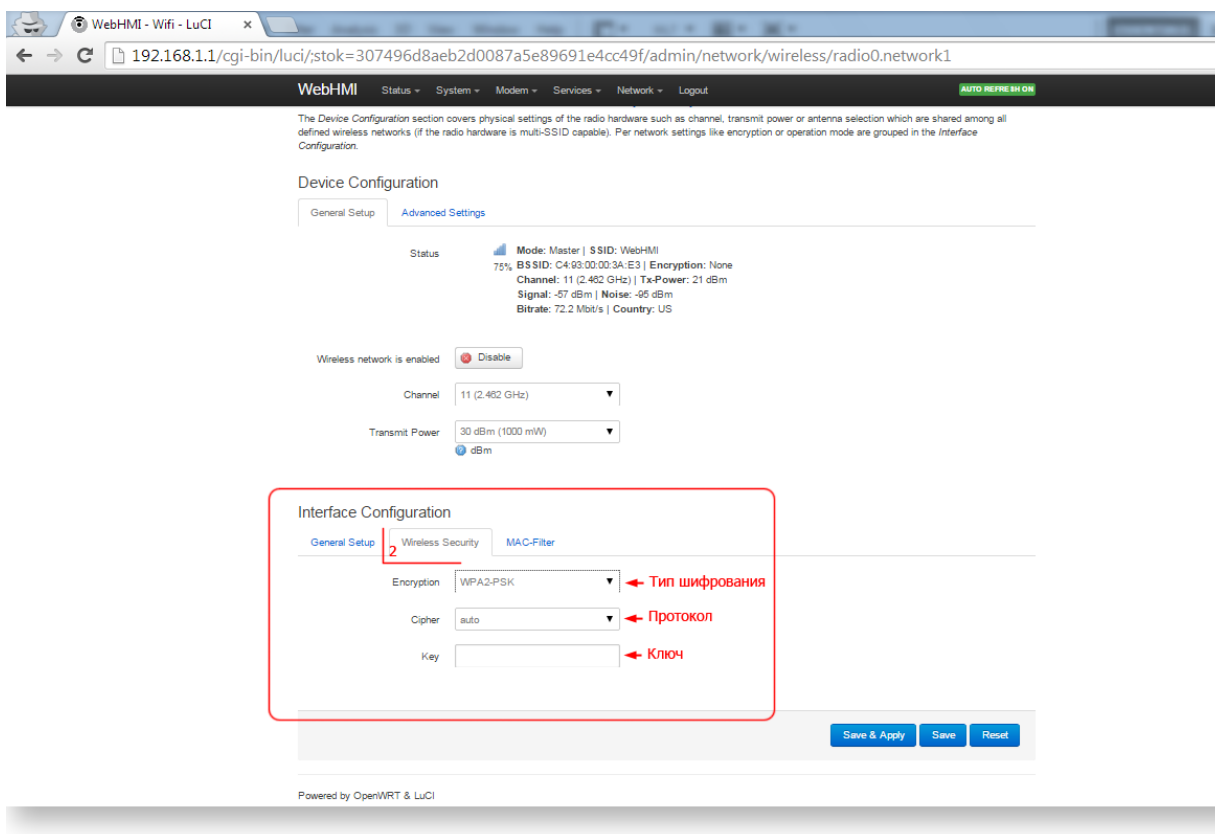
**СОВЕТ.** В случае если к этой беспроводной сети не предполагается подключение гостевых устройств ее можно скрыть, используя опцию **Hide ESSID**



Закладка **Wireless security** определяет настройки безопасности беспроводной сети.

**СОВЕТ.** Для ограничения доступа к беспроводным сетям используйте более совершенную технологию **WPA2-PSK** [12].

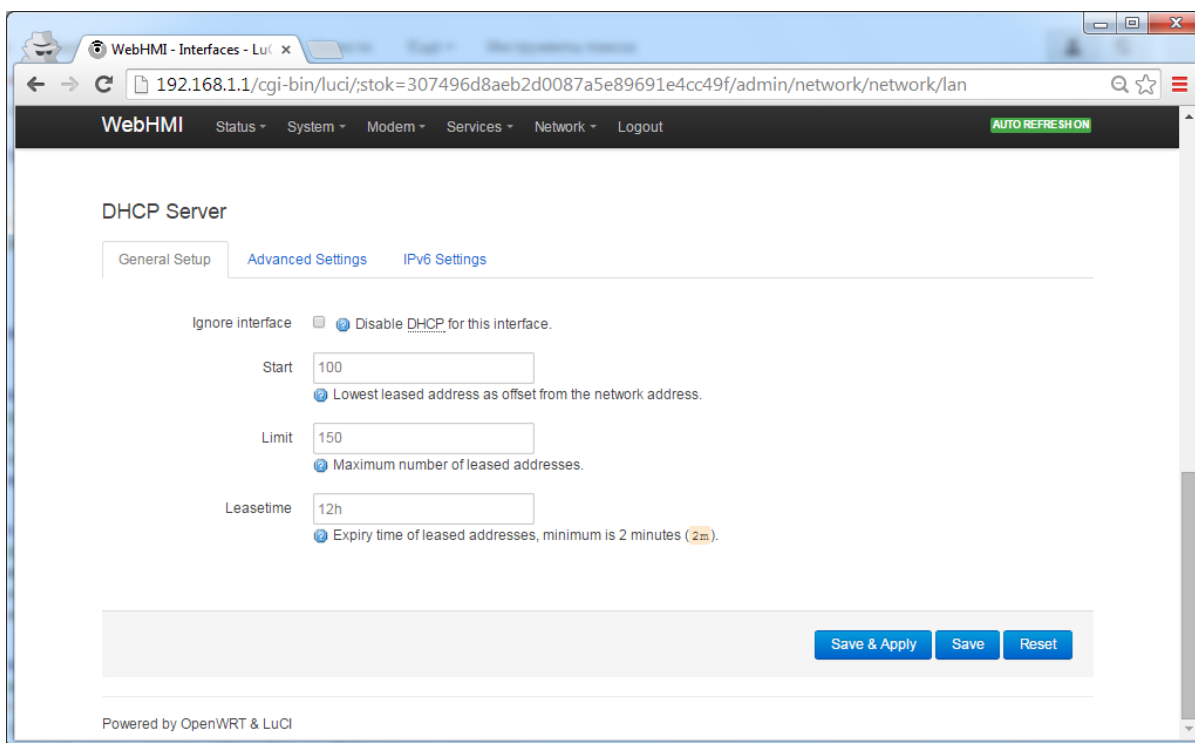
4. **Encryption** - выберите тип шифрования или оставьте доступ к сети открытым
5. **Key** – придумайте ключ безопасности (от 8-ми до 63-ех символов)
6. Сохраните изменения **Save & Apply**



Точно так же можно было бы создать и новую беспроводную сеть.

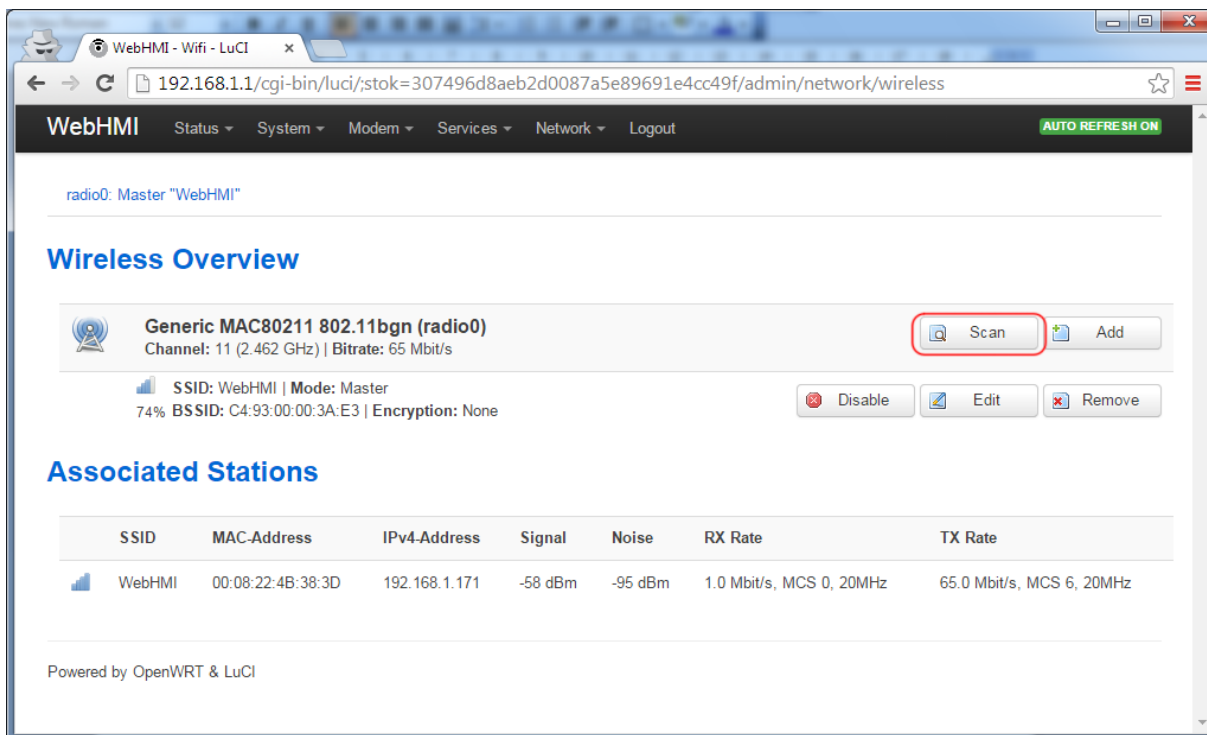
**Однако это еще не конец.** В результате произведенных манипуляций пока создан лишь **новый сетевой интерфейс** (как бы устройство, сетевой адаптер), который появится в списке сетей во вкладке **Interfaces**, меню **Network**. Теперь его нужно будет сконфигурировать для работы в IP сети (назначить адрес, шлюз и пр.) аналогично описанным ранее настройкам интерфейсов LAN и WAN [13]

**СОВЕТ.** Как правило, режим работы «Точка доступа» предполагает наличие в сети DHCP сервера, раздающего сетевые настройки подключаемым клиентам. Если это нужно – включите эту опцию при конфигурировании интерфейса.



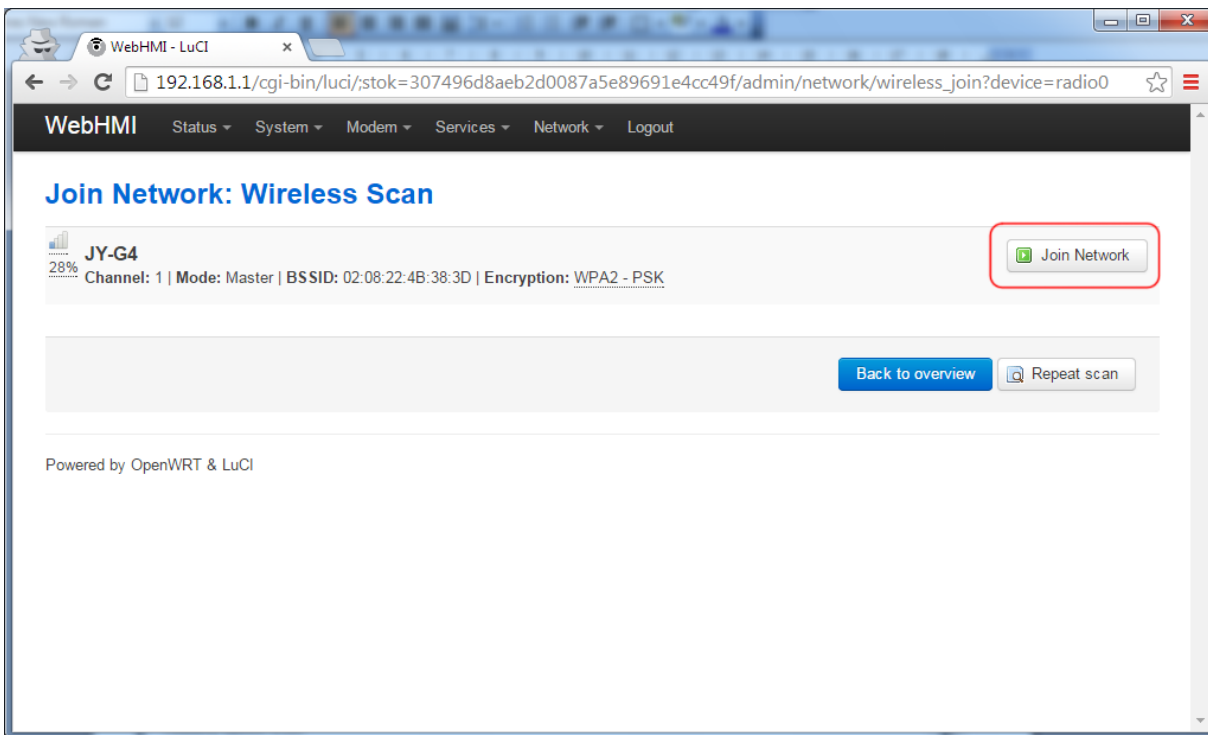
### Подключение к существующей сети (Client mode)

1. Нажмите **Scan** в окне **Wireless Overview**, меню **Network / Wi-Fi**



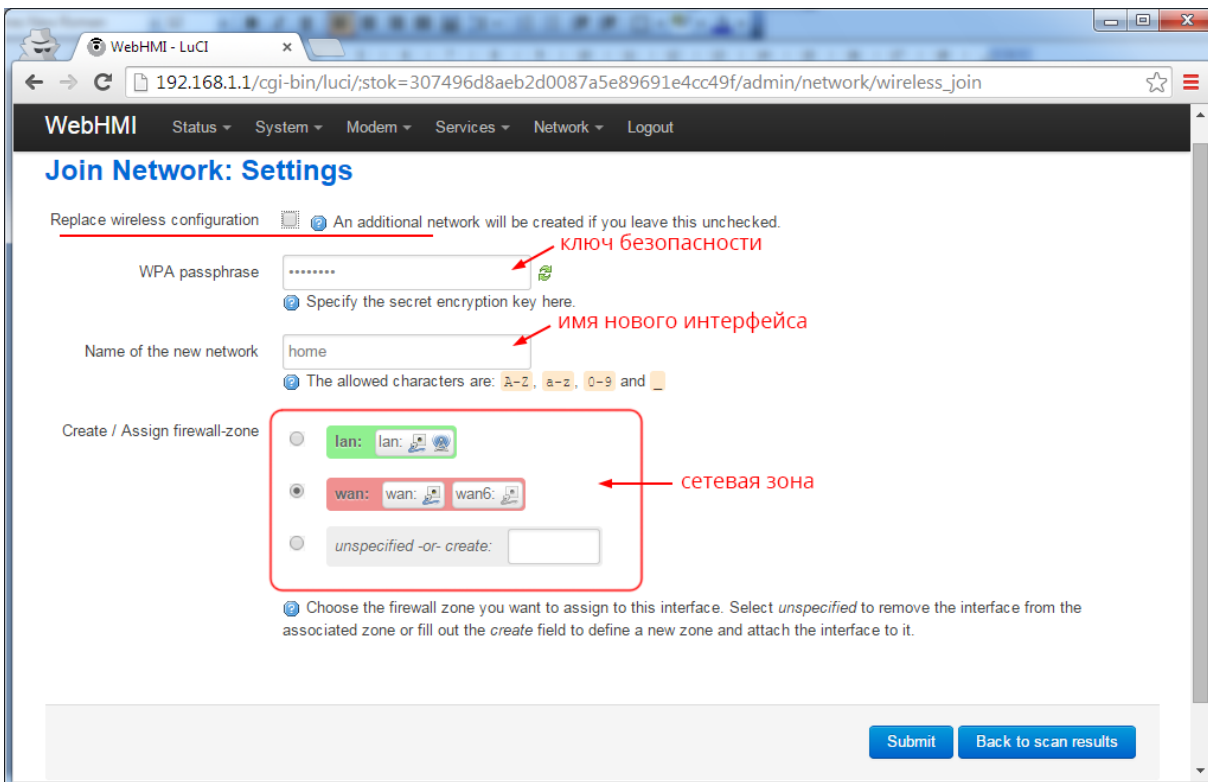
WebHmi покажет список найденных сетей.

2. Нажмите **Join Network** для той сети, к которой собираетесь подключиться



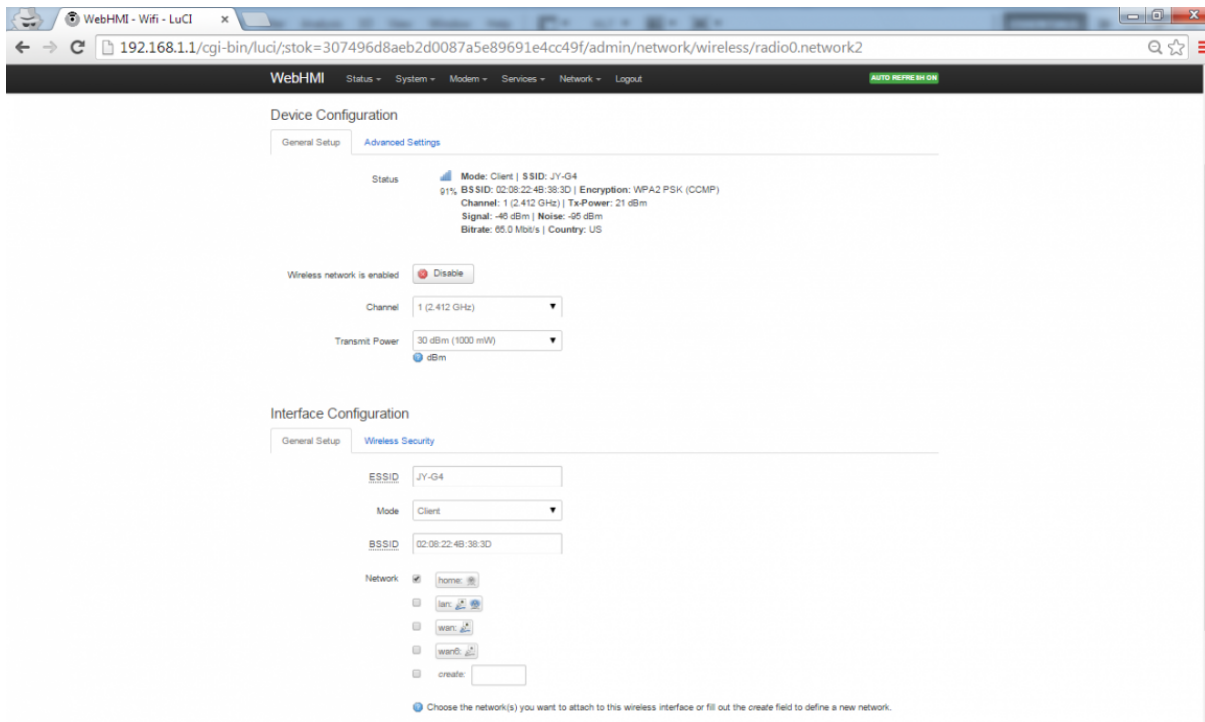
В появившемся окне:

3. Введите ключ безопасности сети, к которой Вы подключаетесь
4. Задайте название сети (то, как она будет отображаться в системе). Обратите внимание на опцию **Replace wireless configuration**, вверху экрана. Если она выбрана, то новая беспроводная сеть заменит существующую, если нет - то будет создана новая сеть.
5. Исходя из уровня доверия к сети, укажите нужную сетевую зону, определяющую правила обработки трафика для данного подключения. Подробнее см. настройка Firewall [2]

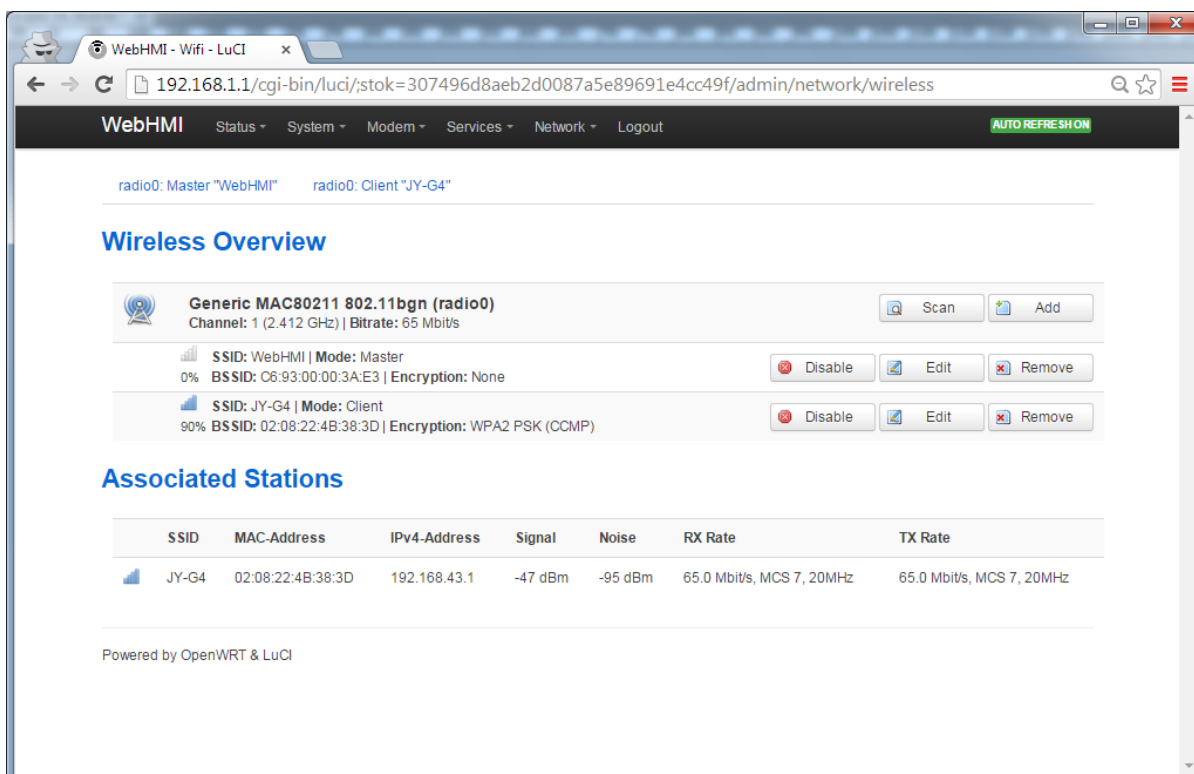




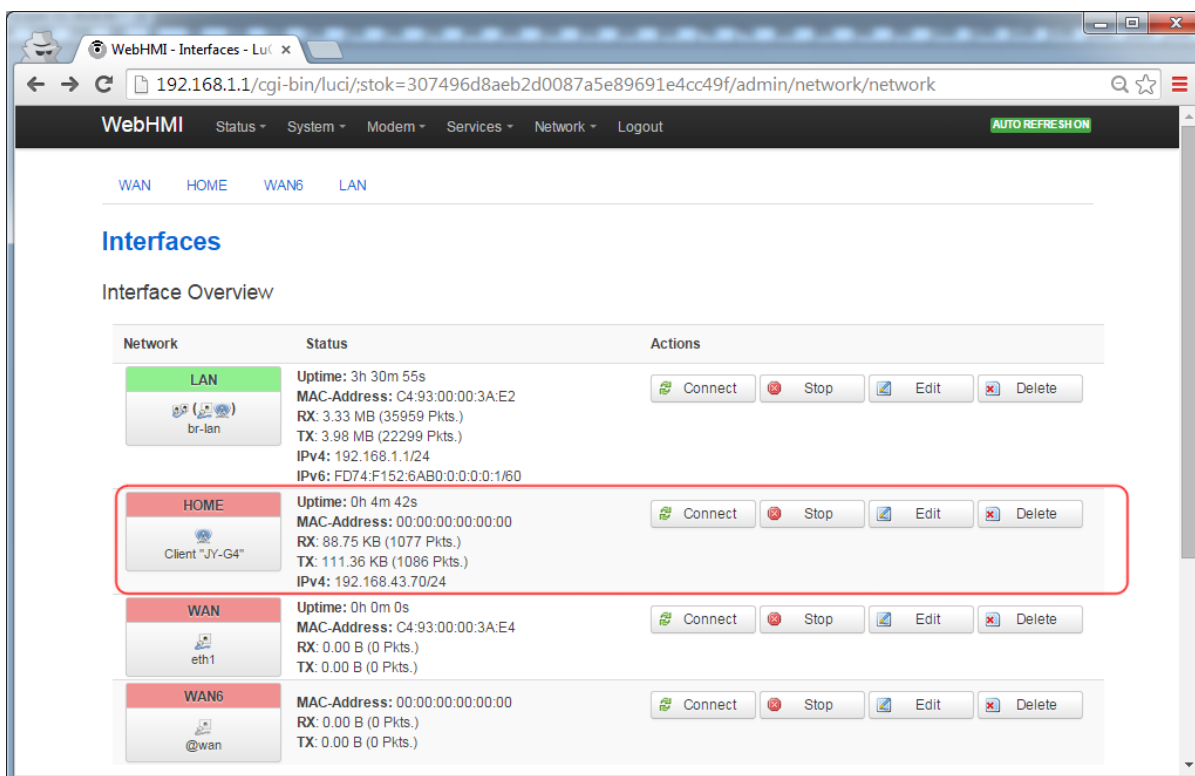
6. В появившемся окне нажмите **Save & Apply**. В принципе это все.



В окне **Wireless Overview** вы увидите новую беспроводную сеть (в данном случае их две: в одной из них WebHmi клиент, а в другой – точка доступа).



Посмотреть или изменить настройки сетевого подключения можно всё там же, во вкладке **Network / Interfaces**



## Беспроводной интернет 3G

При отсутствии проводного подключения к интернету подключить WebHMI можно используя USB-модем.

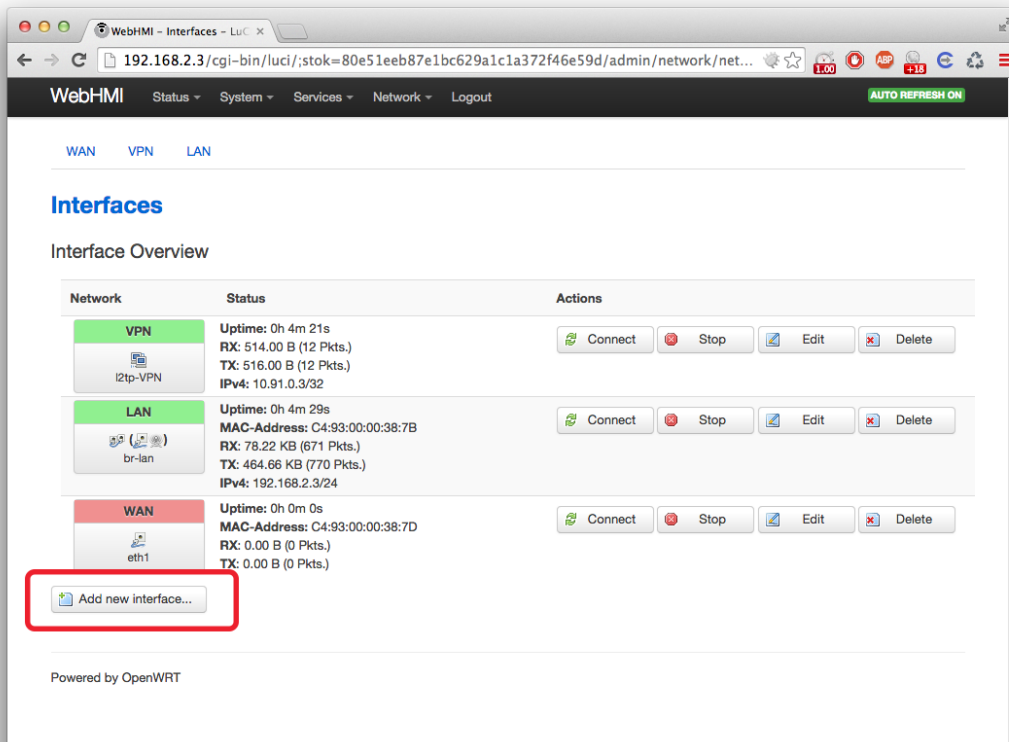
- Подключение к People.net
- Подключение к МТС Коннект

## VPN

В случаях, когда необходим удаленный доступ к устройствам WebHMI, можно воспользоваться технологией VPN.

Свяжитесь с компанией Distributed Data Systems для создания учетной записи для вашего устройства. После этого настройте сетевые соединения согласно следующей инструкции.

1. Перейдите в Network Setup -> Network -> Interfaces. Нажмите кнопку "Add new interface..."



2. Введите VPN в качестве имени соединения. Это имя используется в скрипте watchdog, который проверяет состояние соединения и автоматически может переподключать данный интерфейс. Так что имя интерфейса лучше использовать именно "VPN".

2. Выберите протокол L2TPv2 и нажмите "Apply".

3. На вкладке General Setup укажите такие параметры:

L2TP Server: **webhmicloud.com**

PAP/CHAP username: [логин котовый вам предоставили]

PAP/CHAP password: [пароль котовый вам предоставили]

4. Перейдите на вкладку "Advanced Settings" и введите число 15 или любое другое в поле **Default gateway metric**. Важно что бы Default gateway metric у интерфейса VPN должен быть больше чем Default gateway metric у интерфейса по которому идет соединение с интернетом.

5. На вкладке "Firewall Settings" назначьте зону **lan** для данного интерфейса что бы разрешить входящие соединения. Если этого не сделать то firewall не будет разрешать входящие соединения по VPN и получить доступ к устройству извне не получится.

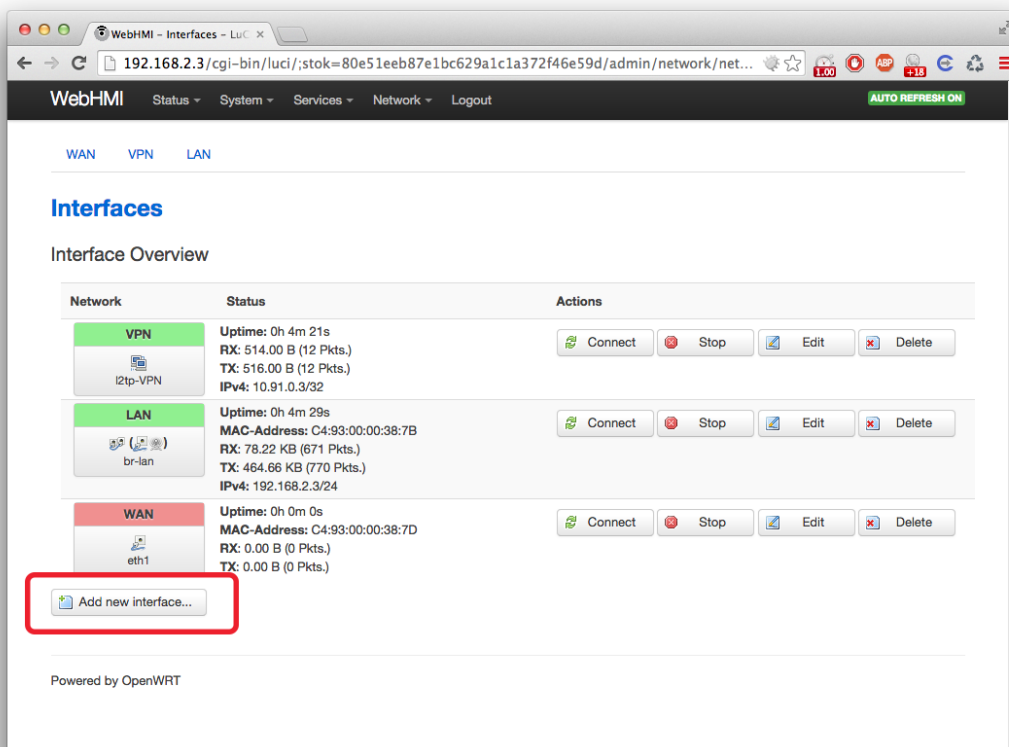
# Подключение к People.net

---

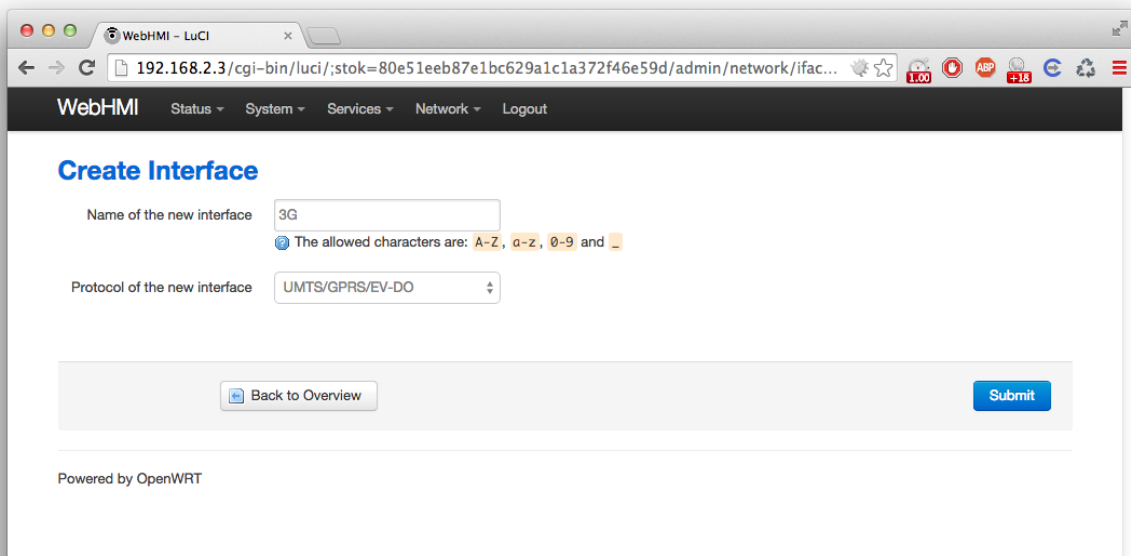
Рассмотрим процесс настройки к оператору People.net на примере CDMA-модема Novatel U760.



1. Перейдите в интерфейс сетевых настроек. Откройте страницу Network->Interfaces. Нажмите кнопку "Add new interface".



2. В качестве имени интерфейса укажите "3G". Выберите протокол "UMTS/GPRS/EV-DO". Нажмите "Submit".



3. В настройках укажите следующие параметры:

Protocol: UMTS/GPRS/EV-DO

Modem device: /dev/ttyUSB1 (зависит от количества USB-устройств в системе и от конкретного модема)

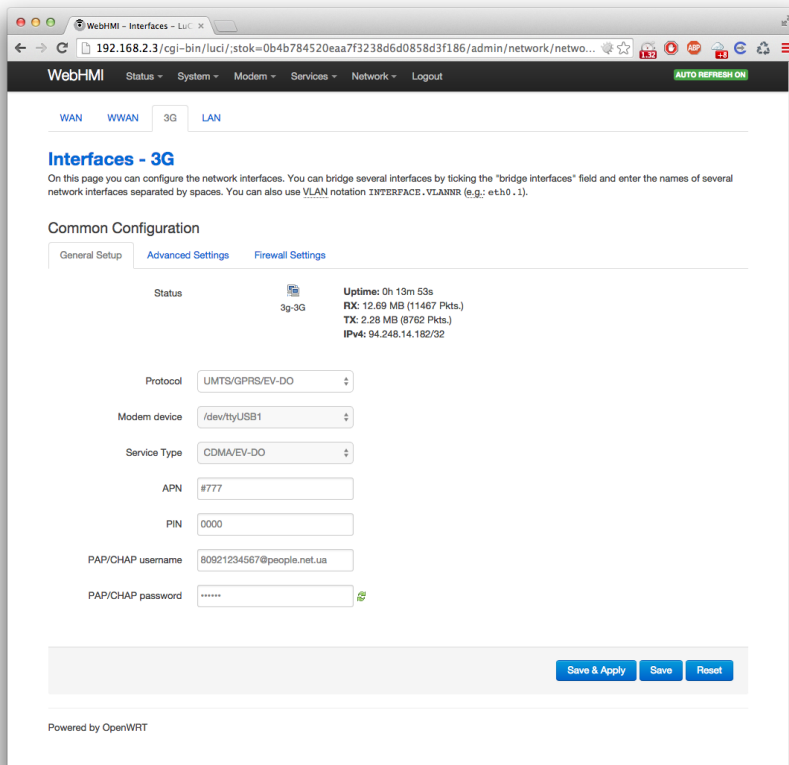
Service Type: CDMA/EV-DO

APN: #777

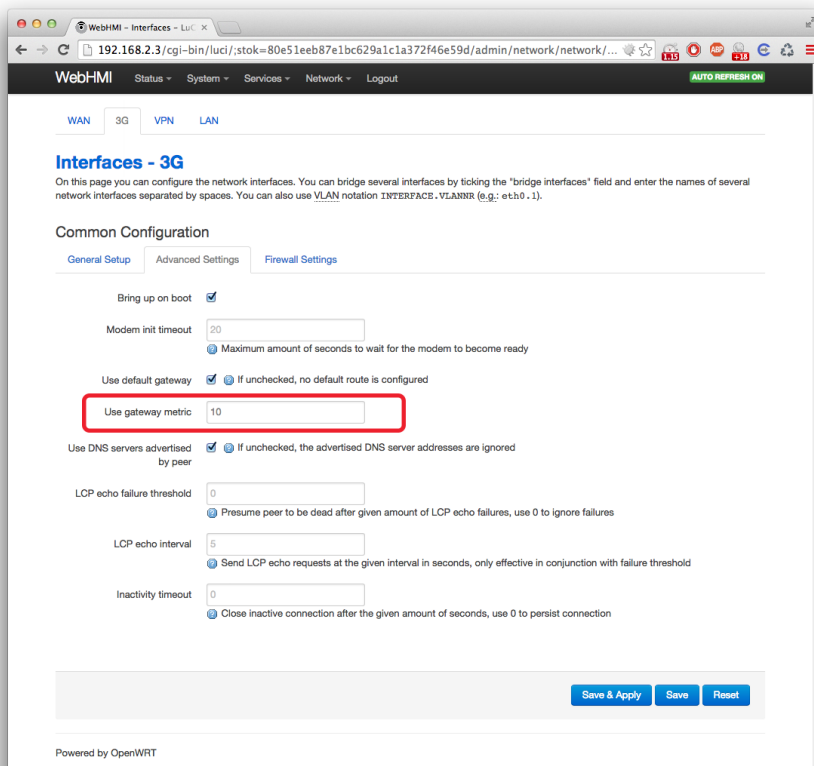
PIN: 0000

PAP/CHAP username: 8092xxxxxxx@people.net.ua (вместо 8092xxxxxxx укажите номер телефона)

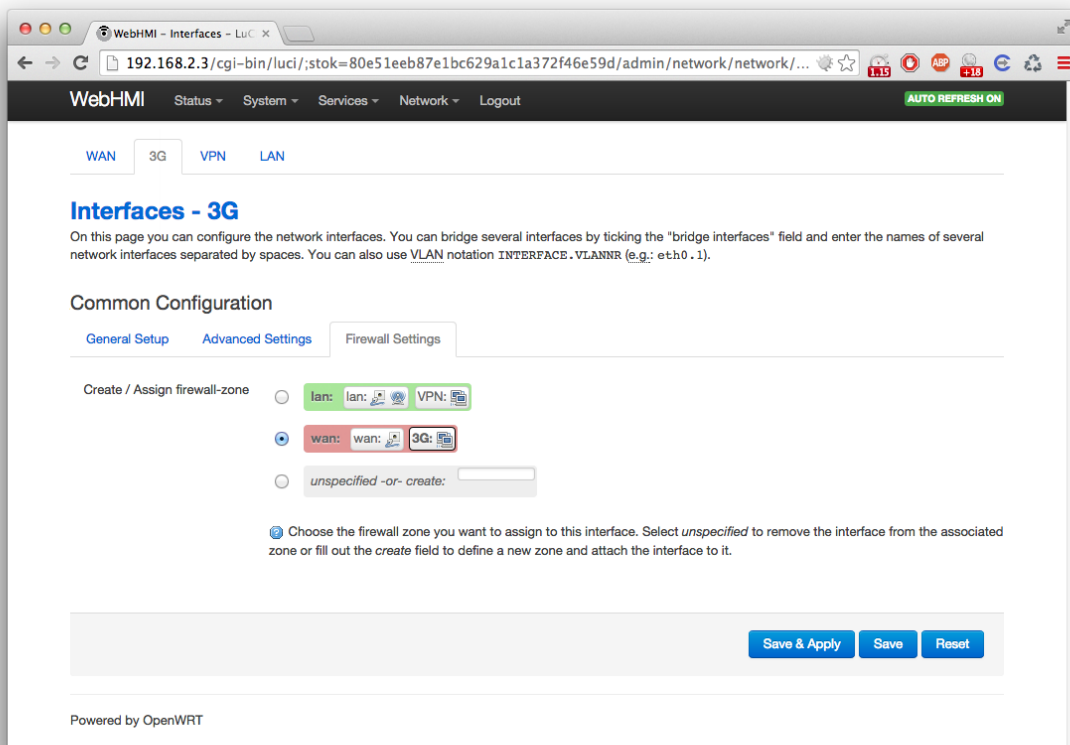
PAP/CHAP password: 000000



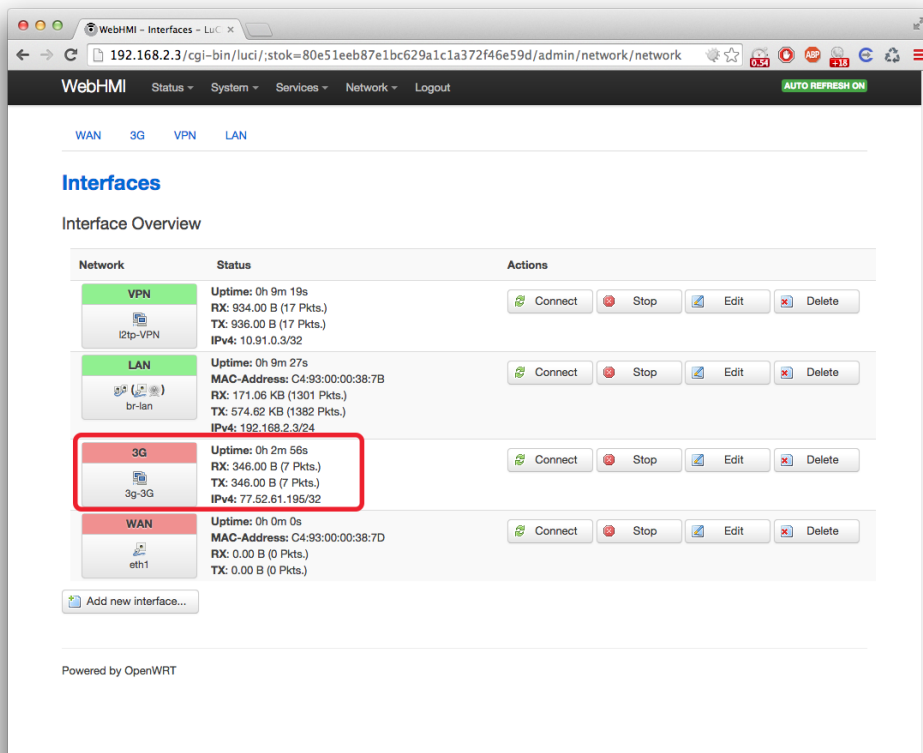
4. Если используется VPN-соединение то необходимо указать метрику шлюза. Она должна быть меньше чем метрика шлюза у VPN-соединения.



5. В настройках Firewall укажите зону WAN что бы исключить входящие соединения из интернета. Как правило, они не нужны и приносят дополнительные риски безопасности.

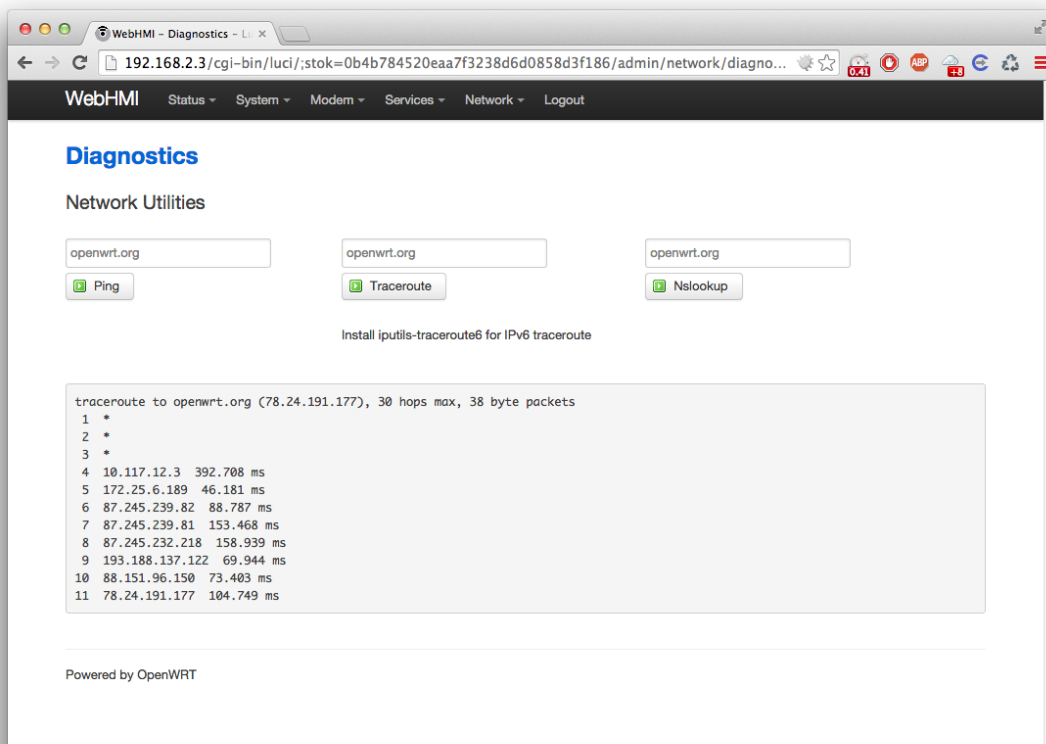


6. Нажмите "Save and apply". Настройки будут применены. Если все сделано правильно то на странице Network->Interfaces вы увидите новый интерфейс "3G" с установленным соединением.



7. Что бы проверить соединение с интернетом перейдите на страницу Network->Diagnostics и нажмите кнопку Traceroute. Если соединение с интернетом работает корректно вы увидите путь пакетов к указанному серверу

(по умолчанию openwrt.org).



Все, интернет подключен. Можно пользоваться.

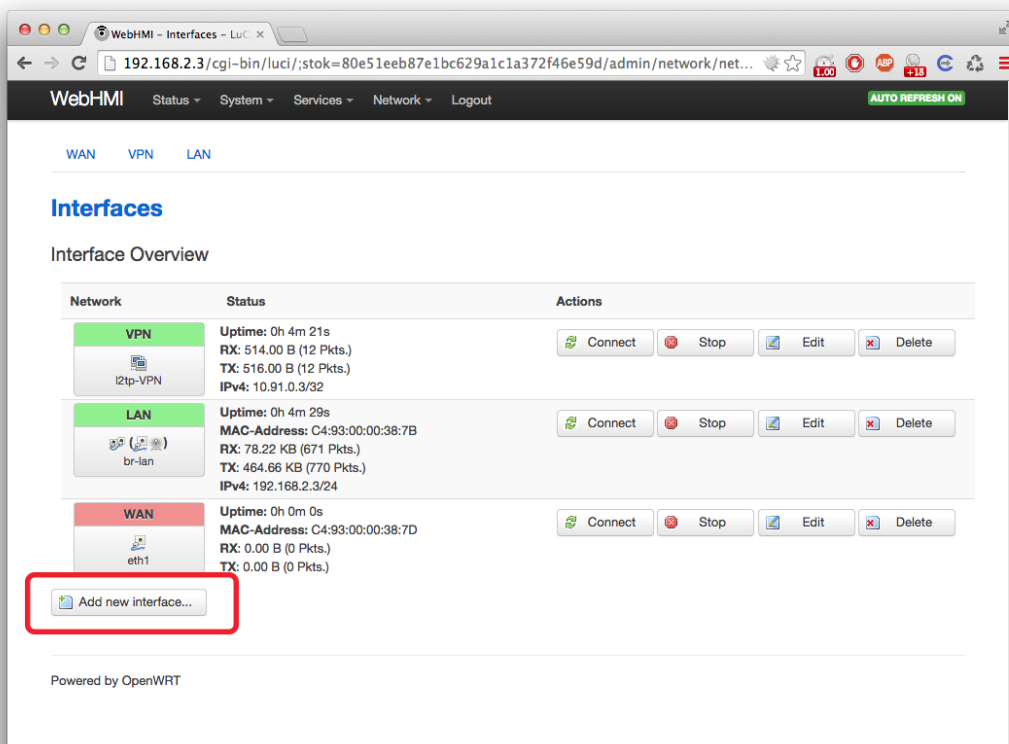
## Подключение к МТС Коннект

Рассмотрим процесс настройки к оператору МТС на примере CDMA-модема WeTelecom WM-D200.

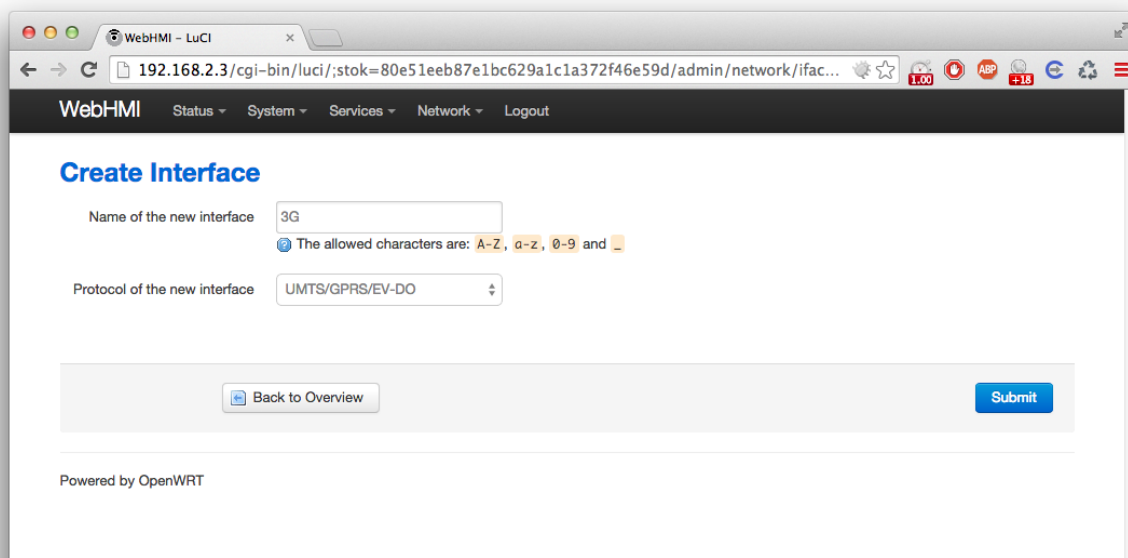


1. Перейдите в интерфейс сетевых настроек. Откройте страницу Network->Interfaces. Нажмите кнопку "Add new interface".





2. В качестве имени интерфейса укажите "3G". Выберите протокол "UMTS/GPRS/EV-DO". Нажмите "Submit".



3. В настройках укажите следующие параметры:

Protocol: UMTS/GPRS/EV-DO

Modem device: /dev/ttyUSB1 (зависит от количества USB-устройств в системе и от конкретного модема)

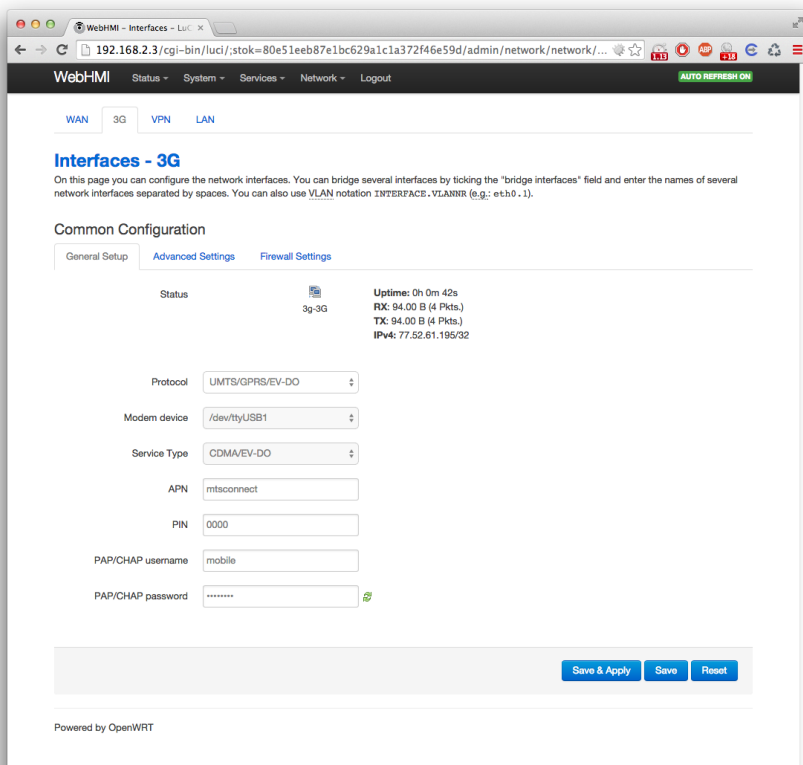
Service Type: CDMA/EV-DO

APN: mtsconnect

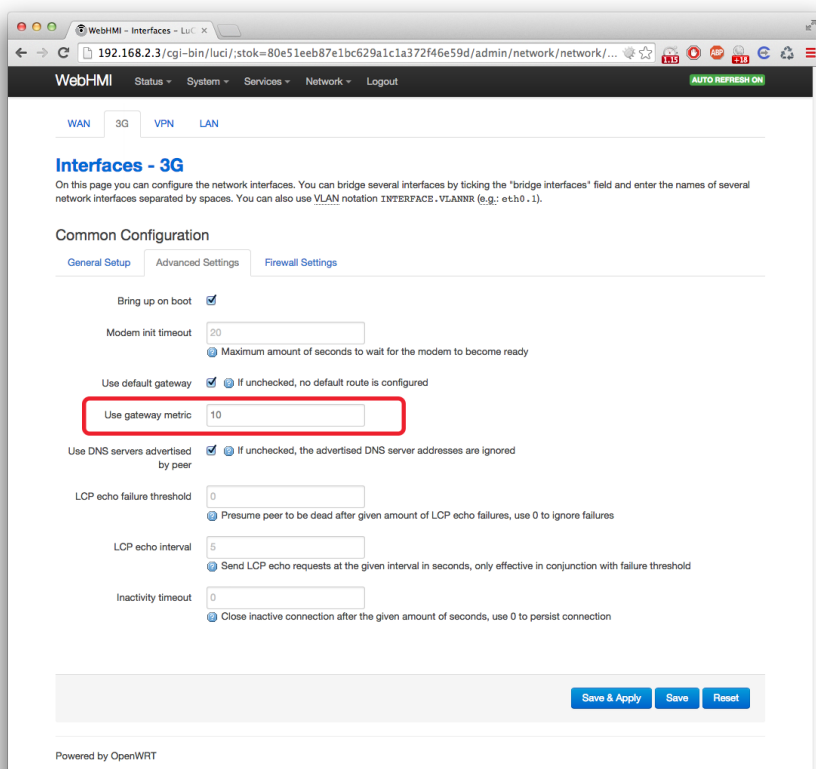
PIN: 0000

PAP/CHAP username: mobile

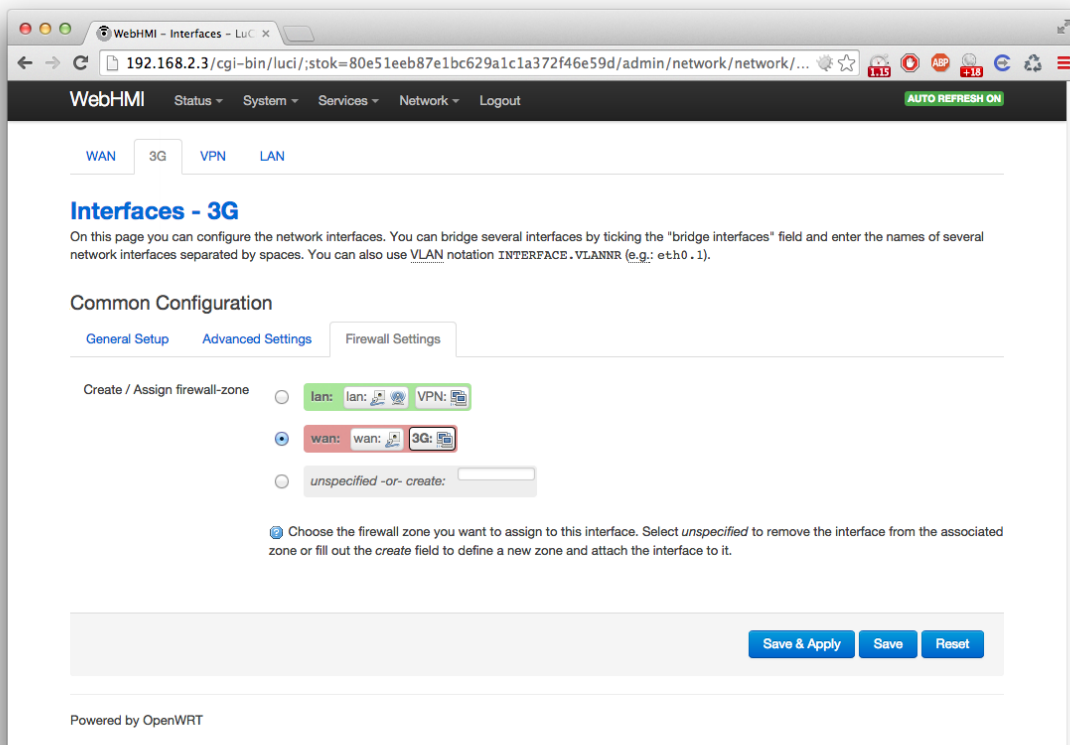
PAP/CHAP password: internet



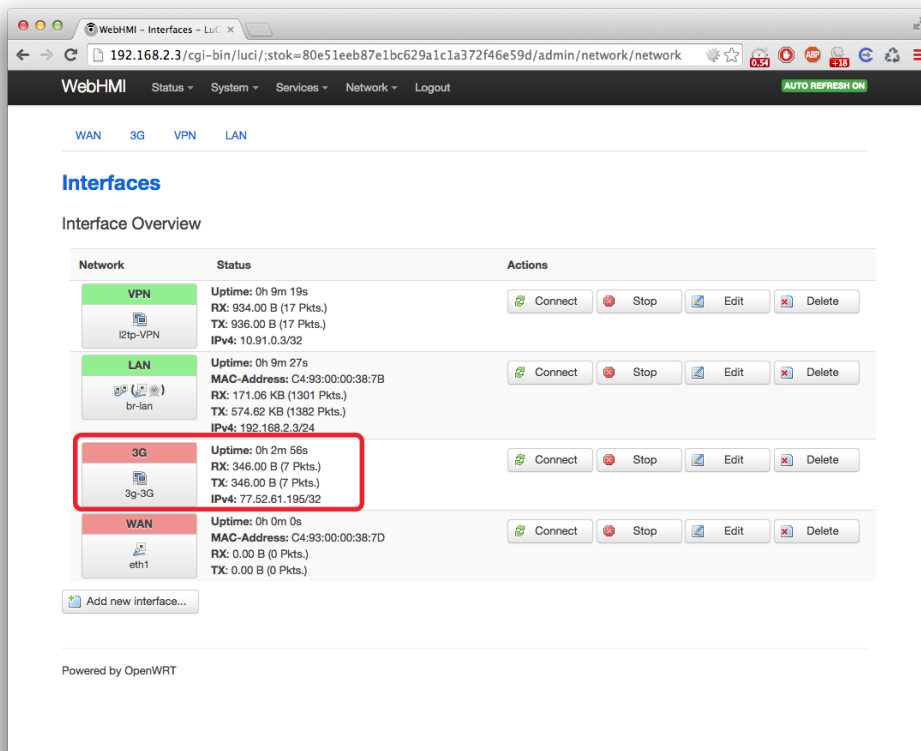
4. Если используется VPN-соединение то необходимо указать метрику шлюза. Она должна быть меньше чем метрика шлюза у VPN-соединения.



5. В настройках Firewall укажите зону WAN что бы исключить входящие соединения из интернета. Как правило, они не нужны и приносят дополнительные риски безопасности.

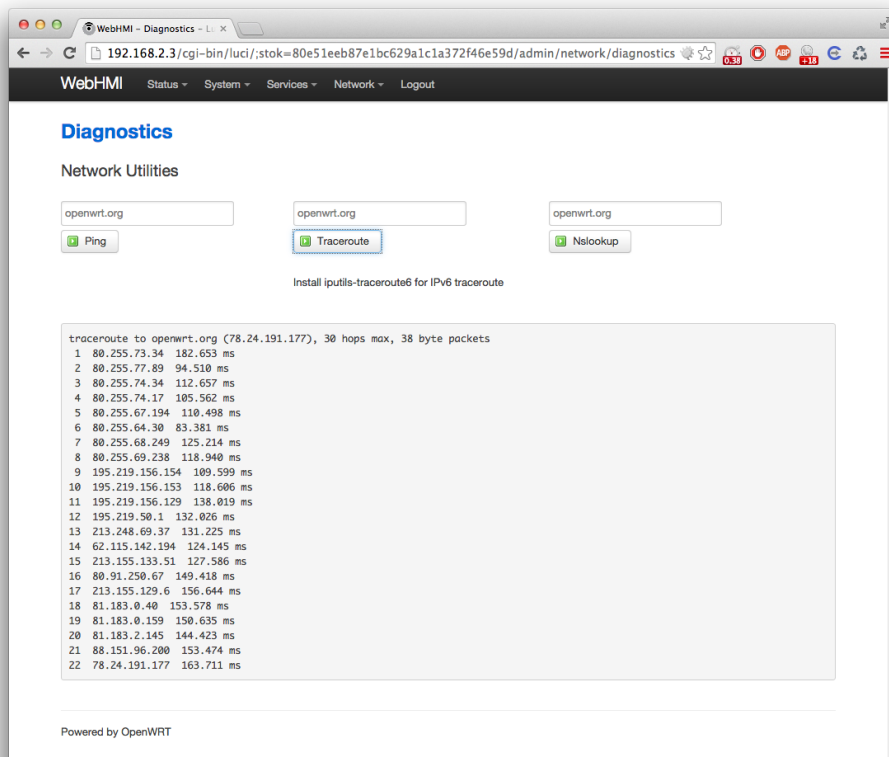


6. Нажмите "Save and apply". Настройки будут применены. Если все сделано правильно то на странице Network->Interfaces вы увидите новый интерфейс "3G" с установленным соединением.



7. Что бы проверить соединение с интернетом перейдите на страницу Network->Diagnostics и нажмите кнопку Traceroute. Если соединение с интернетом работает корректно вы увидите путь пакетов к указанному серверу

(по умолчанию openwrt.org).



The screenshot shows the OpenWRT WebUI Diagnostics page. The browser address bar displays the URL: `192.168.2.3/cgi-bin/luci/stok=80e51eeb87e1bc629a1c1a372f46e59d/admin/network/diagnostics`. The page title is "WebUI - Diagnostics". The navigation menu includes "Status", "System", "Service", "Network", and "Logout". The main heading is "Diagnostics" and the sub-heading is "Network Utilities". There are three input fields, each containing "openwrt.org", with buttons for "Ping", "Traceroute", and "Nslookup". The "Traceroute" button is highlighted. Below the buttons, there is a message: "Install iputils-traceroute6 for IPv6 traceroute". The main content area displays the results of a traceroute to openwrt.org (78.24.191.177), showing 30 hops max and 38 byte packets. The results are as follows:

```
traceroute to openwrt.org (78.24.191.177), 30 hops max, 38 byte packets
 1 80.255.73.34 182.653 ms
 2 80.255.77.89 94.510 ms
 3 80.255.74.34 112.657 ms
 4 80.255.74.17 105.562 ms
 5 80.255.67.194 110.498 ms
 6 80.255.64.30 83.381 ms
 7 80.255.68.249 125.214 ms
 8 80.255.69.238 118.940 ms
 9 195.219.156.154 109.599 ms
10 195.219.156.153 118.606 ms
11 195.219.156.129 138.019 ms
12 195.219.50.1 132.026 ms
13 213.248.69.37 131.225 ms
14 62.115.142.194 124.145 ms
15 213.155.133.51 127.586 ms
16 80.91.250.67 149.418 ms
17 213.155.129.6 156.644 ms
18 81.183.0.40 153.578 ms
19 81.183.0.159 150.635 ms
20 81.183.2.145 144.423 ms
21 88.151.96.200 153.474 ms
22 78.24.191.177 163.711 ms
```

At the bottom of the page, it says "Powered by OpenWRT".

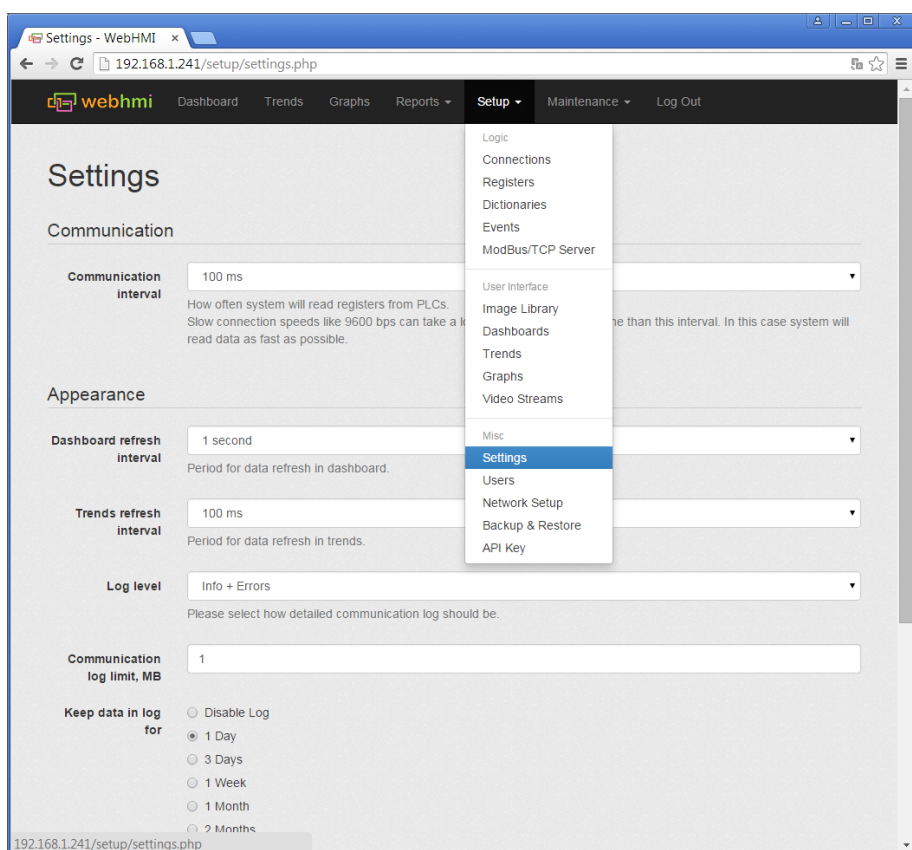
Все, интернет подключен. Можно пользоваться.

# Системные настройки

## Системные настройки и сервис

### Системные настройки

Системные настройки задают параметры производительности, глубину логов данных и детализацию системных логов, включают дополнительные функции - ModBus/TCP server, WebHMI Cloud.



**Communication interval** - задает интервал работы обмена с подключенными устройствами. По истечении этого времени WEBHMI начнет цикл непрерывного обмена с устройствами на соответствующих им скоростях с минимальными задержками между последовательными опросами. Короткий интервал увеличивает частоту опроса, уменьшает "отклик" при изменении регистра на приборной панели и "отклик" изменений на трендах, однако при большом числе опрашиваемых данных они могут по времени не поместиться в этот интервал, также это увеличивает нагрузку на систему. Применение настройки должно быть оправданным. При необходимости "успевать" опрашивать большой объем данных в течение секунды, в первую очередь необходимо повышать скорость связи с устройствами.

**Dashboard refresh interval** - частота обновления данных на приборной панели.

**Trends refresh interval** - частота обновления данных на тренде, возможные значения:

- 50, 100, 200, 333, 500, 800 мс
- 1, 1.5, 2, 5 сек.

**Log level** - уровень детализации системного лога, возможные значения:

- Disable log - отключен;
- Errors - только ошибки;
- Info+Errors - информационные сообщения + ошибки;
- Debug+Info+Errors - отладочная информация + информационные сообщения + ошибки;
- Trace+Debug+Info+Errors - данные трассировки, отладочная информация + информационные сообщения + ошибки;

**Communication log limit, MB** - Размер коммуникационного лога, Мб;

The screenshot shows a settings panel with the following sections:

- Keep data in log for**: A group of radio buttons for selecting the retention period. The options are: Disable Log, 1 Day, 3 Days, 1 Week (selected), 1 Month, 2 Months, 3 Months, 4 Months, 1 Year, and 5 Years.
- MobBus server**: A section with a checkbox labeled "Enable local ModBus/TCP server".
- WebHMI Cloud**: A section with a checkbox labeled "Enable WebHMI Cloud integration".
- At the bottom, there are two buttons: a blue "Save" button and a red "Export Settings to WebHMI Cloud" button.

**Keep data in log for**- настройка глубины хранения сохраненных данных (логи регистров, событий, данные графиков)

**MobBus server**- включение опции делает возможным обращение к WEBHMI по протоколу Modbus/TCP (см. описание функции)

**WebHMI Cloud**- включение опции делает возможным отправку данных WEBHMI на облачный сервер описание функции)

- WebHMI Cloud API Key - ключ API, используя который будут отправляться данные на сервер
- Real-time registers values send interval, Log data send interval - периоды отправки на облачный сервер данных регистров и лога соответственно
  - 1,5,15 секунд
  - 1,5,15,30 минут
  - 1,6,12 часов
  - 1 день

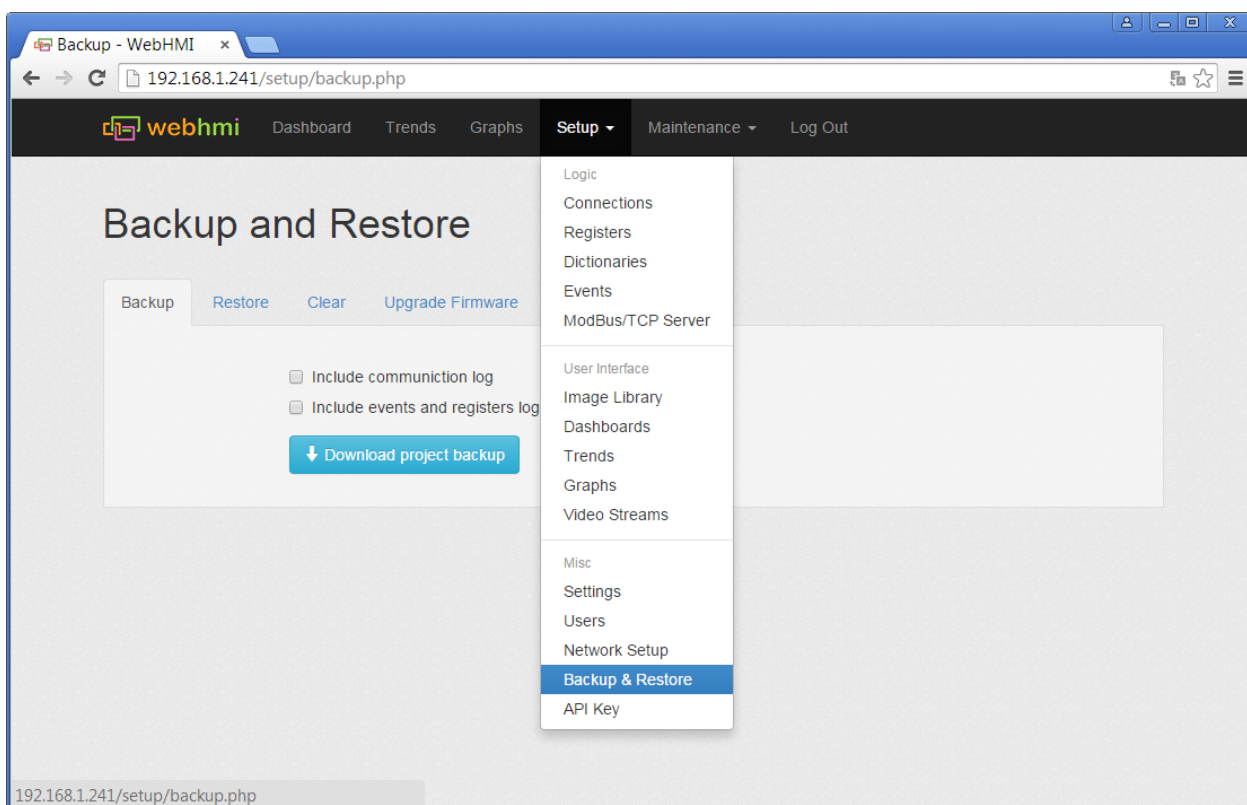
## Резервное копирование и восстановление

Пункт меню **Setup/Backup&Restore** позволяет делать резервное копирование проекта, его восстановлению из архива, очистку данных с флеш-карты, а также удаленно делать обновление версии прошивки.

Вкладка **Backup** - резервное копирование проекта, который может быть дополнен логами при выборе соответствующих опций :

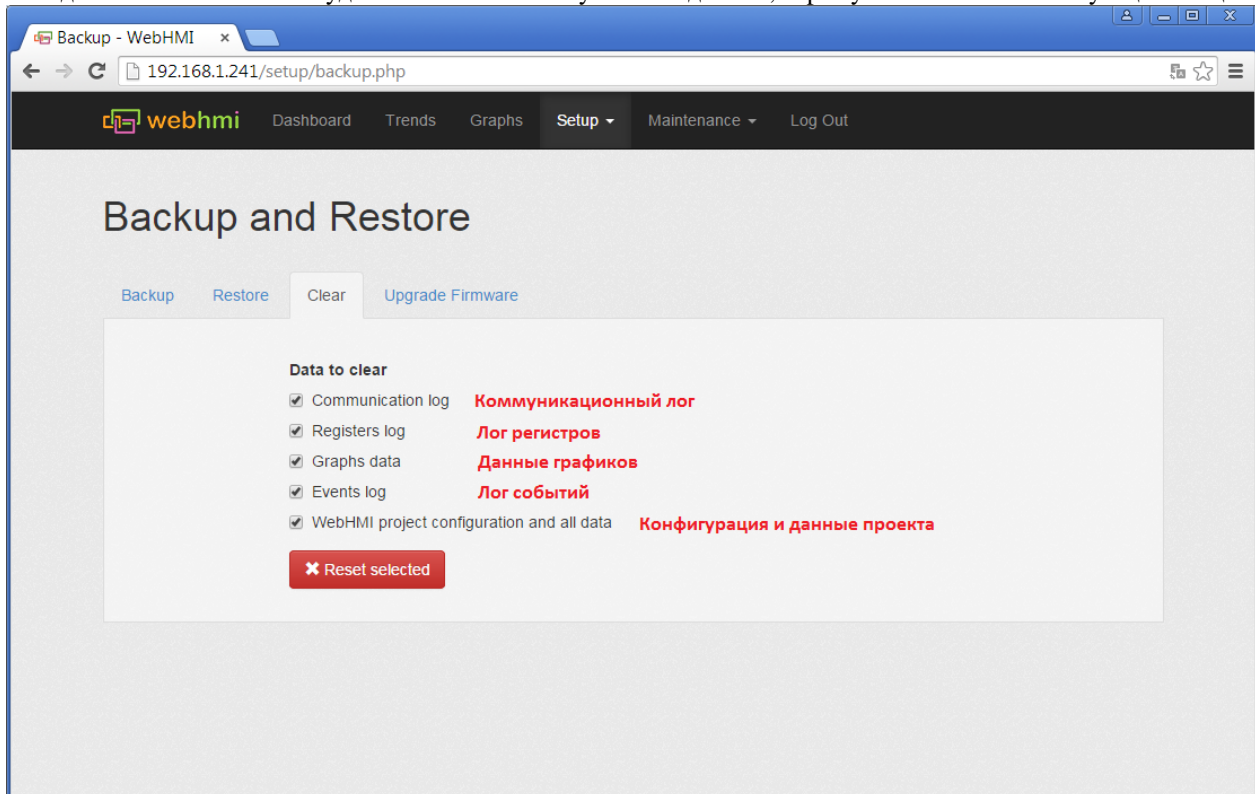
- коммуникационным
- регистров, событий

Примечание. При большом размере лога (более 4МБ) он автоматически не будет включен в резервную копию. Для сохранения лога большого объема можно скопировать его с SD карты, либо, зайдя на WEBHMI по протоколу ftp, (с теми же настройками, что и для входа в настройки сети) скопировать папку \log. Соответственно чтобы восстановить данные, необходимо также скопировать эту папку обратно на SD-карту.



Вкладка **Restore** - загрузка резервной копии проекта. См. примечание к описанию функции Backup.

Вкладка **Clear** позволяет удалить ставшие ненужными данные, при указании соответствующих опций:



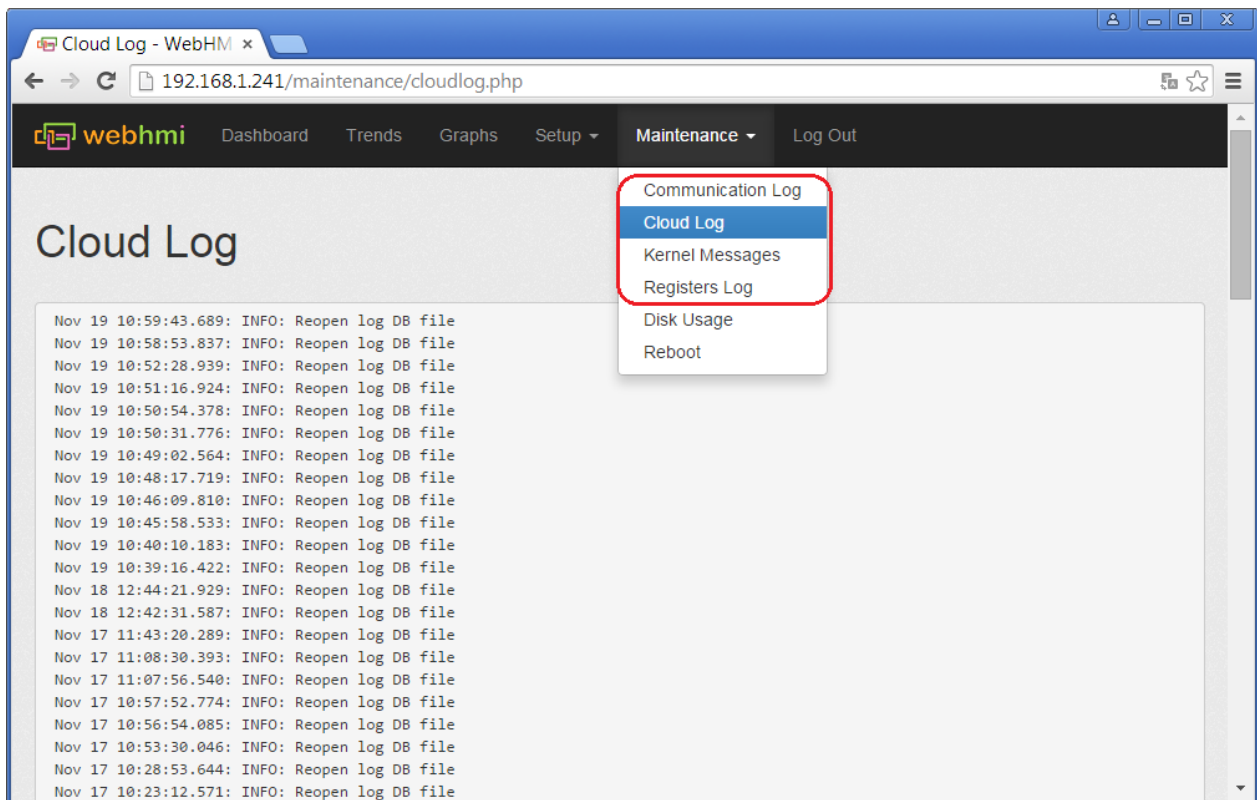
Вкладка **Upgrade Firmware** - обновление версии прошивки. При обновлении удаляется текущий проект и все данные. Перед выполнением этой операции необходимо проконсультироваться с технической службой продавца, поскольку для ее выполнения необходим пароль для доступа на сервер обновлений. Кроме того, при слишком большой разнице версий новой и старой прошивок, проект сохраненный в старой версии, может загрузиться не полностью или с ошибками.

## Системная информация и логи

Меню **Maintenance** служит для просмотра текущих логов:

- коммуникационный
- обмена с облачным сервером
- ядра ОС
- регистров

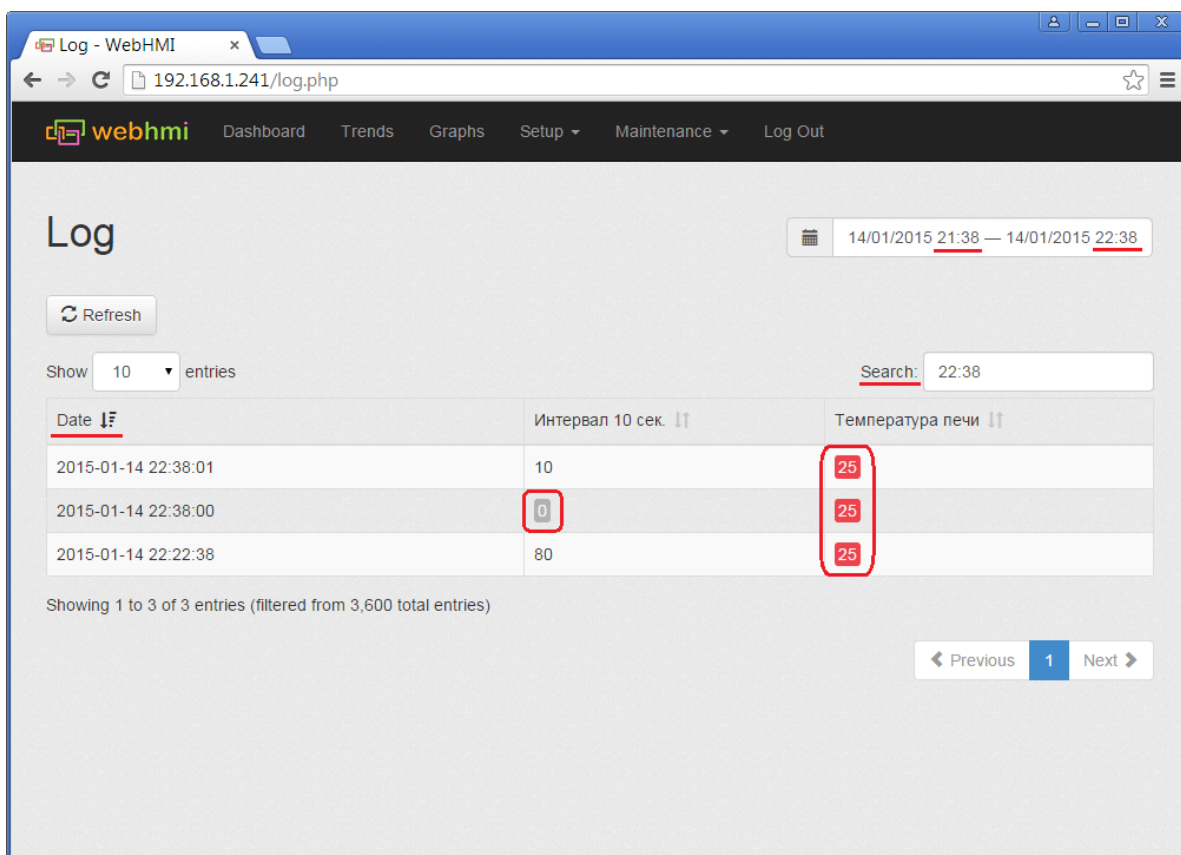




Первые три необходимы для диагностики системы и для пользователя представляют интерес лишь в случае каких-либо неполадок в системе - в этом случае данные логи следует передать в службу технической поддержки продавца.

**Registers log** позволяет протолировать изменения состояния регистров (для которых включена опция записи в лог регистров) либо их текущие значения с определенной периодичностью. Значения в лог также имеют цветовую индикацию состояний регистров, что очень удобно для нахождения моментов времени выхода параметров за нежелательные границы (предаварийные, аварийные и т.д.). Подробно настройка протолирования значений регистров описана в разделе Работа с регистрами, п. 4.1. Настройки лога позволяют:

- выбирать произвольный интервал дат для просмотра
- осуществлять фильтрацию списка по значению, дате
- сортировать список



The screenshot shows the WebHMI Log page. The browser address bar displays "192.168.1.241/log.php". The page title is "Log". The date range is "14/01/2015 21:38 — 14/01/2015 22:38". The search filter is "22:38". The table shows three entries with the following data:

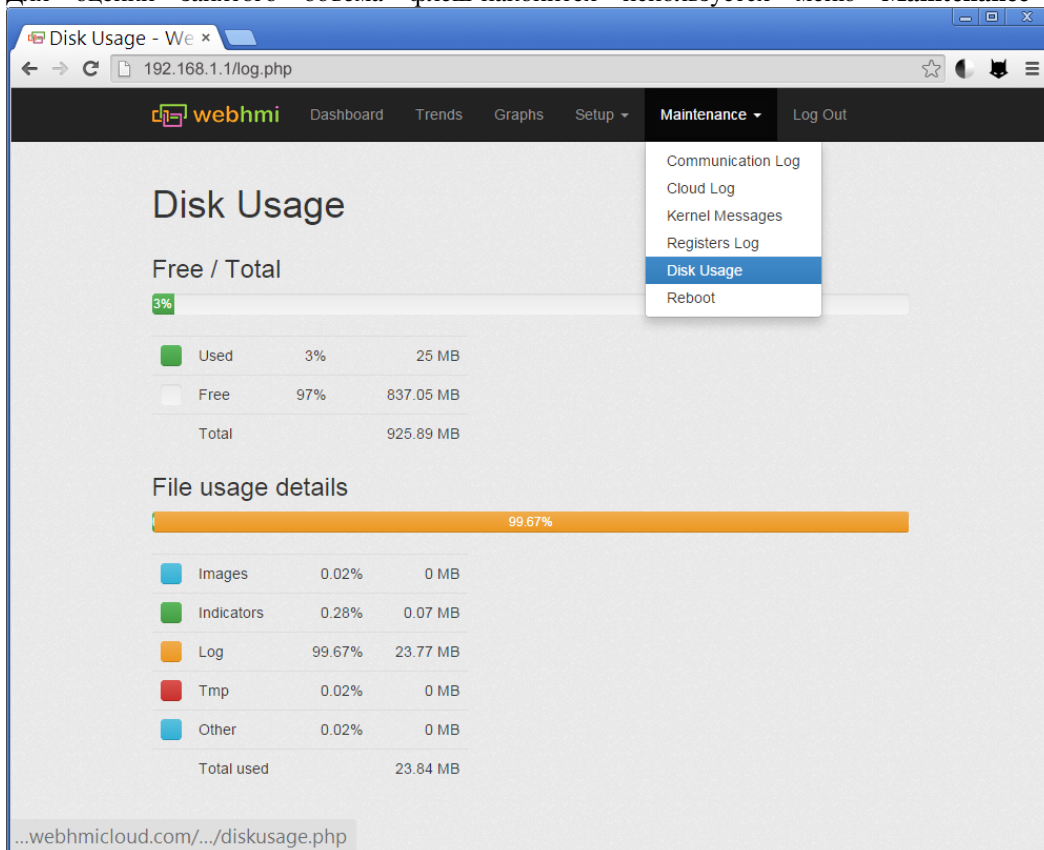
Date	Интервал 10 сек.	Температура печи
2015-01-14 22:38:01	10	25
2015-01-14 22:38:00	0	25
2015-01-14 22:22:38	80	25

Showing 1 to 3 of 3 entries (filtered from 3,600 total entries)

В приведенном примере произведена выборка за последний час, с фильтром поиска "22:38", сортировкой даты "по убыванию", выделены значения с сигнализаций состояний - серый (disable или отключен) и красный (alert или тревога).

## Информация об использовании места на накопителе

Для оценки занятого объема флеш-накопителя используется меню **Maintenance** --> **Disk Usage**



### Общая информация

- **Used** - использовано места
- **Free** - свободное место
- **Total** - Всего

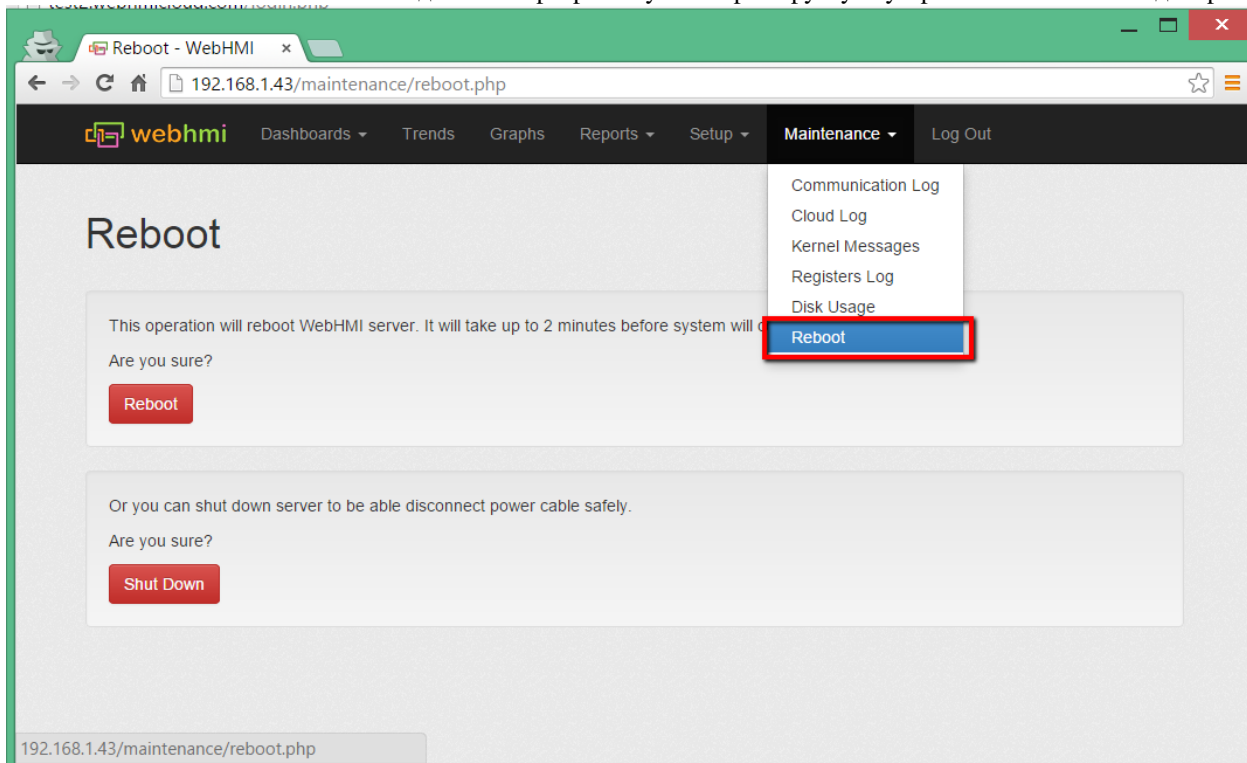
### Детали

- **Images** - занято изображениями
- **Индикаторы** - занято индикаторами
- **Log** - лог файлы
- **Tmp** - временные файлы
- **Other** - остальное

## Перезагрузка и выключение

Перед выключением WEBHMI рекомендуется делать "программное выключение" с целью остановки возможных файловых операций и остановки ОС, для предотвращения возможных ошибок работы с флеш-накопителем (для версий без встроенной батареи). Версии с батареей будут делать это автоматически при пропадании питания.

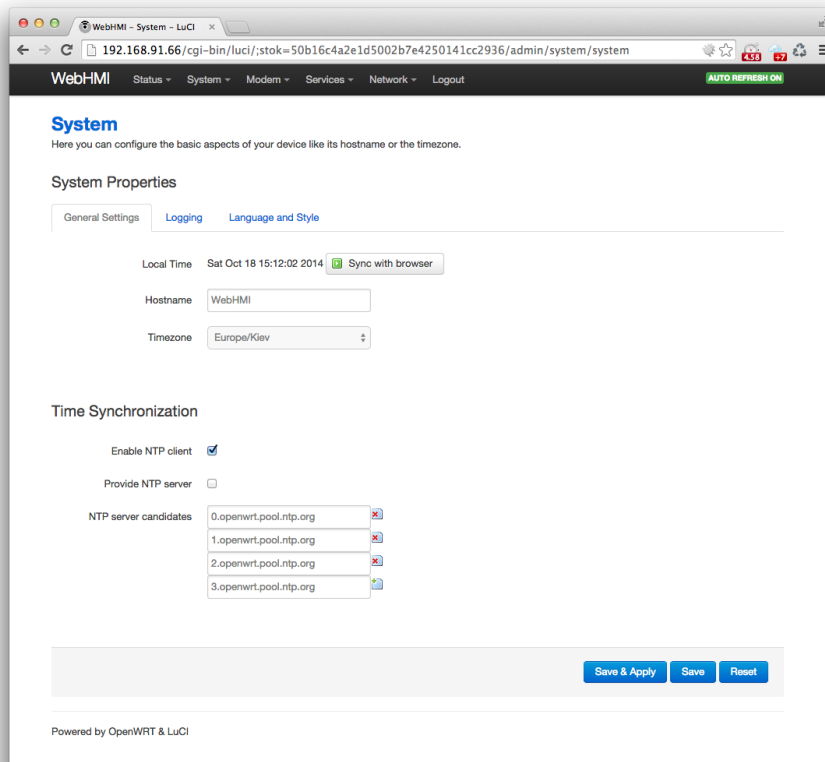
Также имеется возможность сделать программную перезагрузку устройства. См. след. рис.:



# Синхронизация времени

В WebNMI есть NTP<sup>[1]</sup>-сервер. Он позволяет синхронизировать локальные часы с другим сервером.

Настройка этого сервиса выполняется на вкладке System->System интерфейса настройки сети:



По умолчанию NTP-сервер включен в режиме клиента и синхронизирует время с серверами OpenWRT. Для корректной синхронизации времени с этими серверами необходим работающий доступ к интернету.

В **NTP server candidates** нужно указать список серверов с которыми вы желаете синхронизировать время.

При отсутствии доступа к другому NTP-серверу установка времени выполняется с помощью кнопки "**Sync with browser**".

Галочка "**Enable NTP client**" позволяет включить или выключить сервис NTP. Если сервис включен то система будет периодически синхронизировать время с указанными серверами. Время будет подстраиваться *плавно*, без резких изменений. Если сервис отключен то система будет синхронизировать внутренние часы с часами реального времени установленными на борту. Синхронизация будет проходить каждый час на 10-й минуте часа. Коррекция будет выполняться единомоментно, без плавной адаптации часов.

Галочка "**Provide NTP server**" позволяет включить режим сервера NTP. Это позволит данному устройству также быть NTP-сервером что бы другие устройства в сети могли синхронизировать свои часы с ним. Номер порта - UDP 123.

Если доступа к внешним NTP-серверам нет, но вы хотите сделать WebNMI NTP-сервером в локальной сети что бы все другие устройства синхронизировали свое время по ней то включите галочку "**Provide NTP server**" и очистите список **NTP server candidates**. В этом случае WebNMI будет синхронизироваться с аппаратными часами реального времени и предоставлять свое локальное время для NTP-клиентов.

## Сброс настроек

---

В некоторых случаях может потребоваться сбросить настройки WebNMI. Например, если вы забыли пароль доступа или настройки сетевого соединения.

Для выполнения сброса необходимо в момент включения устройства нажать и удерживать кнопку "Factory reset" (расположена на передней панели устройства). Через некоторое время (примерно через 20-30 секунд; в зависимости от версии прошивки возможно дольше) мигнет красный светодиод "Error". Продолжайте держать кнопку "Factory reset" нажатой. После первого мигания светодиод "Error" потухнет а затем загорится на несколько секунд снова.

В этот момент кнопку "Factory reset" уже можно отпустить. В зависимости от объема накопленных данных операция очистки может занять несколько секунд или больше. После сброса система перезагрузится еще раз что бы применить новые настройки.

Сброс очистит такие данные:

- сетевые настройки
- настройки firewall
- проект WebNMI
- загруженные изображения
- базу данных со всеми накопленными данными
- журналы WebNMI и WebNMI Cloud
- пароль входа в Network Setup (LUCI)
- настройки DHCP и DDNS
- настройки NTP

Таким образом WebNMI будет восстановлена к заводскому состоянию.

---

# Взаимодействие с другими устройствами и сбор данных

## Соединения

Соединение определяет, какой коммуникационный драйвер и какие параметры использовать WEBHMI для обмена с оборудованием. Список поддерживаемых протоколов и их параметры приведены здесь. Для создания соединения (подменю **Setup/Connections**) используется кнопка "Add connection". После нажатия пользователю предлагается ввести следующую информацию:

- Базовые настройки

**Title** - Название соединения, которое будет отображаться при работе с проектом

**PLC Model** - Тип устройства, к которому необходимо подключаться

**Protocol** - Какой протокол обмена будет использоваться

**PLC adress** - Адрес подключенного устройства в сети

**Disable** - Возможность приостановить обмен по данному соединению

The screenshot shows the 'Edit Connection' form in the webhmi interface. The form is divided into two main sections: 'Basic Info' and 'Communication'. In the 'Basic Info' section, there are four fields: 'Title' (PPI\_1), 'PLC model' (Siemens Simatic S7 200), 'Protocol' (Siemens PPI), and 'PLC address' (2). There is also a 'Disable' checkbox. In the 'Communication' section, there are two fields: 'Device' (RS-485 (WebHMI)) and 'Baud rate' (19200). The form is displayed in a browser window with the URL 192.168.1.43/setup/plcs.php?edit=2.

- Настройки связи

**Device** - порт WEBHMI по которому будет происходить обмен, RS-485 - встроенный порт, /dev/ttyUSB0 - порт внешнего преобразователя USB-Serial(их может быть несколько).

**Baud rate** - скорость обмена

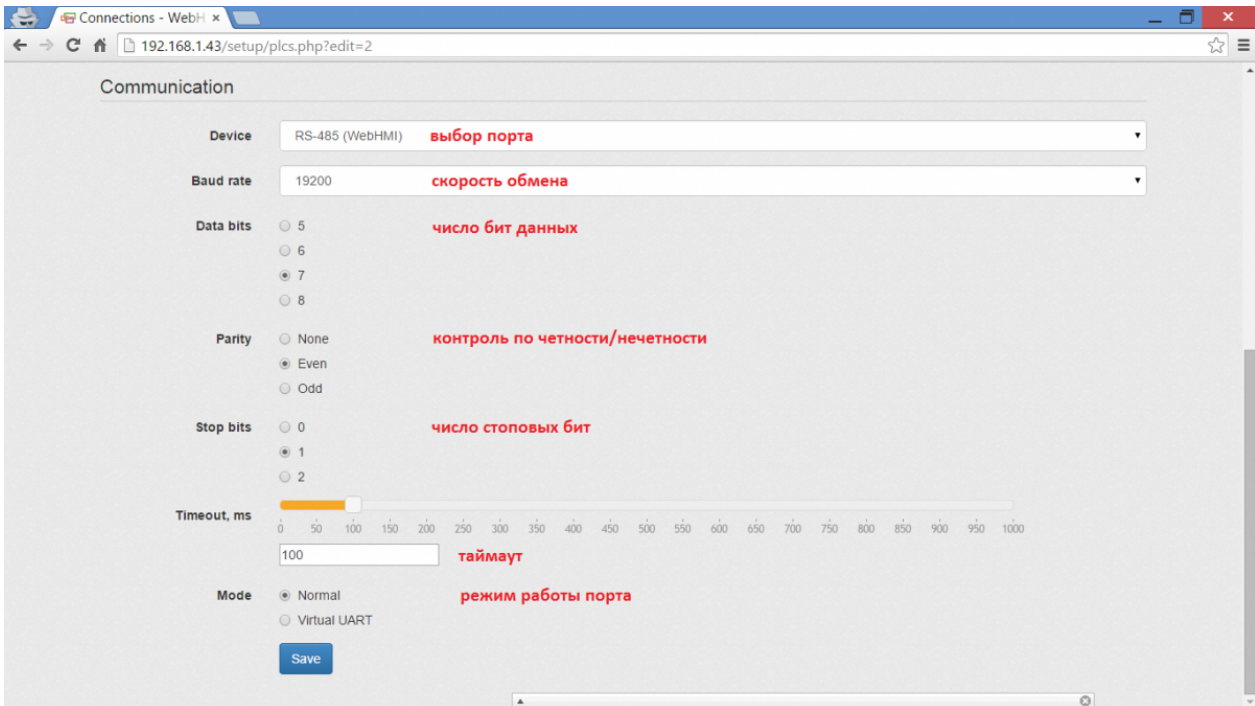
**Data bits** - количество бит данных в кадре

**Parity** - используется ли контроль четности/нечетности

**Stop bits** - количество стоповых бит

**Timeout,ms** - таймаут ожидания ответа по шине, мс.

**Mode** - Режим работы последовательного порта Normal/Virtual. В режиме Virtual порт начинает работать как шлюз Ethernet-serial. Т.е. на компьютере-клиенте устанавливается драйвер виртуального СОМ-порта и таким образом удаленный по сети компьютер получает доступ к последовательному порту (например для программирования или диагностики контроллера). Подробнее о настройке виртуального порта



Примеры настройки параметров связи для различных ПЛК:

Allen-Bradley Micrologix 1200 серии: Контроллеры этой серии имеют 1 или 2 последовательных порта. Порт программатора всегда имеет следующие настройки (фрагмент оригинальной документации Publication 1762-UM001G-EN-P March2011):

### Default Communication Configuration

The MicroLogix 1200 has the following default communication configuration. The same default configuration is applied for both Channel 0 and the Programmer/HMI Port (for 1762-LxxxxR only). The configurations for the Programmer/HMI Port are fixed and you cannot change them.

**TIP**

For Channel 0, the default configuration is present when:

- The controller is powered-up for the first time.
- The communications toggle push button specifies default communications (the DCOMM LED is on).
- An OS upgrade is completed.

See Appendix E for more information about communicating.

**Table 4.1 DF1 Full-duplex Default Configuration Parameters**

Parameter	Default
Baud Rate	19.2K
Parity	none
Source ID (Node Address)	1
Control Line	no handshaking
Stop Bits	1

Пример настройки данного соединения:



## Базовые

The screenshot shows the 'Add Connection' page in the webHMI interface. The browser address bar shows '192.168.1.43/setup/plcs.php?add=1'. The page has a navigation menu with 'Setup' selected. The main heading is 'Add Connection'. Below it is the 'Basic Info' section with the following fields:

- Title:** uLogix1200 example (with a note: 'Enter human readable name for this connection.')
- PLC model:** Generic DF1 (with a note: 'Select type/model of your PLC.')
- Protocol:** Allen Bradley DF1
- PLC address:** 1 (with a note: 'Enter bus address of your PLC.')

At the bottom of this section is a checkbox labeled 'Disable'.

## Настройки связи

This screenshot shows the 'Communication' settings section, which is highlighted with a red border. The 'PLC address' field from the previous section is visible at the top, set to '1'. Below it is the 'Communication' section with the following settings:

- Device:** RS-485 (WebHMI)
- Baud rate:** 19200
- Parity:** None (radio buttons for None, Even, Odd)
- Stop bits:** 1 (radio buttons for 0, 1, 2)
- Timeout, ms:** 100 (with a slider and a text input field)
- Mode:** Normal (radio buttons for Normal, Virtual UART)

An 'Add' button is located at the bottom of the 'Communication' section.

# Поддерживаемые протоколы

---

WebHMI версии 1.5 поддерживает следующие протоколы:

- Внутренние регистры WebHMI
- ModBus ASCII
- ModBus RTU
- ModBus TCP
- Siemens PPI
- Delta DVP
- Owen
- Allen Bradley DF1

## Работа с регистрами

---

### Общие сведения

В контексте задач автоматизации сбора данных, управления оборудованием и диспетчеризации под "Регистрами" понимают данные с которыми работают средства и системы автоматизации. Они могут описывать какие-то внешние параметры технологических процессов (значение температуры, давления, расхода, состояния исполнительных механизмов и т.п.) или внутренние состояния самих устройств (параметры работы регуляторов, фильтров и пр.).

Регистры в проекте WEBHMI соответствуют аналогичным понятиям в системах автоматизации - "тег", "переменная" и указывают на то с какими данными из сопряженных устройств мы собираемся работать (читать/записывать), кроме того, они имеют дополнительные атрибуты, управляющие тем, как будут отображаться и обрабатываться системой данные этих регистров. Регистры могут быть двух типов - **внутренние** (из внутренней памяти WEBHMI) и **внешние** (содержащие данные от подключенных устройств). Настройки регистров позволяют задавать:

- Имя регистра, соединение, адрес в сети соединения, временно отключать регистр, т.е. исключать его из списка опрашиваемых данных.
- Тип данных (бит, байт, слово и т.д.), формат (целый, с плав. точкой, знаковый и т.д. ), ед. измерения, задание масштабных и смещающих коэффициентов, границы допустимых значений, границы значений, вводимых оператором , количество знаков после запятой, сопоставление словаря (для регистра с данными типа перечисления, когда каждому номеру записанному в регистре соответствует пункт словаря)
- настраивать запись регистр в лог и на графики
- задавать 4 состояния сигнализации (цветовой) регистра - отключен, норма, предупреждение, авария. При этом визуальные элементы (текст, пиктограммы) и данные в логе меняют цвет соответственно текущему состоянию.

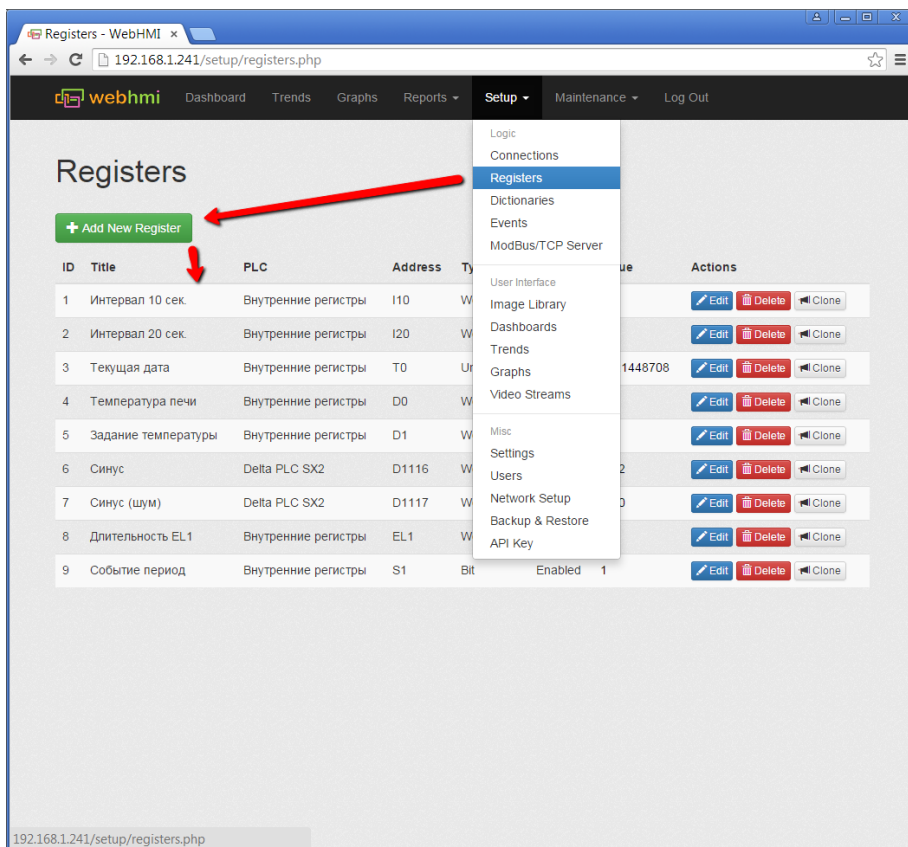
## Создание регистра, базовые настройки

Для создания регистра необходимо зайти в меню **Setup --> Registers** и нажать кнопку **Add new register**. После этого в появившейся вкладке Basic info указать:

**Title** Название регистра

**PLC** Соединение (ПЛК), через которое опрашиваются данные регистра

**Register address** Адрес в пространстве адресов данного соединения (ПЛК)



Для примера (не подключая никаких внешних устройств) можно использовать внутренний регистр интервалов *I10*, значение которого (остаток от деления текущего времени в сек. на 10) меняется с каждой сек. от 0 до 9. Присвоим ему имя "Интервал 10 сек.". Результат базовой настройки:

Registers - WebHMI x

192.168.1.241/setup/registers.php?edit=1

webhmi Dashboard Trends Graphs Reports Setup Maintenance Log Out

## Edit Register

Basic Info Value Log States

**Title**   
Enter human readable name for this image.

**PLC**   
Select PLC to read/write register from. You can manage PLCs [here](#).

**Register address**   
Enter address of register. Example: I0.0, X1, etc.

Disable

После нажатия кнопки Save , поскольку не был определен тип данных, система предложит указать тип на вкладке **Value**, для *I10* нужно указать Word и снова нажать **Save** - мы вернемся к списку всех регистров:

Registers - WebHMI x

192.168.1.241/setup/registers.php

webhmi Dashboard Trends Graphs Reports Setup Maintenance Log Out

## Registers

ID	Title	PLC	Address	Type	State	Value	Actions
1	Интервал 10 сек.	Внутренние регистры	I10	Word	Enabled	80	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
2	Интервал 20 сек.	Внутренние регистры	I20	Word	Enabled	8	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
3	Текущая дата	Внутренние регистры	T0	Unix Time	Enabled	1421449068	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
4	Температура печи	Внутренние регистры	D0	Word	Enabled	0	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
5	Задание температуры	Внутренние регистры	D1	Word	Enabled	0	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
6	Синус	Delta PLC SX2	D1116	Word	Enabled	374	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
7	Синус (шум)	Delta PLC SX2	D1117	Word	Enabled	293	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
8	Длительность EL1	Внутренние регистры	EL1	Word	Enabled	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>
9	Событие период	Внутренние регистры	S1	Bit	Enabled	1	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>

Формат списка регистров:

**ID** - уникальный номер регистра в проекте.

**Title, PLC, Address** - введенные ранее поля Название, содержание (ПЛК), Адрес.

**Type** - тип данных регистра

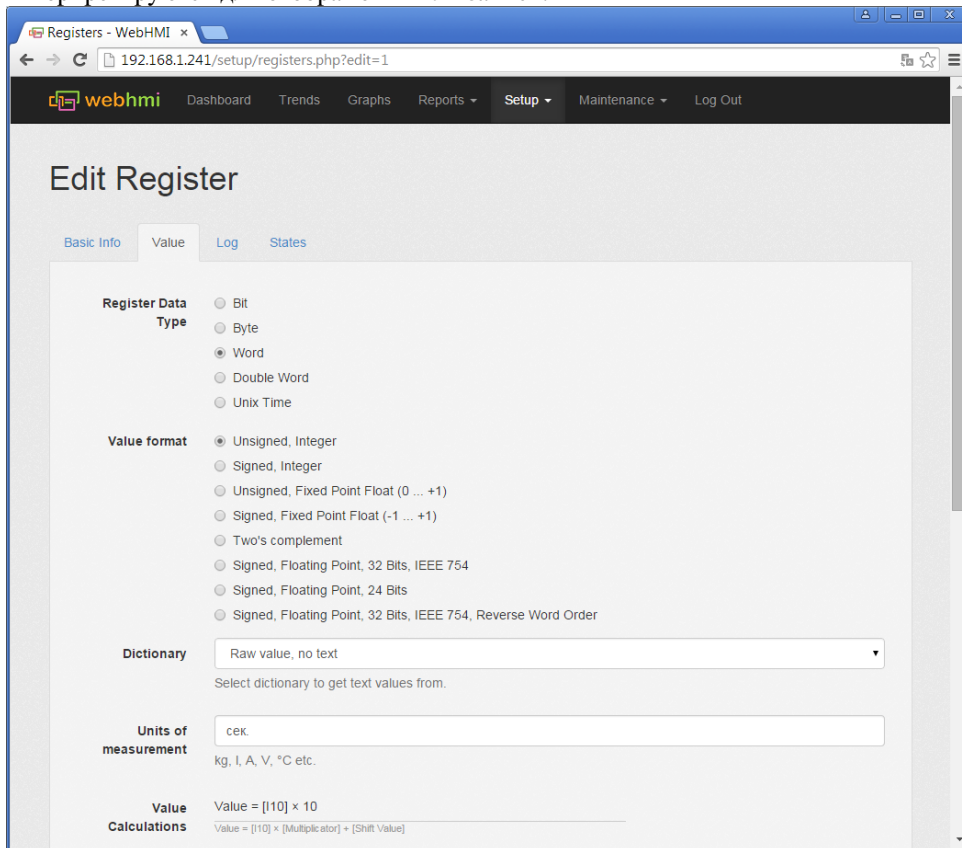
**State** - включен или отключен опрос этого регистра драйвером.

**Value** - отображение текущего состояния регистра, удобная функция для отладки - сразу же видно корректность настройки и наличие соединения.

Кнопки **Edit, Delete, Clone** - соответственно Редактирование, Удаление, Копирование регистра.

## Настройка типа и формата данных регистра

Вкладка **Value** меню редактирования регистра служит для настройки типа и формата данных регистра. Данные настройки влияют на то, как считанные двоичные данные из регистров подключенного устройства интерпретируются для отображения или записи.



Описание настроек на верхнем рисунке (начало):

### **Register Data Type** - Тип данных:

- Bit - Битовое значение,
- Byte - размер данных байт
- Word - размер данных слово
- Unix time - формат времени принятый в PHP, форматирование задается в виде Y-m-d H:i:s, например, для отображения даты в формате день - мес. - год (ДД-ММ-ГГГГ), необходимо указать d-m-Y.

### **Value format** - Задает формат считанным данным.

- Целое (integer)- со знаком или без знака (signed / unsigned)
- С фиксированной плавающей точкой (fixed point float) со знаком (диапазон -1..+1) и без знака (0..1) (signed / unsigned)
- В дополнительном коде (Two's complement)
- В формате с плавающей точкой со знаком (signed floating point), в стандарте IEEE754 - 24 бит(такой формат используется, например, в устройствах ОВЕН), 32 бит (ПЛК uLogix Allen-Bradley, Delta

Electronics и др. ), 32 бит с обратным порядком слов (Reverse Word Order)

**Dictionary** - Словарь. Содержит перечисление пар "значение, слово" , при выборе словаря для регистра считанные из устройства значения сопоставляются с данными словаря и при совпадении в регистре будет находится соответствующее значению слово. При несовпадении - регистр будет содержать само значение. Словари создаются и редактируются в меню **Setup --> Dictionary**

**Units of measurement** - указание инженерных единиц для данных регистра. Указанная в поле строка будет отображаться после текстовой строки со значением регистра на приборной панели.

Описание настроек на нижнем рисунке (Продолжение):

**Value Calculations** - Масштабирование со смещением данных в регистре. Используется для приведения шкал считываемых данных например из АЦП в инженерные единицы. Данные преобразуются по формуле:

$$\text{Итоговое значение} = [\text{Считанное в регистр значение}] \times \text{Множитель} + \text{Смещение}$$

**Multiplicator** - Множитель

**Shift Value** = Смещение

**Precision** = Количество знаков после запятой для значений с плав. точкой.

**Min/Max Allowed Value For Operator** - Данные поля позволяют ограничить мин. и макс. возможное значение для регистра, который может меняться пользователем с приборной панели.

**Min/Max Reasonable Value** - Данные поля позволяют обозначить зону недоверных значений регистра, таким образом можно судить о неисправности датчика. При чтении значения вне зоны, в регистр будет записан ноль, который в свою очередь можно определить как "отключенное состояние" - см. ниже по тексту.

The screenshot shows a web browser window titled "Registers - WebHMI" with the URL "192.168.1.241/setup/registers.php?edit=1". The interface contains the following fields and labels:

- Dictionary:** A dropdown menu set to "Raw value, no text". Below it, the text "Select dictionary to get text values from." is visible.
- Units of measurement:** A text input field containing "сек.". Below it, the text "Kg, l, A, V, °C etc." is visible.
- Value Calculations:** A section with the formula "Value = [10] x 10" and "Value = [10] x [Multiplicator] + [Shift Value]".
- Multiplicator:** A text input field containing "10".
- Shift Value:** An empty text input field.
- Precision:** An empty text input field. Below it, the text "How many digits you want to see after decimal point" is visible.
- Min Allowed Value For Operator:** An empty text input field. Below it, the text "Minimum value that is allowed to enter by operator" is visible.
- Max Allowed Value For Operator:** An empty text input field. Below it, the text "Maximum value that is allowed to enter by operator" is visible.
- Min Reasonable Value:** An empty text input field. Below it, the text "Minimum value that can be read from register. If register's value is less than this value system will think that read failed and will override value with 0." is visible.
- Max Reasonable Value:** An empty text input field. Below it, the text "Maximum value that can be read from register. If register's value is more than this value system will think that read failed and will override value with 0." is visible.

A blue "Save" button is located at the bottom of the form.

## Настройка записи регистра в лог и на графики

Регистры имеют настройки, позволяющие записывать их по определенным правилам в лог и на графики.

### Запись регистров в лог

После выбора опции "Enable Log" в настройках регистра, в пункте меню **Maintenance** появится новый пункт - "Registers Log" и появятся 4 настройки:

- **Change tolerance** - "Допуск" (в абсолютном значении или в %) на отклонение значения регистра от текущего, при превышении которого он запишется в лог. Если поле оставить пустым, запись произойдет при любом изменении. Частота записи при этом будет определяться настройками последующих пунктов.
- **Min log interval** - Служит для прореживания данных, ограничивает частоту записей на величину в сек., указанных в поле ввода.
- **Max log interval** - Если регистр не менял значение, либо изменения находились внутри "допуска", тогда эта настройка позволяет тем не менее записать регистр в лог по истечении указанного в поле ввода интервала в сек.
- **Max graph interval** - Данные лога могут быть источником для исторических графиков (Graphs) (см. настройки регистра для записи на график ниже), тогда если не было данных в течение указанного интервала, на графике будет соответствующий разрыв.

Настройка вывода информации в логе приведена в разделе Системная информация и логи

### Запись регистров на графики

После выбора опции "Save data for graphs" в настройках регистра, появится возможность указать частоту записи на графики и выбрать цвет отображения данных регистра на графиках. Появятся 2 дополнительные настройки:

- **Time interval** - Частота записи на графики, возможны варианты: 1 hour (1 час), 15,5,2,1 мин., Use data from log (использовать данные из лога).
- **Color for graphs** - Указывается цвет в формате RGB --> #00(R)00(G)00(B), например для выбора синего цвета - #0000ff. Либо можно выбрать из палитры справа от поля ввода.

Для сохранения изменений служит кнопка **Save**.

## Примеры

Допустим, имеется внутренний интервальный регистр *I10*, меняющий свои значения от 0 до 9 каждые 10 сек. Лог с настройками по умолчанию, т.е. когда установили опцию **Enable log** при незаполненных остальных настройках, сохранит все значения по нарастающему:

The screenshot shows the WebHMI interface with the 'Log' section. The date range is 25/02/2015 19:04 — 25/02/2015 20:04. The log shows 6 entries with the following data:

Date	I10
2015-02-25 20:03:39	9
2015-02-25 20:03:38	8
2015-02-25 20:03:37	7
2015-02-25 20:03:36	6
2015-02-25 20:03:35	5
2015-02-25 20:03:34	4

Если теперь установить значение **Tolerance = 2**, тогда лог станет выглядеть так:

The screenshot shows the WebHMI interface with the 'Log' section. The date range is 25/02/2015 00:00 — 25/02/2015 23:59. The log shows 7 entries with the following data:

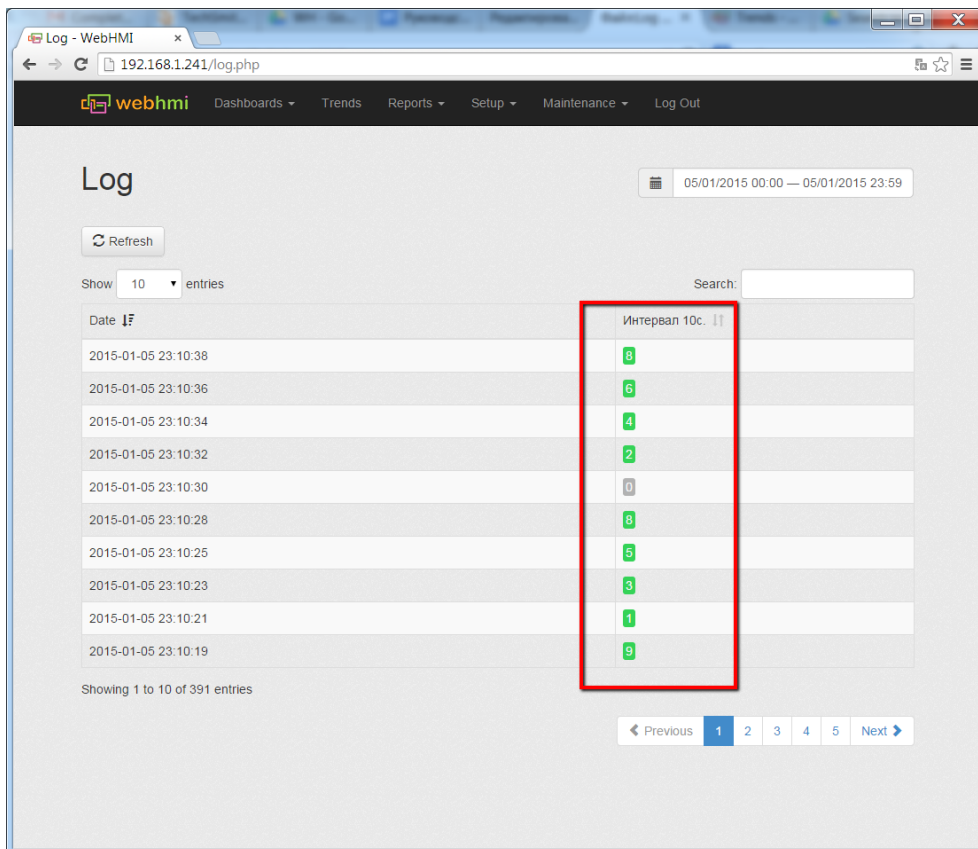
Date	I10
2015-02-25 20:10:38	8
2015-02-25 20:10:36	6
2015-02-25 20:10:34	4
2015-02-25 20:10:32	2
2015-02-25 20:10:30	0
2015-02-25 20:10:28	8
2015-02-25 20:10:26	6

Т.е.

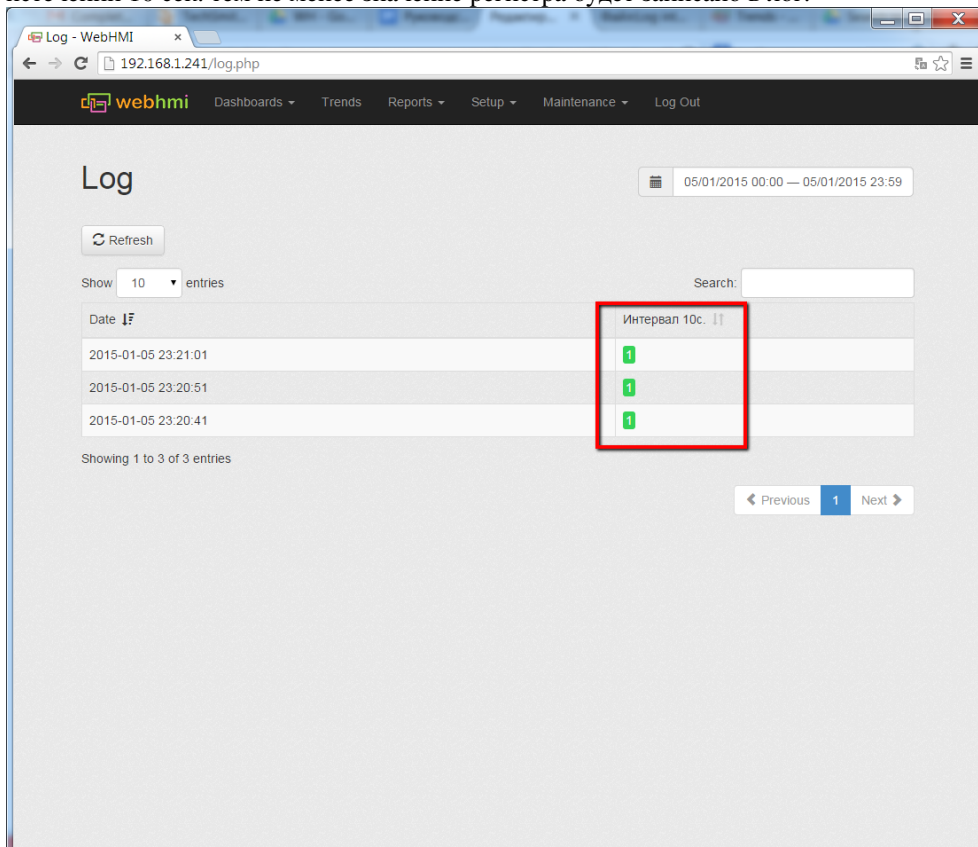
значения пишутся при изменении более чем на 2.

Проредить данные по частоте записи можно установив **Min log interval = 2**, записывая таким образом данные не чаще чем 2 сек. Результат изменений:



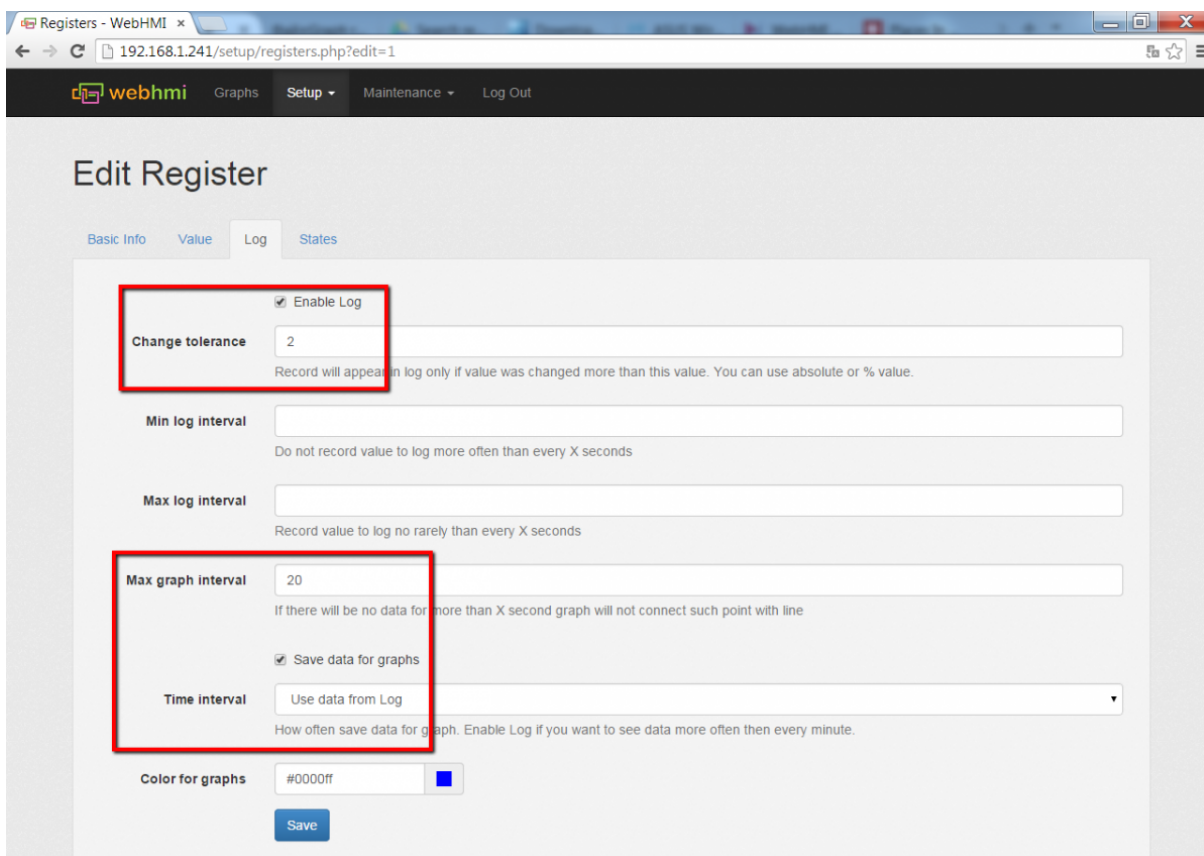


Чтобы проверить работу настройки **Max log interval**, можно установить **Tolerance** = 10 (тогда все значения будут "отбрасываться", так как они не в "допуске"), но благодаря установке **Max log interval** = 10, по истечении 10 сек. тем не менее значение регистра будет записано в лог.



Поскольку период изменения регистра *I10* равен как раз 10 сек., в логе находится всегда одно значение 1с.

Для примера настройки записи регистра *I10* на исторический график по данным лога, воспользуйтесь предыдущим примером с установкой **Tolerance** = 2. Настройки лога:

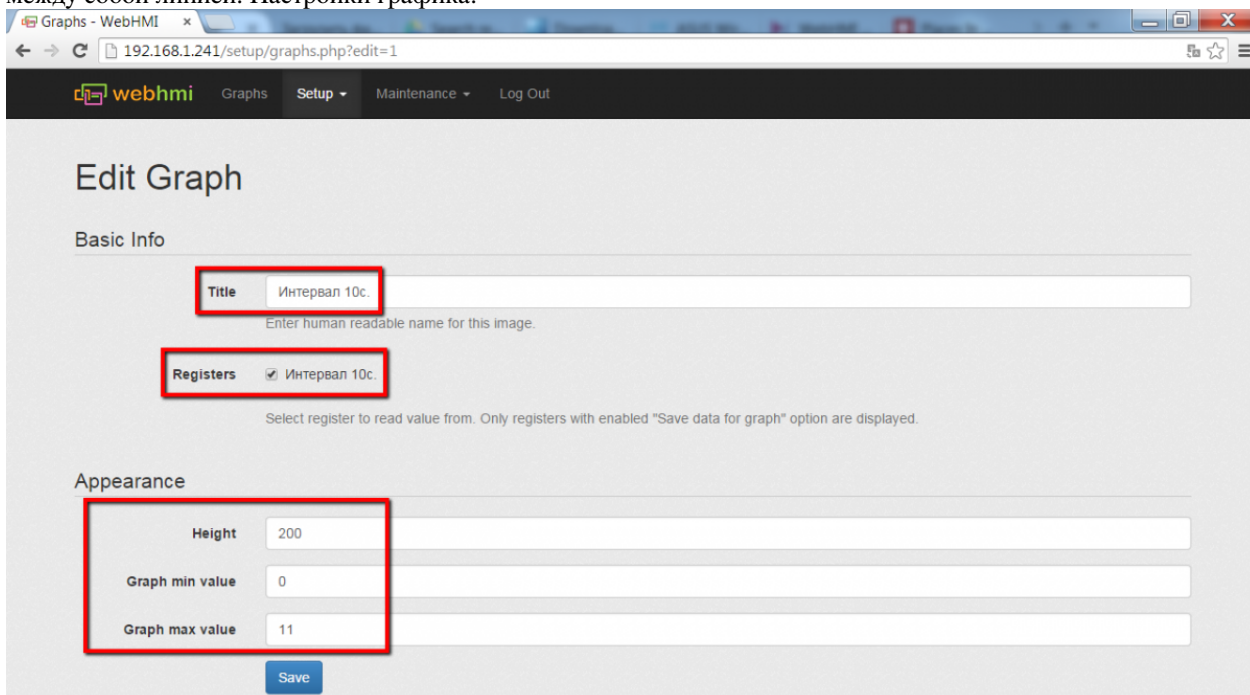


The screenshot shows the 'Edit Register' interface with the 'Log' tab selected. The following settings are highlighted with red boxes:

- Change tolerance:** 2
- Max graph interval:** 20
- Time interval:** Use data from Log

Other visible settings include 'Enable Log' (checked), 'Min log interval', 'Max log interval', and 'Color for graphs' (#0000ff).

Необходимо указывать значение **Max graph interval**, иначе на графике точки значений не будут соединены между собой линией. Настройки графика:

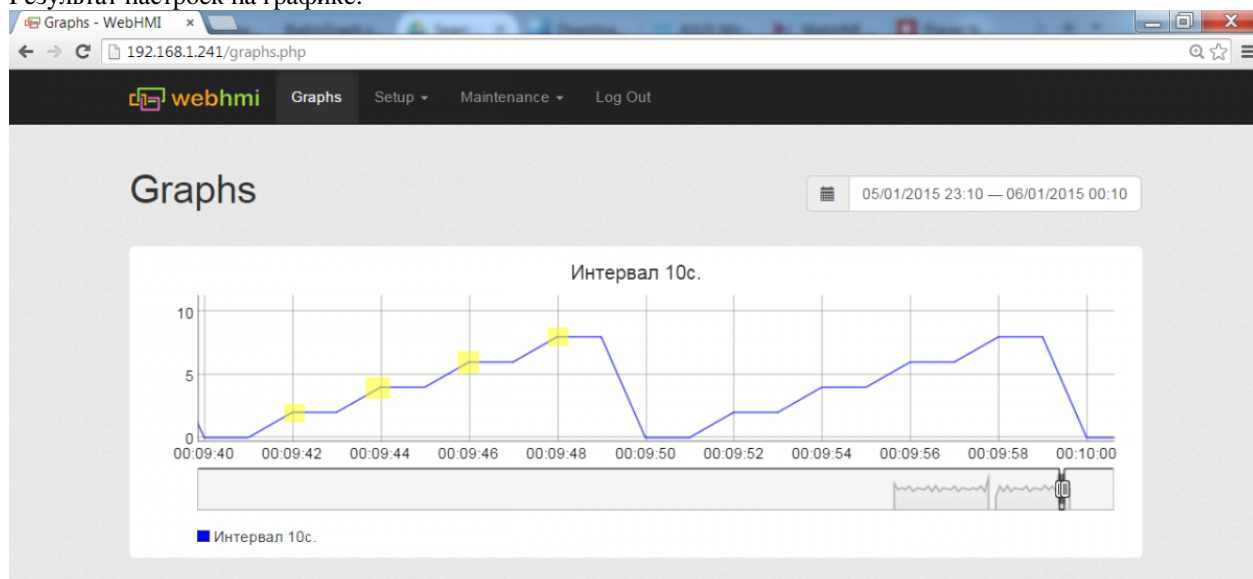


The screenshot shows the 'Edit Graph' interface with the 'Basic Info' section expanded. The following settings are highlighted with red boxes:

- Title:** Интервал 10с.
- Registers:** Интервал 10с.
- Appearance:**
  - Height:** 200
  - Graph min value:** 0
  - Graph max value:** 11

The 'Save' button is located at the bottom of the form.

Результат настроек на графике:

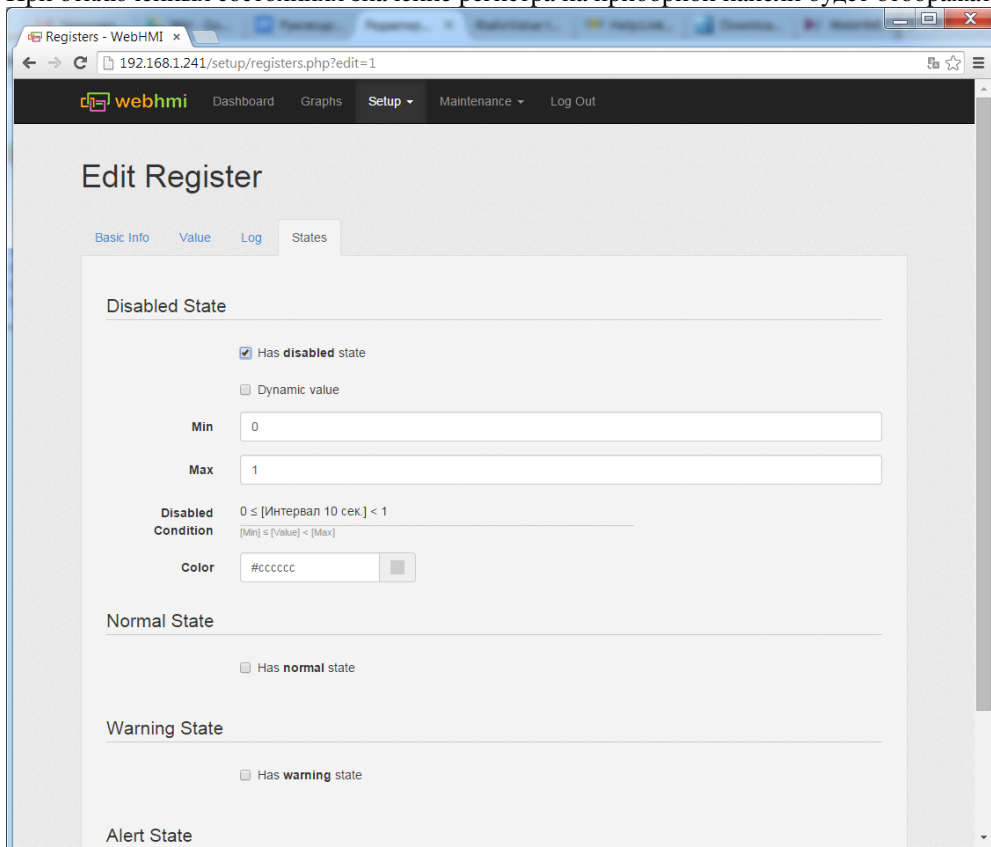


## Настройка состояний

Вкладка **States** меню редактирования регистров позволяет задавать 4 состояния цветовой сигнализации соответственно определенным значениям считанным из регистра:

- **Disabled state** - "отключен", данные в регистре говорят о том что механизм выключен, датчик отключен (или обрыв, например для сигнала 4..20мА меньше 4мА), нет рабочего давления и т.п. Цвет сигнализации по умолчанию - серый
- **Normal state** - "норма", значение соответствующее нормальной работе. Цвет сигнализации по умолчанию - зеленый
- **Warning state** - "предупреждение", параметр приближается к опасной границе. Цвет сигнализации по умолчанию - желтый
- **Alert state** - "авария" - аварийное состояние параметра. Цвет сигнализации по умолчанию - красный

При отключенных состояниях значение регистра на приборной панели будет отображаться черным цветом.

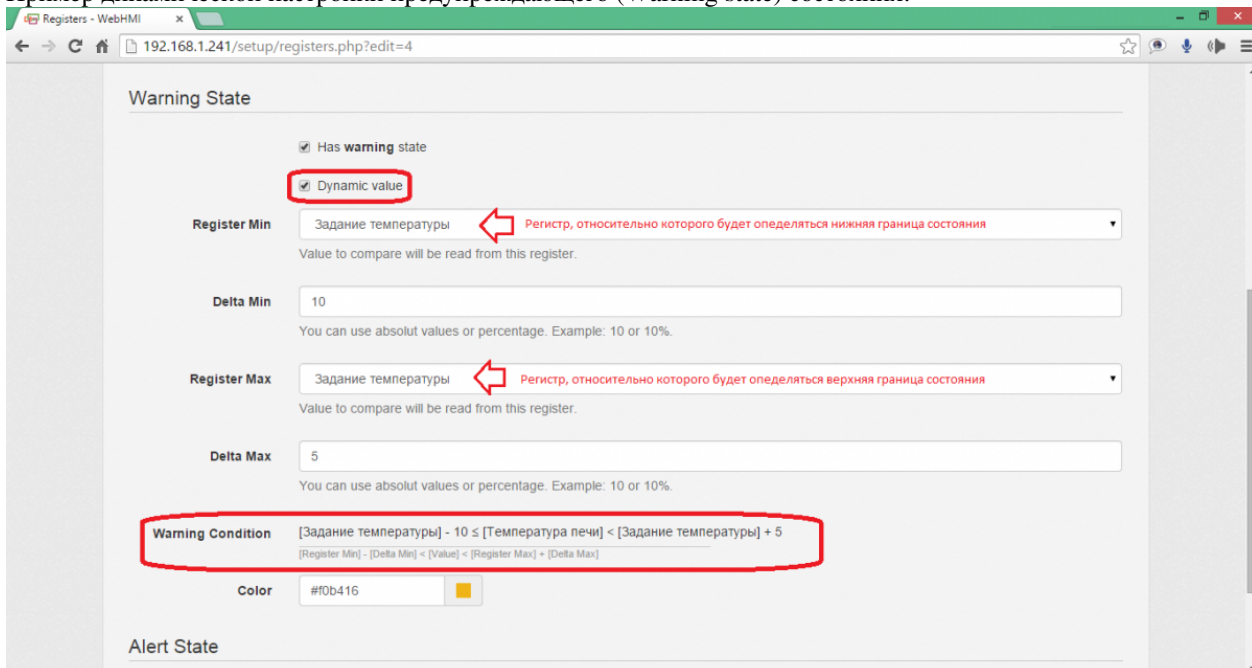


Настройка границ значений для каждого из состояний может быть **статической** и **динамической**.

При **статической** (по умолчанию) настройке интервал значений состояния регистра определяется по формуле:  $Min \leq (\text{значение регистра}) < Max$ . При этом  $Min$ ,  $Max$  являются константами

При **динамической** настройке  $Min$ ,  $Max$  также являются константами, но заданными относительно значений других регистров, и таким образом, например, в зависимости от рецептуры производимого продукта, будут меняться зоны предупреждений и аварийных состояний.

Пример динамической настройки предупреждающего (Warning state) состояния:



Здесь

**Delta Min** и **Max** - смещения от нижней и верхней границ зоны состояния предупреждения используемые в

формуле, приводимой для удобства здесь же в интерфейсе. Настройки состояний **Disable**, **Normal**, **Alert** одинаковы изображенному выше на рис. **Warning state**.

## СОБЫТИЯ

События представляют собой механизм, позволяющий регистрировать данные только по наступлению неких условий, когда началась определенная технологическая операция, либо например, определенный параметр процесса вышел за допустимые границы, и необходимо получить отдельный набор данных, связанный с этим моментом, таких как время начала, окончания, значения или изменения других параметров (влияющих на процесс или зависящих от него) в этот момент, либо на протяжении данного события, что позволяет эффективно анализировать такие ситуации.

Таким образом, события позволяют удобно отфильтровать необходимые данные по каждому из таких условий их наступления, и получить отдельные структурированные отчеты по каждому событию в отдельности. В отличие от построения необходимых протоколов по "срезу" информации в базе данных (логах), событийные отчеты формируются "на лету" т.е. в темпе с процессом мониторинга. После создания события, его данные становятся доступными в виде отчетов в меню **Reports**, а также имеется возможность представлять данные на временной оси ("ленте"), доступной в меню **Reports/Timeline** (после выбора опции Show on timeline для одного из событий).

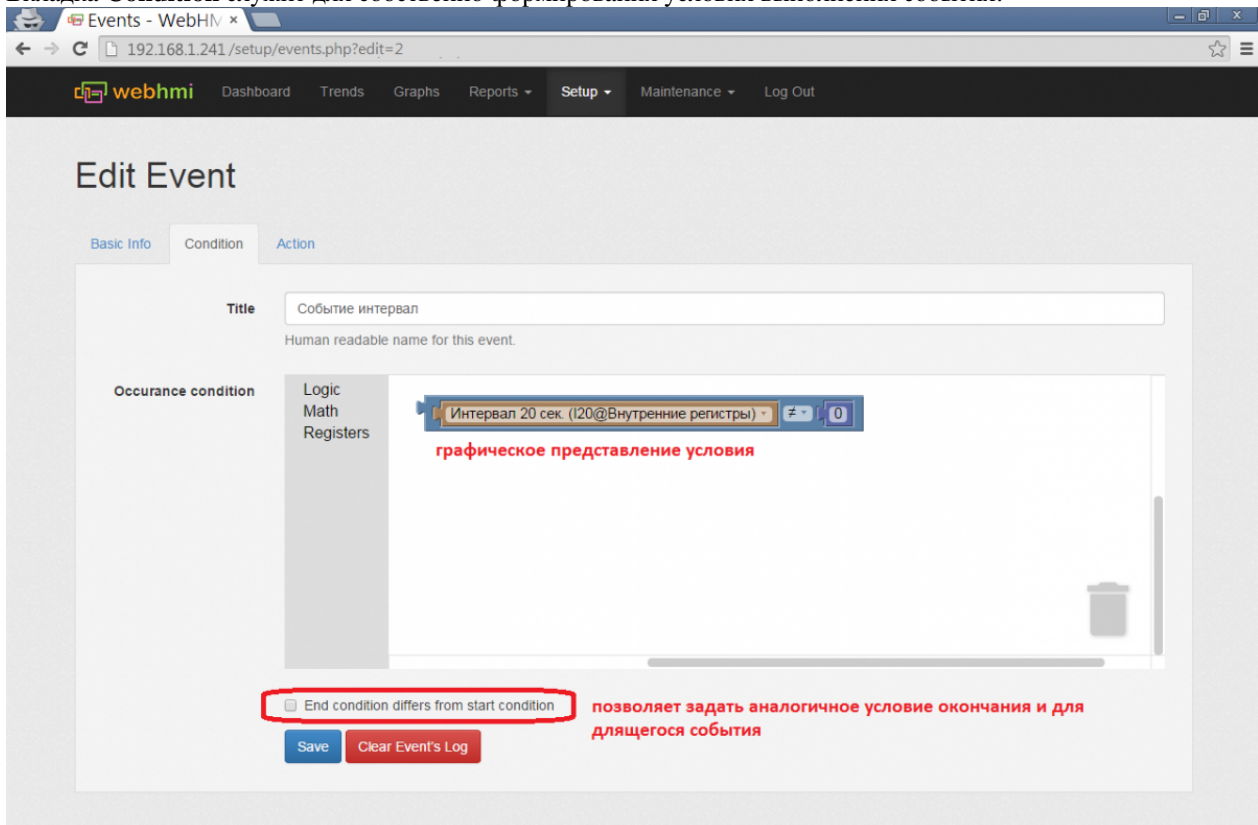
Рассмотрим в качестве примера работы событийного отчета задачу определения интервала времени, когда некоторый параметр вышел за определенные границы, и дополнительные данные связанные с этим событием - время начала и окончания этого интервала, среднее значение параметра в этом интервале, минимальное и максимальное. В качестве источника сигнала будем использовать данные регистра "интервал 20 сек." (I20), меняющего свои значения от 0 до 19 в течении 20 сек.

Для создание событий выбираем п. меню **Setup/Events**, затем **+Add new event**. Назначение базовых настроек показано на рисунке. Поскольку событие длящееся, необходимо не забыть указать соответствующую опцию.

The screenshot shows the 'Edit Event' form in the webhmi interface. The form is divided into three tabs: 'Basic Info', 'Condition', and 'Action'. The 'Basic Info' tab is selected. It contains the following fields and annotations:

- Title:** A text input field containing 'Событие интервал'. A red annotation 'название события' points to this field.
- Description:** A text area containing 'Краткое описание - интервал 20 сек. из внутр. регистра I20'. A red annotation 'краткое описание' points to this field.
- Has some duration in time:** A checked checkbox. A red annotation 'указать для длящегося события' points to it.
- Parent Event:** A dropdown menu showing '--'. A red arrow points to the dropdown, and a red annotation 'при выборе события-родителя, данное событие будет выполняться только когда выполняется родительское' points to the dropdown.
- Buttons:** 'Save' and 'Clear Event's Log' buttons are located at the bottom.

Вкладка **Condition** служит для собственно формирования условия выполнения события:



Визуальный редактор позволяет "склеивать" логические выражения в цепочки условий, в т.ч. с несколькими уровнями вложенности. Доступны следующие блоки:

**Logic** - логические операторы "НЕ", "И", "ИЛИ", и сравнения - "равенство", "неравенство", "больше", "меньше", "больше или равно", "меньше или равно";

**Math** - арифметические блоки - константа, либо операция сложения, вычитания, умножение, деление;

**Registers** - блоки выбора регистра для включения в логические или арифметические выражения;

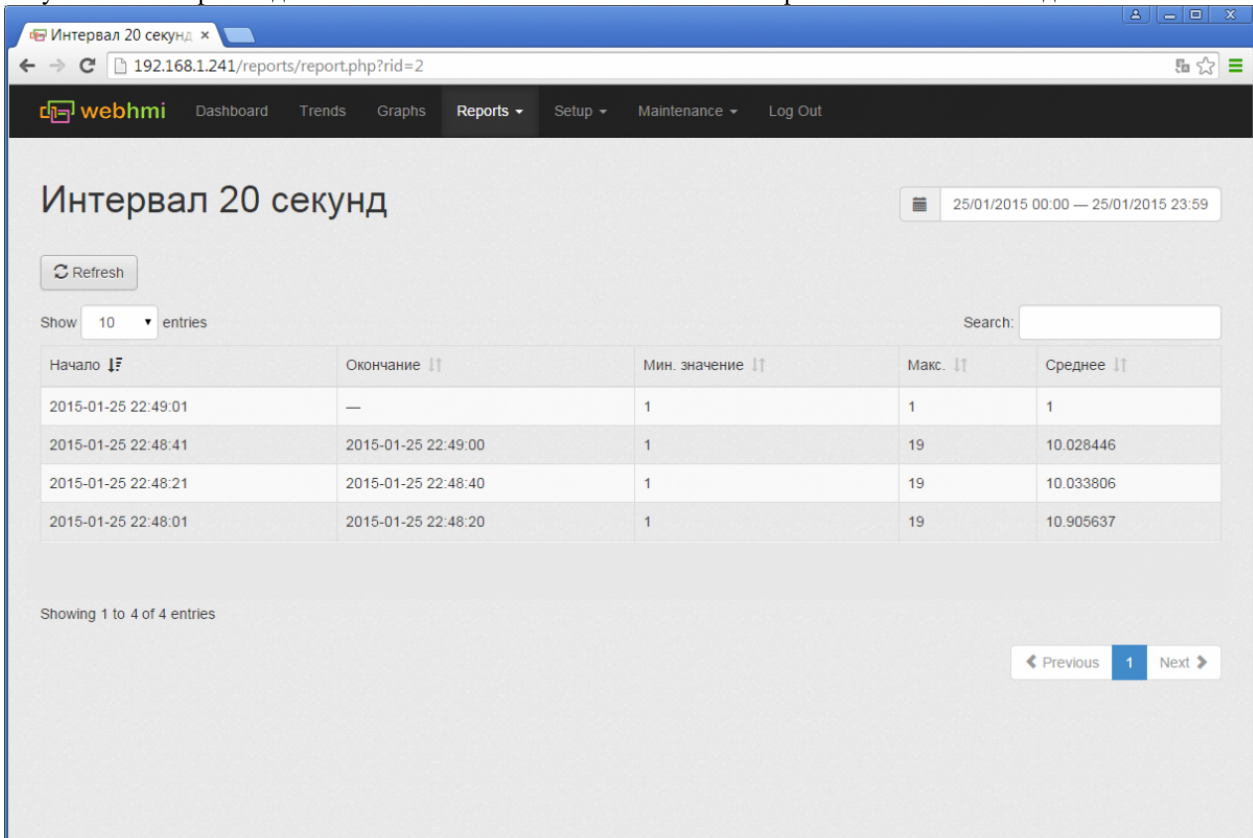
Вкладка **Action** настройки событий определяет что, как, и когда записывается в структуру данных отчета событий.

При выборе опции "Save to log" появится возможность задания регистров условия вида E<sub>y</sub>.x. (полное описание см здесь). Регистры условий нумеруются от 0 и далее по возрастанию (x - это индекс), y - идентификатор события, который присваивается ему при создании и отображается в списке событий. В качестве параметра записываемого в регистр условий может быть 4 варианта:

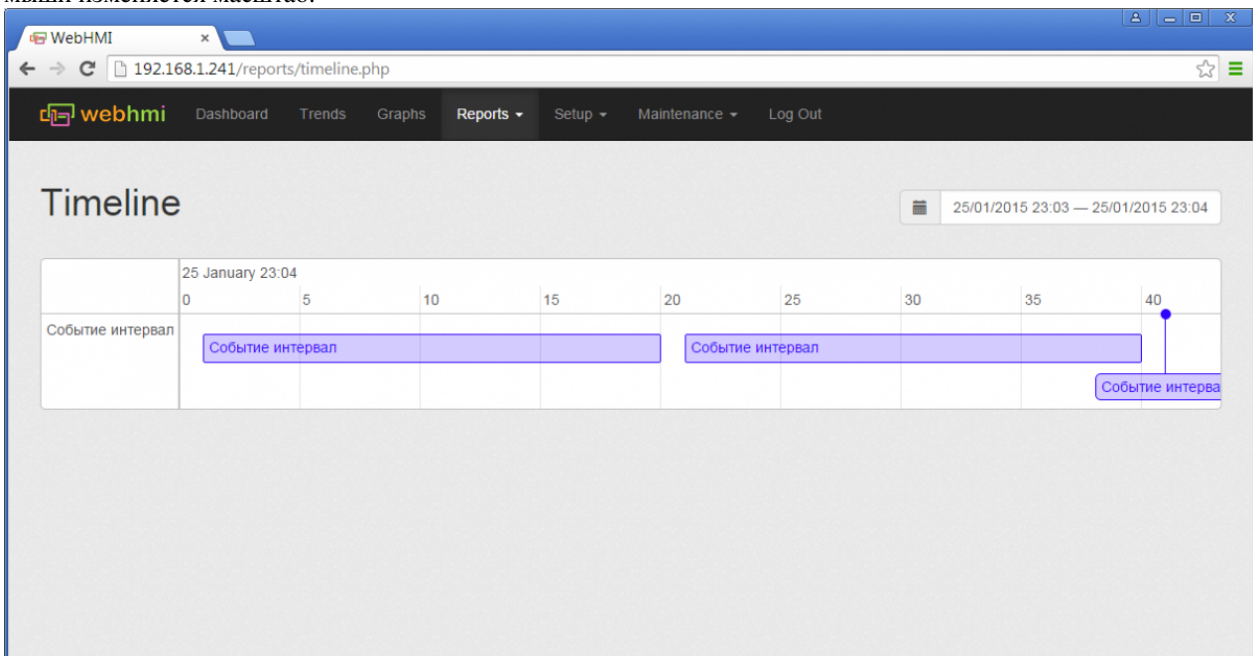
- Start time - время начала события
- End time - время окончания события
- Last update time - время последнего обновления, фактической регистрации события
- Произвольный регистр из списка объявленных в меню Registers - при этом в третьей колонке настроек отчета можно будет дополнительно выбрать обработку этого регистра - взять минимальное (Min), максимальное значение (Max), первое значение при старте события (First Value) или последнее при его окончании (Last Value), среднее значение (Average Value).

Настройки для регистров условий в и используемом нами примере приведены на предыдущем рисунке. Название столбцов отчета могут изменены. Для регистров E2.0, E2.1, содержащих время, используется строка задания формата, принятая в PHP. Для регистров E2.2, E2.3 заданы мин и макс. значения соответственно. E2.4 - усредненное значение регистра на протяжении события.

Результатом настройки данного события станет появление в меню Reports отчетов такого вида:



Поскольку при создании события была также указана опция "Show on timeline", все события будут сохранены и также и на ленте. Выноской с точкой на оси отмечается начало события, либо однократное событие. Удерживая нажатой основную кнопку мыши, можно сдвинуть график по горизонтали, "прокруткой" колеса мыши изменяется масштаб.





# Отображение данных

## Приборные панели

### Создание приборной панели

В WEBHMI приборные панели (Dashboards) являются аналогом экранов для операторских панелей или мнемосхем в SCADA системах. Можно создавать несколько приборных панелей, которые будут доступны после создания в меню **Dashboard**. Для примера создадим приборную панель "Приборная панель 1", используя п. меню **Setup/Dashboards**.

Dashboard - WebHMI x  
192.168.1.43/setup/dashboards.php?edit=1

webhmi Dashboard Trends Graphs Reports Setup Maintenance Log Out

### Edit Dashboard

Basic Info

**Название** Title   
Enter human readable name for this dashboard.

Hide title **Скрывать название**

**Ширина** Width

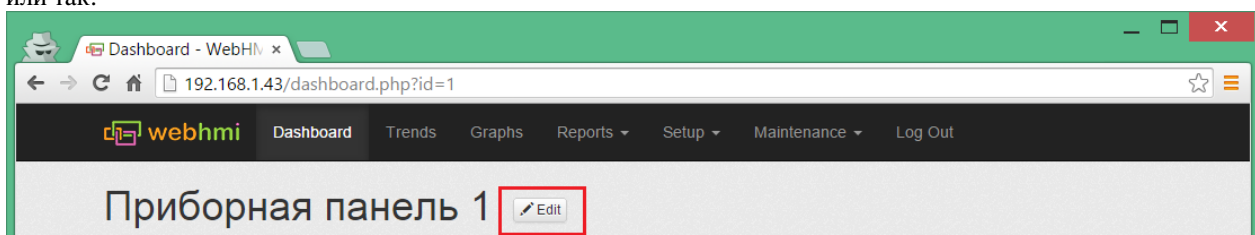
**Высота** Height

Save

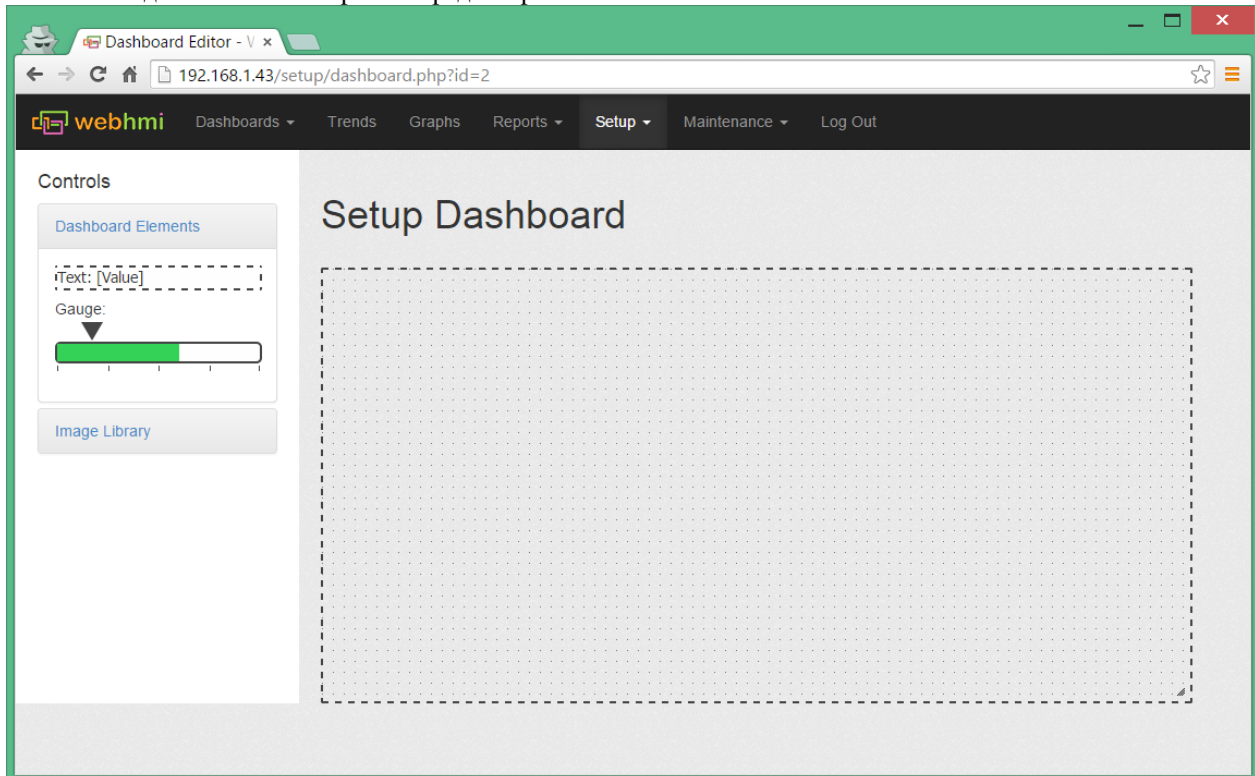
Перейти к редактированию панели можно двумя способами: нажав кнопку "Edit" в списке приборных панелей, либо нажав такую же кнопку, находясь на ее странице.

Id	Title	Width	Height	Actions
1	Приборная панель 1	800	400	<input type="button" value="Visual Editor"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Clone"/>

или так:



Внешний вид новой панели в режиме редактирования:



После

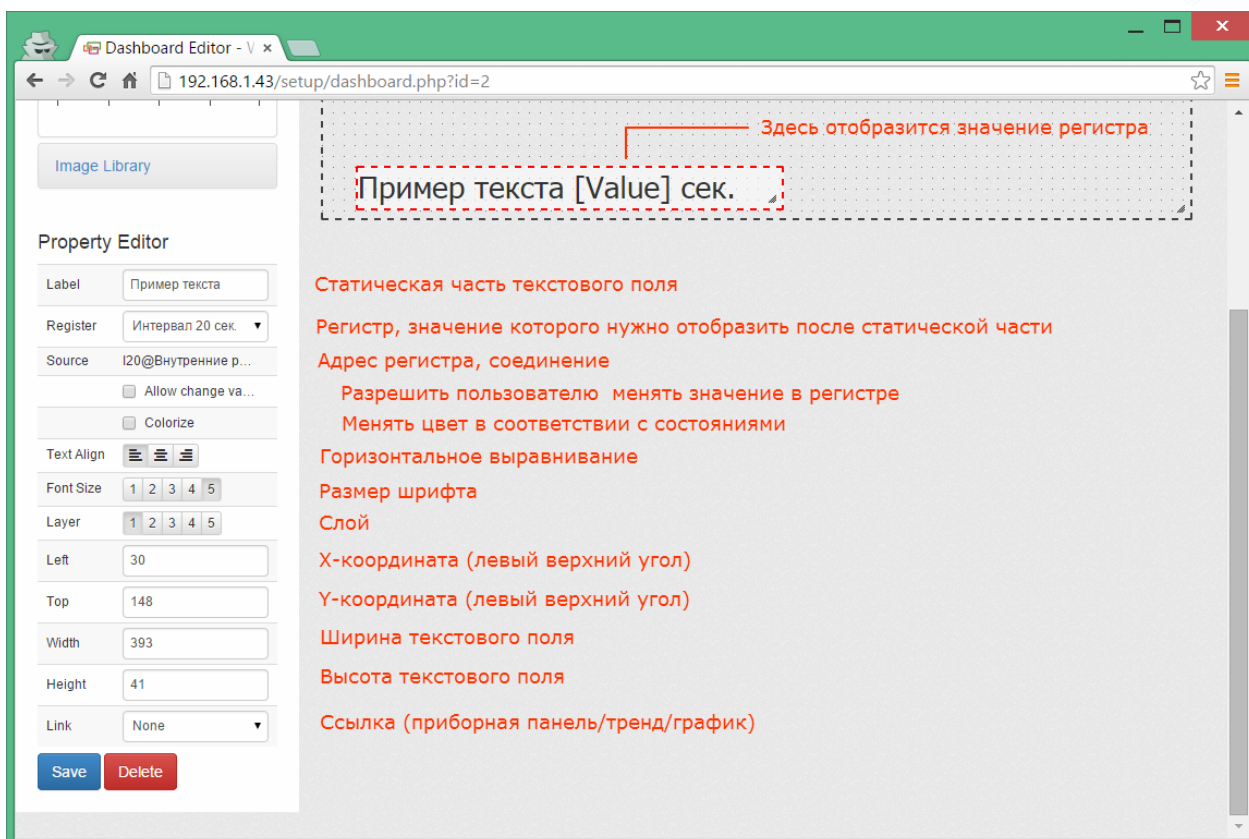
этого можно приступить к компоновке элементов визуализации на приборной панели. Необходимые элементы помещаются на поле их "перетаскиванием" с левой дополнительной панели. Доступны четыре типа элементов визуализации:

- текст
- датчики (столбиковый индикатор)
- изображения
- индикаторы

Поле редактирования имеет сетку с шагом 10 пикселей, по которой будут выравниваться элементы. При необходимости выровнять объект более точно, это можно сделать удерживая *Shift*.

## Элементы текста

Элементы текста позволяют разместить на поле приборной панели статический текст, либо значение регистра, либо составляющую их строку. Если в свойствах регистра были определены единицы измерения, то они также отобразятся после значения регистра. Установка атрибута "Allow change value" превращает текст в поле для ввода значения в регистр. Элемент текста также может наследовать цветовую сигнализацию состояния регистра. Имеется 5 вариантов размера шрифта (в относительных единицах) и 5 слоев (слой с большим номером находится ближе к пользователю смотрящему на экран). Текст может также быть удобной ссылкой на другую приборную панель, тренд или график. См. рис.:

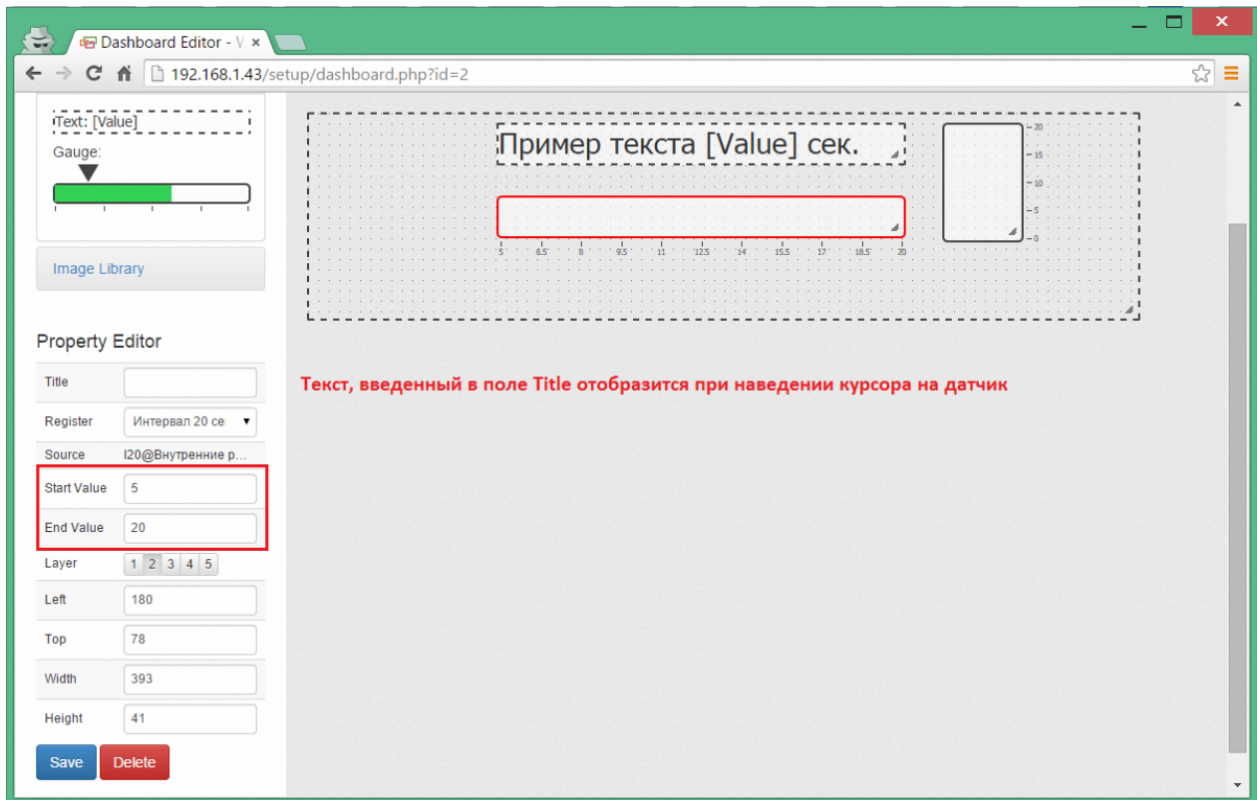


## Датчики

Значения отображаемые этими элементами представляются в виде столбиковых индикаторов, являющихся удобным графическим представлением величины параметра, позволяющим быстро оценить текущее значение оператору. Настройки датчика аналогичны настройкам элементам текста, за исключением двух полей:

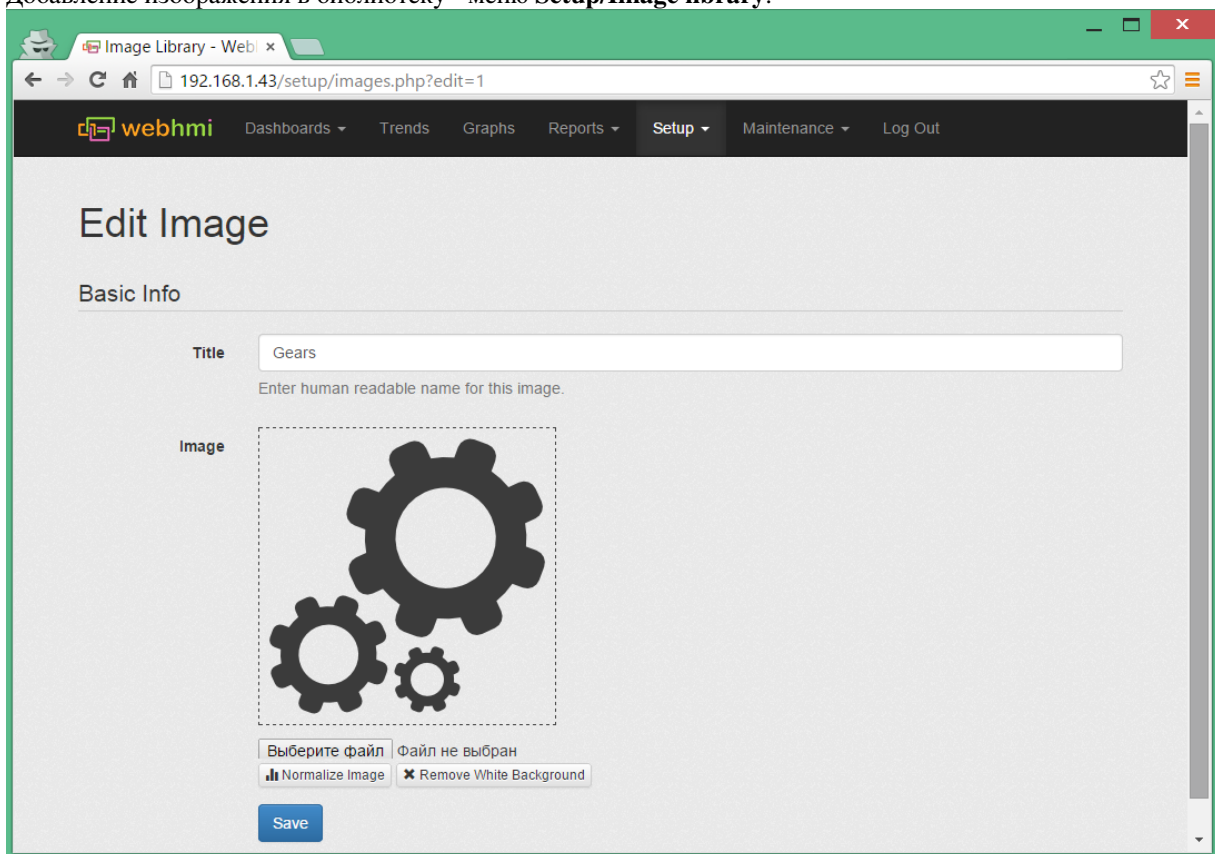
- Start value - начальное значение шкалы, с которой начнется движение стрелки указателя.
- End value - соответственно конечное значение шкалы.

Датчик не отобразит значений, находящихся вне этого диапазона. Ширину и длину, горизонтальную либо вертикальную ориентацию индикатору можно придать, потянув за правый нижний угол. Более точно задать размеры и положение, можно используя непосредственный ввод значений в поля координат элементов. Например, текст и нижний датчик выровнены путем ввода одного и того же значения в поле *Left*, и т.д.



## Изображения

На приборную панель может быть помещено изображение, содержащееся в библиотеке изображений. Добавление изображения в библиотеку - меню **Setup/Image library**.



полей:

Назначение

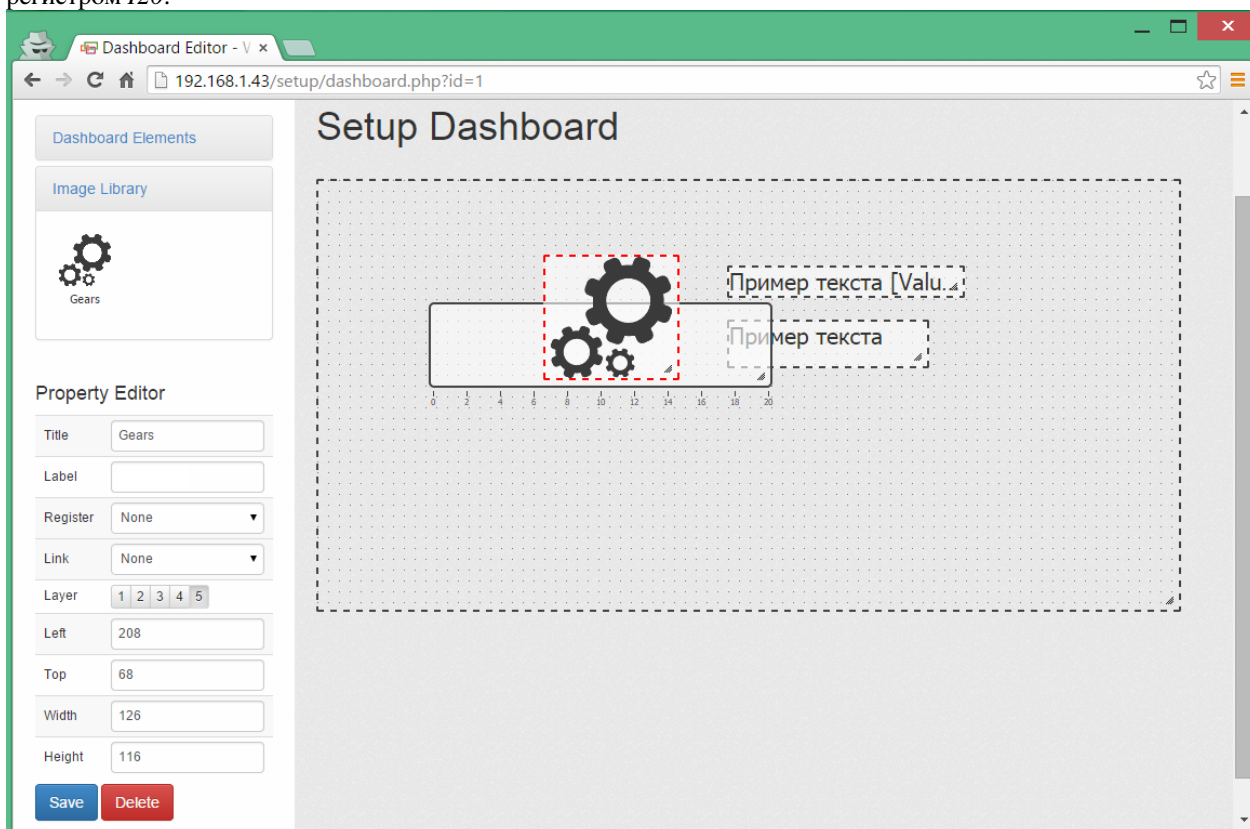
**Title** - название которое будет отображаться при наведении курсора на изображении на странице приборной панели.

**Image** - область предварительного просмотра изображения

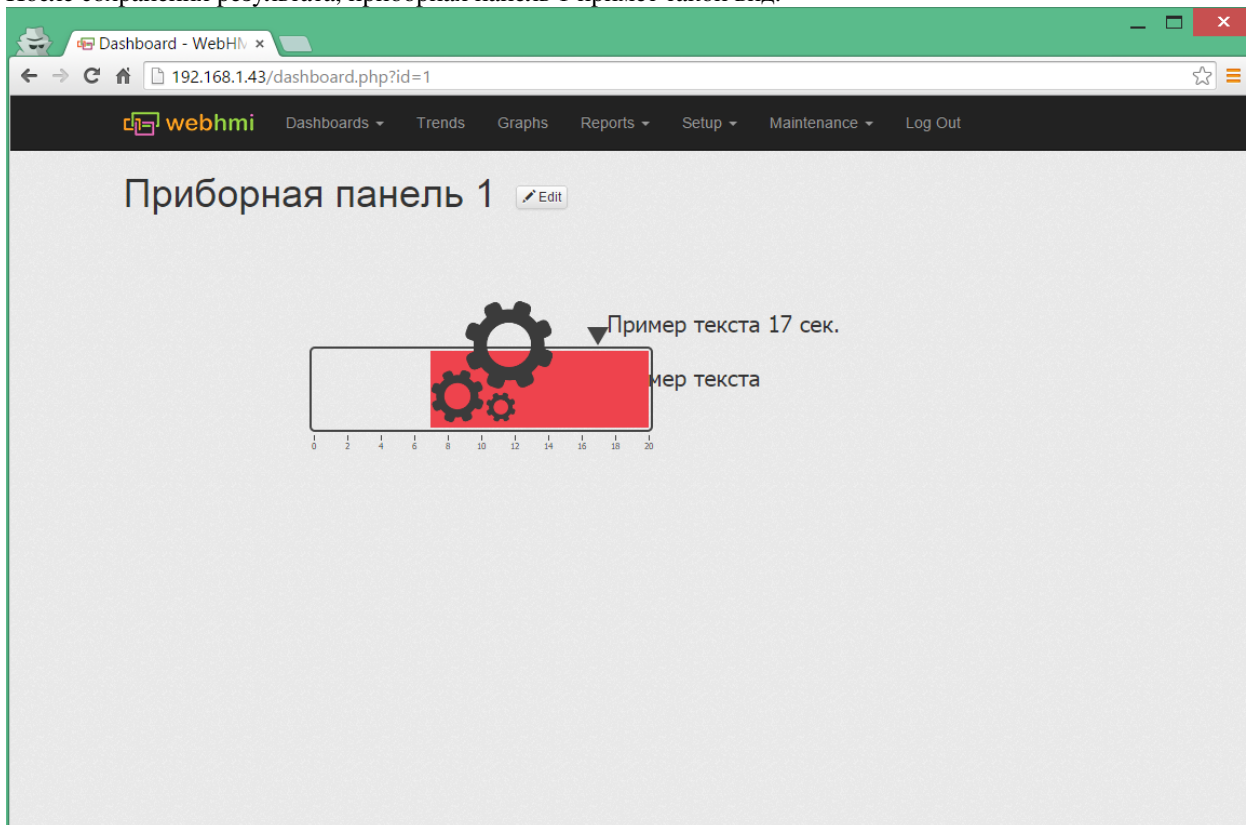
**Normalize image** - суть нормализации состоит в том, чтоб привести изображения индикаторов к одному стандартному виду, при этом они обесцвечиваются и выравниваются по яркости

**Remove white background** - целесообразно использовать изображения с прозрачным фоном. в случае если в изображении используется белый цвет как фон, данная операция позволяет заменить его на прозрачный.

Для примера возьмем произвольное изображение с прозрачным фоном (в формате \*.png) и используя возможности слоев, вынесем его на самый верхний, пятый слой, расположив над датчиком, связанным с регистром *I20*.



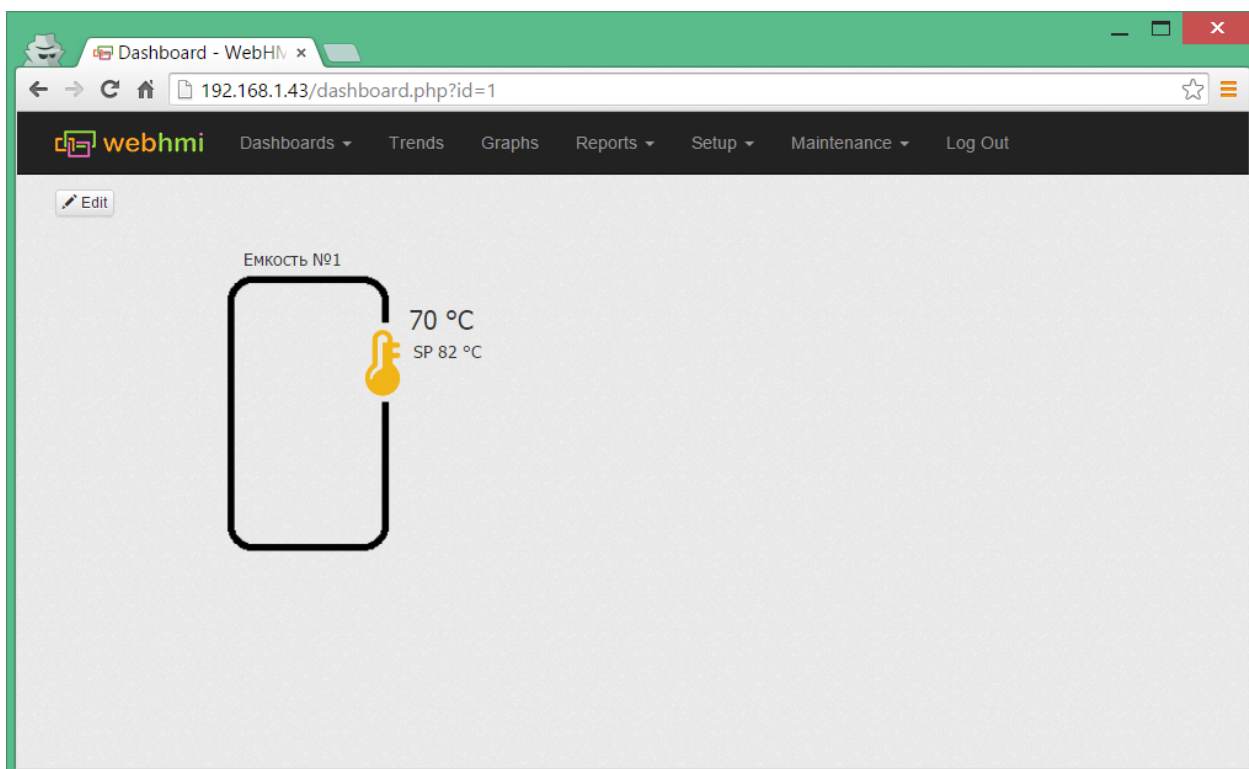
После сохранения результата, приборная панель 1 примет такой вид:



## Индикаторы

Индикатор - это вид изображения, которое предназначено для индикации состояния связанного с ним параметра (регистра). Для того, чтобы все состояния выглядели одинаково во всем проекте, желательно использовать цвета по умолчанию, которые предлагаются при описании состояний (хотя их можно и переопределить). Также желательно производить нормализацию изображений для индикаторов (см. описание опции в разделе "Изображения").

Еще одним применением индикатора может быть создание кнопки с фиксацией, т.е. если данные связанного с ним регистра битовые, тогда "клик" на таком индикаторе вызовет инверсию текущего состояния, т.е. действие будет эквивалентно работе такой кнопки (соответственно можно назначить состояние Disabled для 0, и Normal для 1). Пример использования индикатора приведен ниже. Индикатором в данном случае является пиктограмма термометра (см. раздел изображения), привязанная регистру, имеющему состояние предупреждения (желтый цвет). Также в примере используется изображение прямоугольника на прозрачном фоне для обозначения емкости, и три текстовых элемента, привязанных к регистрам текущей и заданной температуры, и подпись для емкости.



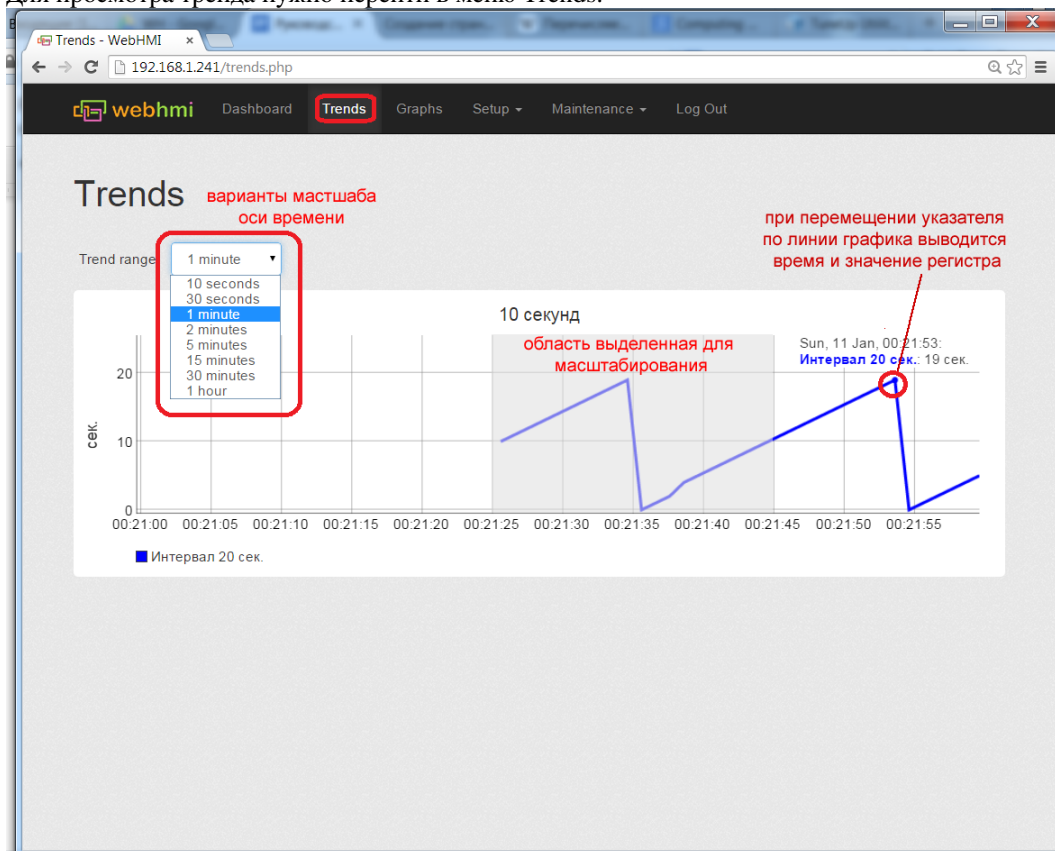
Изображение становится индикатором после привязки регистра с цветовой сигнализацией. Все настройки точно такие же, как и для изображений.

## Тренды

Тренды (графики текущих значение) представляют собой удобный инструмент анализа процесса в реальном времени. Для создания графика необходимо перейти в меню **Setup --> Trends --> Add New Trend**. Создадим тренд с использованием регистра *Интервал 20 сек*(Внутренний регистр меняющий значение 0..19 сек.).

The screenshot shows the 'Edit Trend' configuration page in the webhmi interface. The browser address bar shows '192.168.1.241/setup/trends.php?edit=1'. The page has a navigation menu with 'webhmi', 'Dashboard', 'Trends', 'Graphs', 'Setup', 'Maintenance', and 'Log Out'. The main content area is titled 'Edit Trend' and contains a 'Basic Info' section with a 'Title' field set to '10 секунд' and a red annotation 'Название тренда'. Below this is a 'Registers' section with radio buttons for 'Задание температуры', 'Интервал 10 сек.', 'Интервал 20 сек.' (which is selected), and 'Температура печи'. A red arrow points to the 'Интервал 20 сек.' option with the annotation 'Выбор регистра для вывода в тренде'. The 'Appearance' section has three input fields: 'Height' (200, 'Высота графика в пикселах'), 'Graph min value' (0, 'Мин. значение на оси'), and 'Graph max value' (25, 'Макс. значение на оси'). A 'Save' button is at the bottom.

Для просмотра тренда нужно перейти в меню Trends.





# Исторические графики

Исторические графики (далее по тексту просто графики), в отличие от трендов, позволяют сохранять свои данные в архив, глубина которого настраивается в системных настройках **Setup/Settings**, настройка **Keep data in log for** (макс. срок 5 лет). Теоретически объем архива ограничен только размером флеш-карты.

На практике при увеличении размеров архива свыше 300Мб, "время отклика" в интерфейсе при работе с WEBHMI может увеличиться и быть заметным для комфортной работы пользователя. Следует сказать, что хранение больших объемов данных не является целевым применением устройства, хотя оно и позволяет это делать. Данные графики целесообразно использовать на этапе исследования объекта мониторинга, держать архив для анализа определенных ситуаций (который после использования может быть удален) и т.п.

Для длительного хранения большого объема данных мониторинга рекомендуется использовать специальный сервер - через API либо опцию "облачного" сервера.

Для создания графика необходимо выбрать п. меню **Setup --> Graphs**. Интерфейс создания здесь такой же как и у трендов, за исключением того, что в списке регистров будут отображены только те регистры, у которых установлена опция "Save data for graphs"(см. работа с регистрами). Также под осью времени добавлены 2 "ползунка" ("слайдера") для изменения масштаба в одну из сторон.

Пример работы графика.



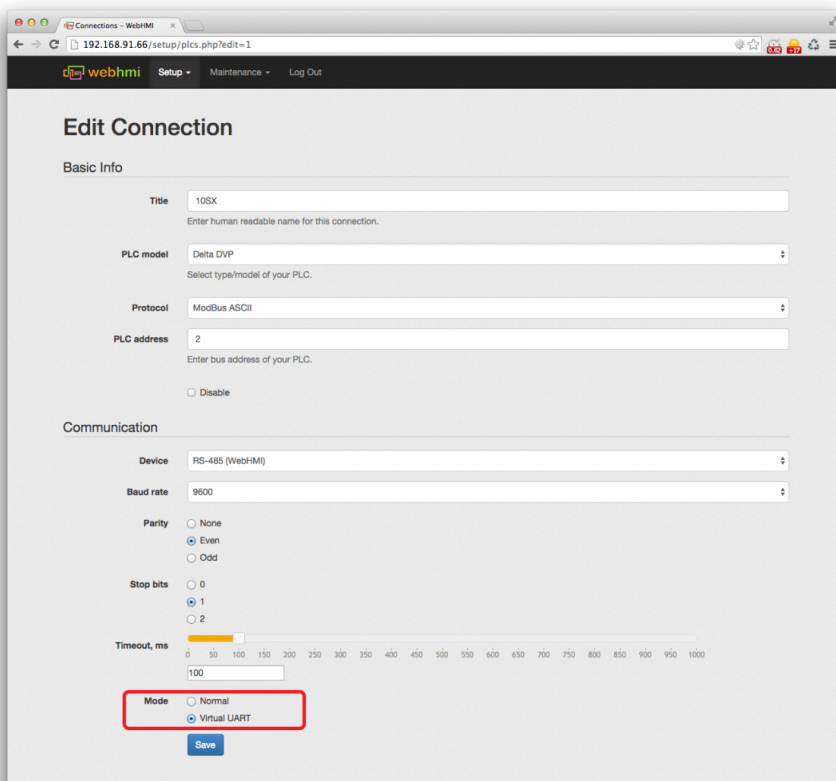
# Дополнительные функции

## Настройка виртуального UART

Иногда возникает необходимость получить доступ к удаленному ПЛК или другому оборудованию с интерфейсом RS-485 или RS-232. Для решения этой задачи можно использовать функцию Virtual UART которая встроена в WebHMI.

Предположим, что нам необходимо обновить прошивку в контролере Delta DVP-10SX. Он подключен к WebHMI с помощью RS-485.

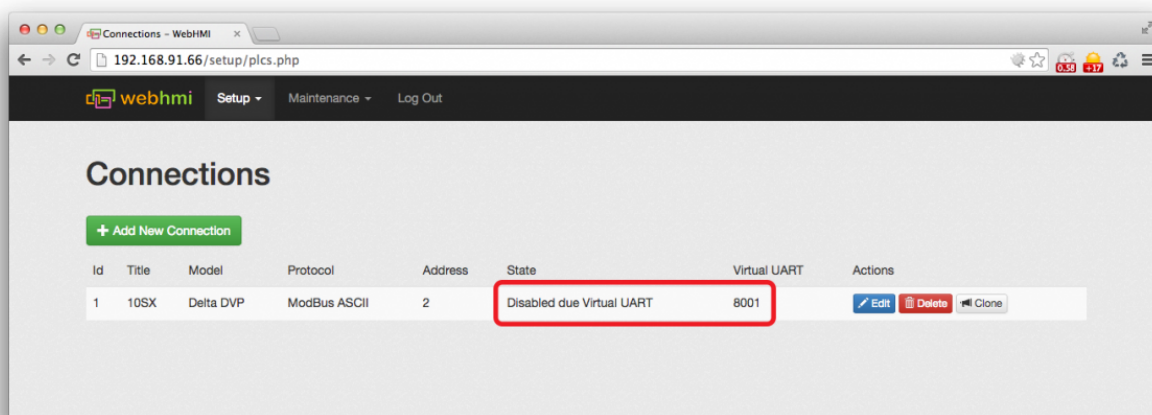
1. Откройте настройки соединения и измените Mode с Normal на Virtual UART.



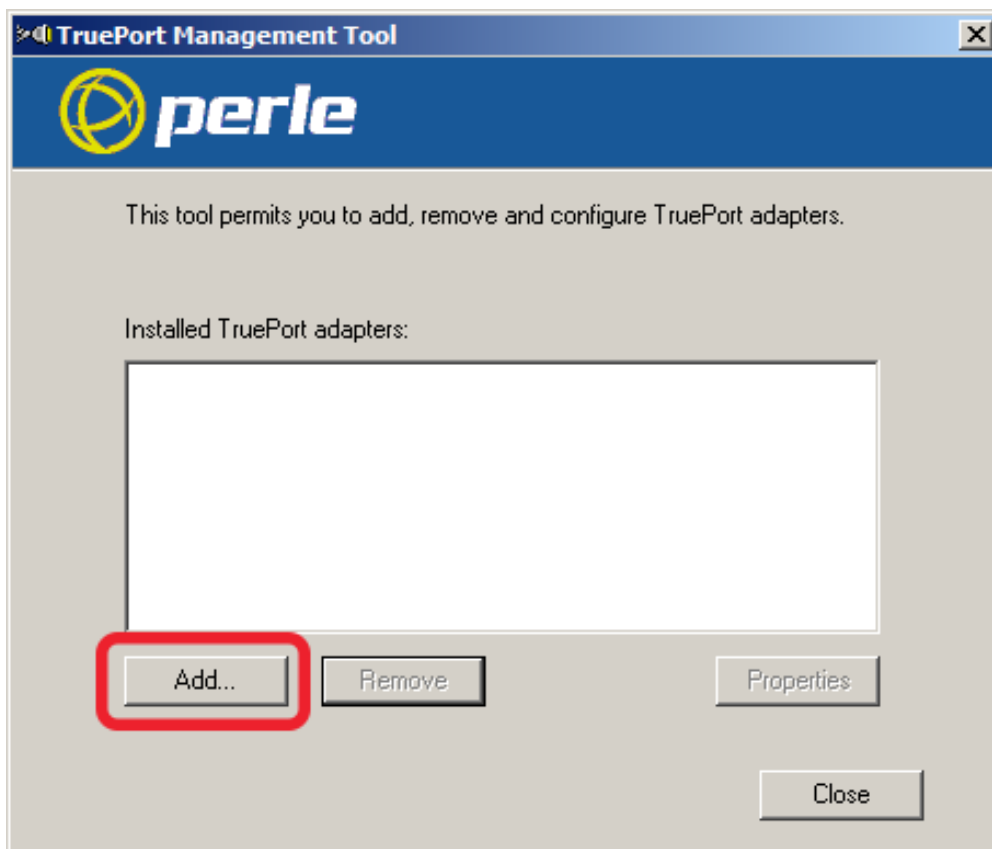
2. Сохраните изменения.

После сохранения вы увидите что указанное соединение отключено из-за Virtual UART. Это значит что WebHMI не будет работать с ним пока включен режим Virtual UART. Если этот UART используется и для других ПЛК то они также будут отключены и WebHMI не сможет работать с ними.

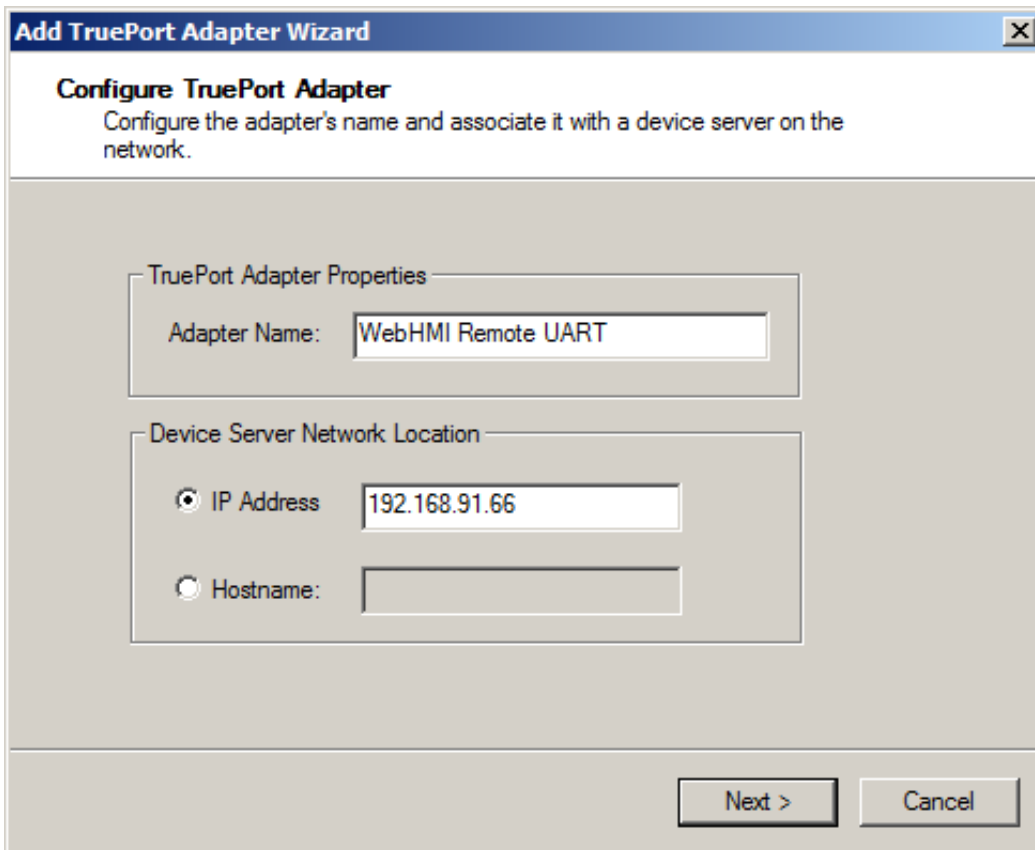
Также в списке соединений вы увидите колонку Virtual UART. В ней указан номер TCP-порта данного UART по которому следует производить подключение.



3. Загрузите и установите драйвер [Perle TruePort <sup>[1]</sup>] для вашей операционной системы.
4. Запустите TruePort Management Tool. Нажмите "Add..." для добавления нового виртуального порта.

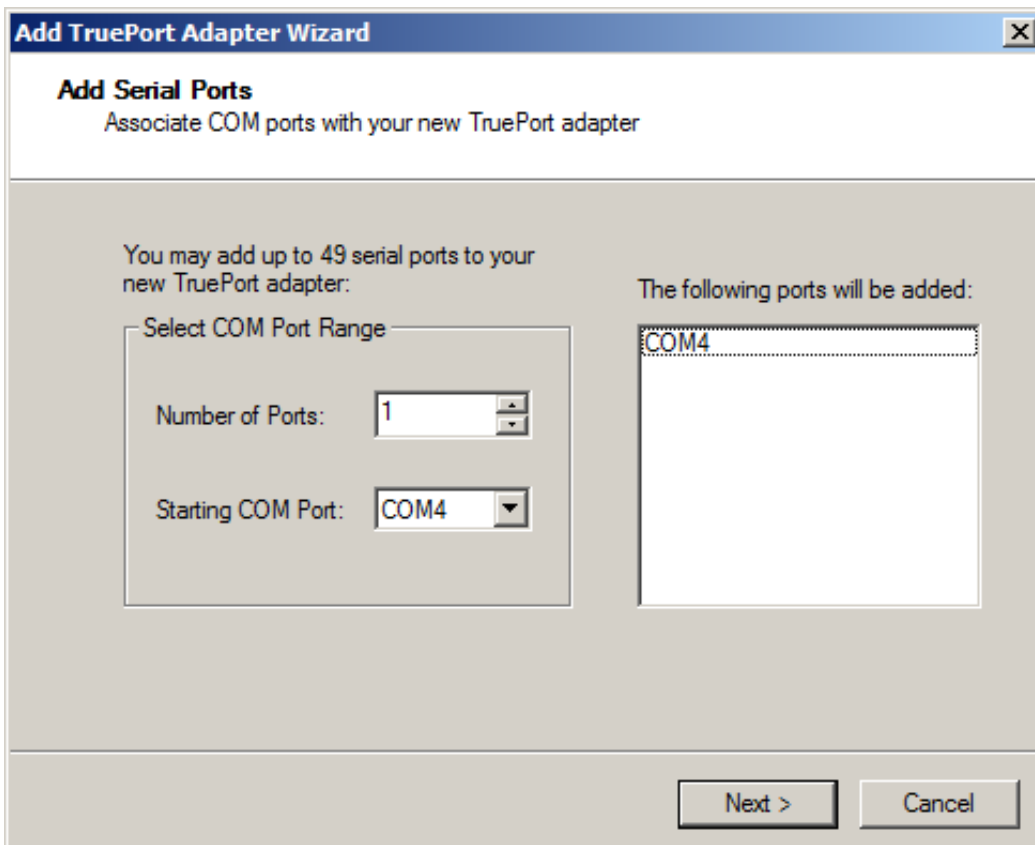


5. Укажите удобное название для виртуального порта и IP-адрес WebHMI. Нажмите "Next".

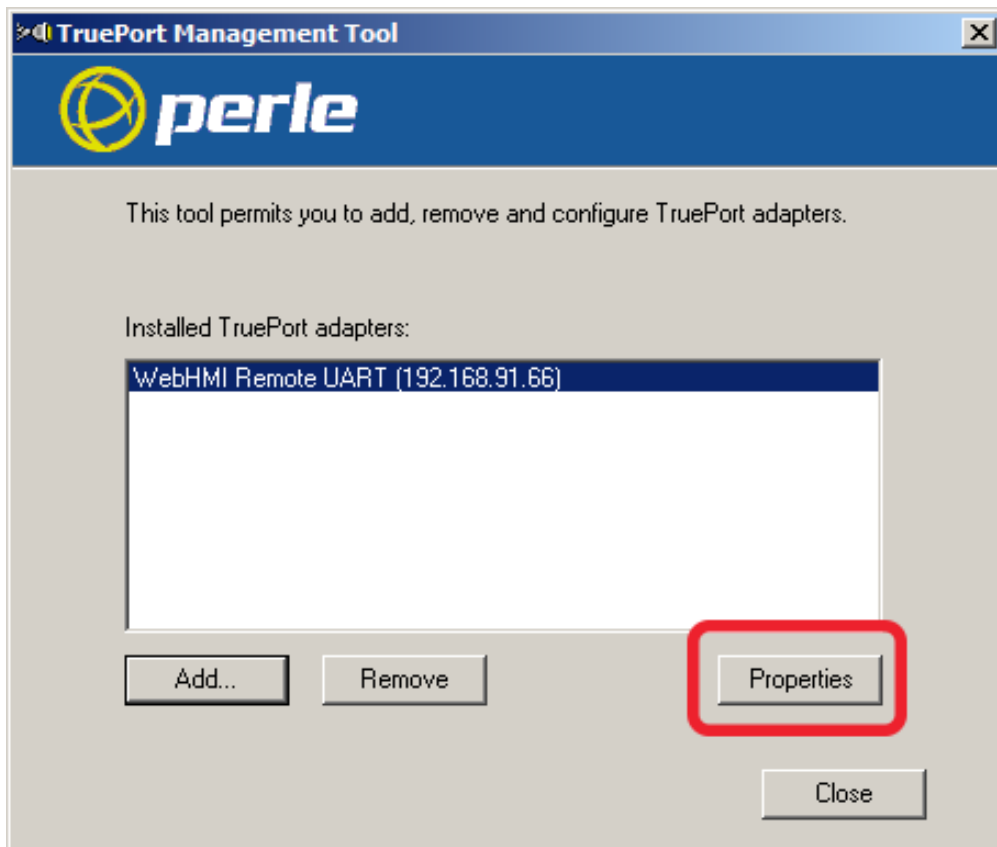


6. Укажите какое количество виртуальных портов вы хотите создать и стартовый адрес для них. Нажмите "Next".

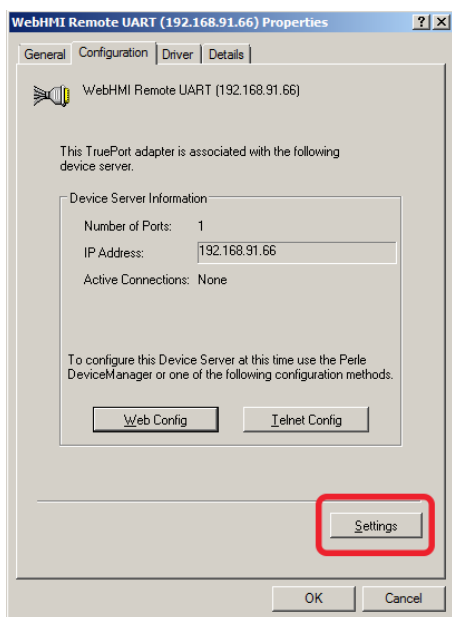
WebHMI позволяет создать множество виртуальных портов. Для подключения дополнительных UART необходимо использовать внешний USB-хаб и USB-RS485 или USB-RS232 преобразователи.



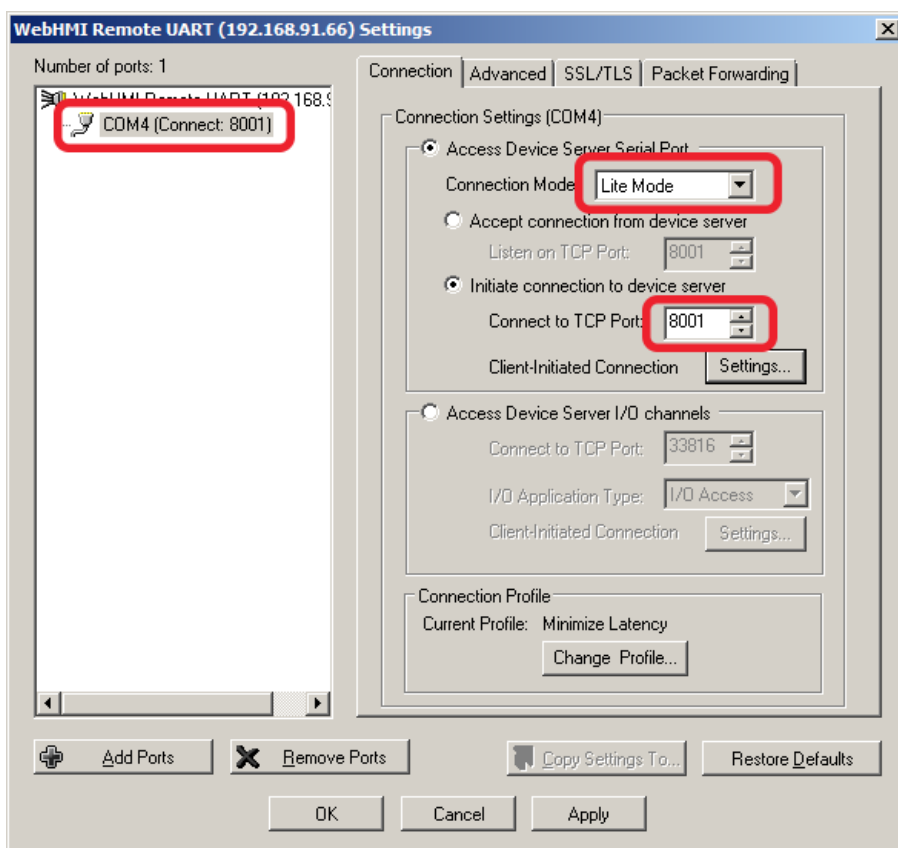
7. Выберите только что созданный порт и нажмите кнопку "Properties".



8. Откройте вкладку "Configuration" и нажмите кнопку "Settings".



9. В окне настроек портов выберите нужный порт в списке слева. Справа выберите "Initiate connection to device server" и укажите порт, который вы увидели на шаге 2. В нашем случае это 8001. Нажмите "OK", затем "Close".



Если вы все сделали правильно то теперь у вас в системе появился виртуальный COM-порт COM4 который можно использовать в любом приложении для доступа к ПЛК.

Обратите внимание что все настройки самого порта указываются в "Connections" WebHMI. Все настройки (скорость, четность, стоп биты), указанные в Windows-приложениях будут проигнорированы.

## Функция Modbus/TCP сервер

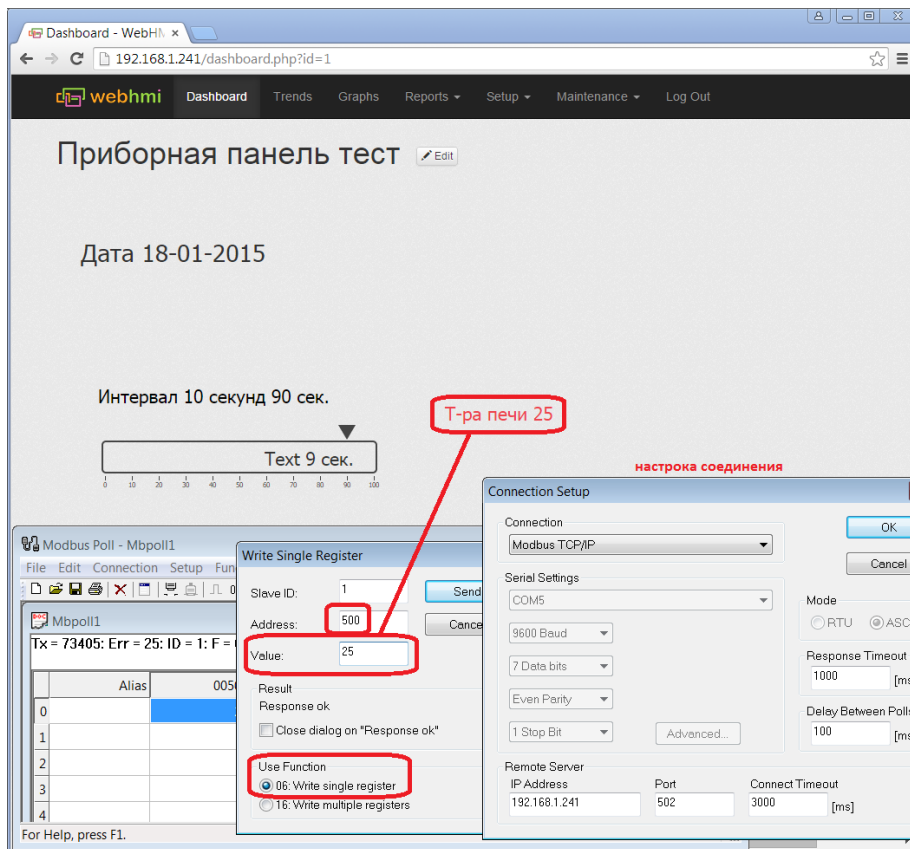
Существует возможность внешнего доступа к регистрам WebHMI по протоколу Modbus/TCP. Для этого служит меню **Setup --> Modbus/TCP server**. При указании в списке для регистров опции Export, данный регистр может быть доступен извне по протоколу Modbus/TCP, с адресом, введенным в поле **ModBus/TCP Address**. Формат адреса - значение от 0..65535. В данном случае речь идет не об адресе модели данных Modbus (например 40001-49999 для holding - регистров), а об адресе в сообщении (адресе PDU).

- **чтение** - используется функция Modbus номер 4, Input Register;
- **запись** - используются либо **функция номер 6 (Write Single)** для записи одного слова, либо **16 (Write Multiple)** для записи нескольких;

Интересным применением является связь разнородных систем между собой, когда устройства WebHMI установленные в разных системах, позволяют им обмениваться данными между собой. В этом случае, например для чтения регистра по протоколу Modbus/TCP с другого устройства WebHMI, необходимо использовать адрес в формате IRxxxxx (где xxxxx - адрес на сервере).

Id	Register	Connection	Address	Export	ModBus/TCP Address
1	Интервал 10 сек.	Внутренние регистры	I10	<input checked="" type="checkbox"/>	1000
2	Интервал 20 сек.	Внутренние регистры	I20	<input type="checkbox"/>	
3	Текущая дата	Внутренние регистры	T0	<input type="checkbox"/>	
4	Температура печи	Внутренние регистры	D0	<input checked="" type="checkbox"/>	500
5	Задание температуры	Внутренние регистры	D1	<input type="checkbox"/>	
6	Синус	Delta PLC SX2	D1116	<input type="checkbox"/>	
7	Синус (шум)	Delta PLC SX2	D1117	<input type="checkbox"/>	
8	Длительность EL1	Внутренние регистры	EL1	<input type="checkbox"/>	
9	Событие период	Внутренние регистры	S1	<input type="checkbox"/>	

Пример проверки работы функциональности с помощью утилиты modbus poll (или аналогичной - modscan и др.):





---

# Взаимодействие через API

---

## Описание API

---

С WebNMI можно взаимодействовать с помощью REST<sup>[1]</sup>-подобного программного интерфейса (API).

API предоставляет доступ ко всем основным функциям необходимым для сбора данных и управления системой. Вызов метода API представляем собой HTTP-запрос с нужному URL с необходимыми параметрами.

Доступны такие методы:

- Список соединений<sup>[2]</sup>;
- Список регистров<sup>[3]</sup>;
- Получение лога регистров<sup>[4]</sup>;
- Получение текущих значений регистров<sup>[5]</sup>;
- Запись нового значения в регистр<sup>[6]</sup>;
- Список словарей<sup>[7]</sup>;
- Список трендов<sup>[8]</sup>;
- Список графиков<sup>[9]</sup>;
- Список изображений<sup>[10]</sup>;
- Получение данных для графика<sup>[11]</sup>;
- Список событий<sup>[12]</sup>;
- Получение данных для события<sup>[13]</sup>;
- Получение данных о локальном времени<sup>[14]</sup>;
- Список панелей<sup>[15]</sup>;
- Список блоков панелей<sup>[16]</sup>;

Все данные возвращаются в формате JSON<sup>[17]</sup>. Некоторые параметры необходимо передавать в виде части URL (обычно это ID события, регистра и т.д). Остальные параметры передаются в виде заголовков HTTP-запроса (временной интервал, API Key и другие). Более подробно смотрите в описании конкретного метода.

Для обеспечения безопасности и разграничения прав доступа используются ключи доступа API (API Keys<sup>[18]</sup>).

На устройстве WebNMI есть песочница API. Она доступна по адресу вида <http://192.168.1.1/api-docs/>. Это мини-версия документации по API с возможностью выполнить любой запрос прямо из браузера и увидеть результат его работы.

## API - Список соединений

---

Для получения списка всех соединений необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/connections`.

Для получения информации о конкретном соединении необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/connections/1`. **Здесь 1 - это ID нужного соединения.**

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса списка соединений:

**URL:** `http://192.168.1.1/api/connections`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "title": "Внутренние регистры",
```

```
    "addr": "",
    "baudrate": "9600",
    "parity": "e",
    "stopbit": "1",
    "bitscount": "8",
    "model": "3",
    "protocol": "4",
    "timeout": "100",
    "timeout2": "",
    "device": "/dev/rs485",
    "disabled": "0",
    "mode": "0"
  },
  {
    "id": "2",
    "title": "S7 200",
    "addr": "2",
    "baudrate": "9600",
    "parity": "e",
    "stopbit": "1",
    "bitscount": "8",
    "model": "1",
    "protocol": "3",
    "timeout": "200",
    "timeout2": "",
    "device": "/dev/rs485",
    "disabled": "1",
    "mode": "0"
  }
]
```

Пример запроса информации об соединений номер 2:

**URL:** <http://192.168.1.1/api/connections/2>

**Method:** GET

**Заголовки:**

Accept: application/json

Content-Type: application/json

Accept-Encoding: gzip, deflate, sdch

X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7

**Пример ответа:**

```
{
  "id": "2",
  "title": "S7 200",
  "addr": "2",
  "baudrate": "9600",
```

```

"parity": "e",
"stopbit": "1",
"bitscount": "8",
"model": "1",
"protocol": "3",
"timeout": "200",
"timeout2": "",
"device": "/dev/rs485",
"disabled": "1",
"mode": "0"
}

```

---

#### Атрибуты соединения:

<b>id</b>	int	Уникальный идентификатор соединения
<b>title</b>	string	Название соединения
<b>addr</b>	string	Адрес устройства на шине или сетевой адрес для TCP соединения
<b>baudrate</b>	int	Скорость обмена данными для последовательных соединений
<b>parity</b>	string	Контроль четности для последовательных соединений. e = Even, o = Odd, n = None
<b>stopbit</b>	int	Количество стоповых битов для последовательных соединений
<b>model</b>	int	Идентификатор типа устройства
<b>protocol</b>	int	Идентификатор протокола обмена
<b>timeout</b>	int	Максимальное время ожидания ответа на запрос от устройства
<b>device</b>	string	Системное имя адаптера для последовательных соединений
<b>disabled</b>	int	0 = соединение активно, 1 = соединение отключено
<b>mode</b>	int	0 = нормальный режим работе. 1 = режим Virtual UART

---

#### Идентификаторы типа устройства:

0 = Delta DVP  
1 = Siemens Simatic S7 200  
2 = Generic ModBus  
3 = Internal WebHMI Registers  
4 = Generic Owen  
5 = Generic DF1

---

#### Идентификаторы протокола обмена:

0 = ModBus ASCII  
1 = ModBus RTU (еще не доступен)  
2 = ModBus TCP  
3 = Siemens PPI  
4 = Owen  
5 = Allen Bradley DF1  
6 = Internal WebHMI Registers

---

# API - Список регистров

---

Для получения списка всех регистров необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/registers`.

Для получения информации о конкретном регистре необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/registers/1`. **Здесь 1 - это ID нужного регистра.**

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

Пример запроса списка регистров:

**URL:** `http://192.168.1.1/api/registers`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "plcid": "2",
    "title": "Идет упаковка",
    "addr": "M1.0",
    "modbusAddr": "",
    "type": "0",
    "dictid": "0",
    "value_format": "0",
    "multiplicator": "",
    "shift": "",
    "save_to_log": "0",
    "measureUnits": "",
    "has_normal": "0",
    "normal_regid": "0",
    "normal_regid2": "0",
    "normal_dynamic_value": "0",
    "normal_binary_value": "",
    "normal_binary_text": "",
    "normal_color": "#33d356",
```

```
"normal_min_regid": "-1",
"normal_min_delta": "",
"normal_max_regid": "-1",
"normal_max_delta": "",
"normal_min_val": "",
"normal_max_val": "",
"has_disabled": "0",
"disabled_regid": "0",
"disabled_regid2": "0",
"disabled_dynamic_value": "0",
"disabled_binary_value": "",
"disabled_binary_text": "",
"disabled_color": "#cccccc",
"disabled_min_regid": "-1",
"disabled_min_delta": "",
"disabled_max_regid": "-1",
"disabled_max_delta": "",
"disabled_min_val": "",
"disabled_max_val": "",
"has_warning": "0",
"warning_regid": "0",
"warning_regid2": "0",
"warning_dynamic_value": "0",
"warning_binary_value": "",
"warning_binary_text": "",
"warning_color": "#f0b417",
"warning_min_regid": "-1",
"warning_min_delta": "",
"warning_max_regid": "-1",
"warning_max_delta": "",
"warning_min_val": "",
"warning_max_val": "",
"has_alert": "0",
"alert_regid": "0",
"alert_regid2": "0",
"alert_dynamic_value": "0",
"alert_binary_value": "",
"alert_binary_text": "",
"alert_color": "#ee434d",
"alert_min_regid": "-1",
"alert_min_delta": "",
"alert_max_regid": "-1",
"alert_max_delta": "",
"alert_min_val": "",
"alert_max_val": "",
"timeformat": "Y-m-d H:i:s",
"log_tolerance": "",
```

```
"precision_digits": "",
"min_log_interval": "",
"max_log_interval": "",
"max_graph_interval": "",
"max_allowed_value": "",
"min_allowed_value": "",
"rw_mode": "",
"min_reasonable_value": "",
"max_reasonable_value": "",
"save_graph_data": "0",
"graph_interval": "1",
"graph_color": "#ff0000",
"disabled": "0",
"nocloud": "",
"delta": "0",
"delay": "0"
},
{
  "id": "2",
  "plcid": "2",
  "title": "Рецепт",
  "addr": "VW100",
  "modbusAddr": "",
  "type": "2",
  "dictid": "1",
  "value_format": "0",
  "multiplier": "",
  "shift": "",
  "save_to_log": "0",
  "measureUnits": "",
  "has_normal": "0",
  "normal_regid": "0",
  "normal_regid2": "0",
  "normal_dynamic_value": "0",
  "normal_binary_value": "",
  "normal_binary_text": "",
  "normal_color": "#32d355",
  "normal_min_regid": "-1",
  "normal_min_delta": "",
  "normal_max_regid": "-1",
  "normal_max_delta": "",
  "normal_min_val": "",
  "normal_max_val": "",
  "has_disabled": "0",
  "disabled_regid": "0",
  "disabled_regid2": "0",
  "disabled_dynamic_value": "0",
```

```
"disabled_binary_value": "",
"disabled_binary_text": "",
"disabled_color": "#cccccc",
"disabled_min_regid": "-1",
"disabled_min_delta": "",
"disabled_max_regid": "-1",
"disabled_max_delta": "",
"disabled_min_val": "",
"disabled_max_val": "",
"has_warning": "0",
"warning_regid": "0",
"warning_regid2": "0",
"warning_dynamic_value": "0",
"warning_binary_value": "",
"warning_binary_text": "",
"warning_color": "#f0b416",
"warning_min_regid": "-1",
"warning_min_delta": "",
"warning_max_regid": "-1",
"warning_max_delta": "",
"warning_min_val": "",
"warning_max_val": "",
"has_alert": "0",
"alert_regid": "0",
"alert_regid2": "0",
"alert_dynamic_value": "0",
"alert_binary_value": "",
"alert_binary_text": "",
"alert_color": "#ee434d",
"alert_min_regid": "-1",
"alert_min_delta": "",
"alert_max_regid": "-1",
"alert_max_delta": "",
"alert_min_val": "",
"alert_max_val": "",
"timeformat": "Y-m-d H:i:s",
"log_tolerance": "",
"precision_digits": "",
"min_log_interval": "",
"max_log_interval": "",
"max_graph_interval": "",
"max_allowed_value": "",
"min_allowed_value": "",
"rw_mode": "",
"min_reasonable_value": "",
"max_reasonable_value": "",
"save_graph_data": "0",
```



```
"graph_interval": "1",
"graph_color": "#ff0000",
"disabled": "0",
"nocloud": "",
"delta": "0",
"delay": "0"
},
{
  "id": "3",
  "plcid": "2",
  "title": "Счетчик упаковок",
  "addr": "C1",
  "modbusAddr": "",
  "type": "3",
  "dictid": "0",
  "value_format": "0",
  "multiplicator": "",
  "shift": "",
  "save_to_log": "0",
  "measureUnits": "",
  "has_normal": "0",
  "normal_regid": "0",
  "normal_regid2": "0",
  "normal_dynamic_value": "0",
  "normal_binary_value": "",
  "normal_binary_text": "",
  "normal_color": "#33d356",
  "normal_min_regid": "-1",
  "normal_min_delta": "",
  "normal_max_regid": "-1",
  "normal_max_delta": "",
  "normal_min_val": "",
  "normal_max_val": "",
  "has_disabled": "0",
  "disabled_regid": "0",
  "disabled_regid2": "0",
  "disabled_dynamic_value": "0",
  "disabled_binary_value": "",
  "disabled_binary_text": "",
  "disabled_color": "#cccccc",
  "disabled_min_regid": "-1",
  "disabled_min_delta": "",
  "disabled_max_regid": "-1",
  "disabled_max_delta": "",
  "disabled_min_val": "",
  "disabled_max_val": "",
  "has_warning": "0",
```

```
"warning_regid": "0",
"warning_regid2": "0",
"warning_dynamic_value": "0",
"warning_binary_value": "",
"warning_binary_text": "",
"warning_color": "#f0b417",
"warning_min_regid": "-1",
"warning_min_delta": "",
"warning_max_regid": "-1",
"warning_max_delta": "",
"warning_min_val": "",
"warning_max_val": "",
"has_alert": "0",
"alert_regid": "0",
"alert_regid2": "0",
"alert_dynamic_value": "0",
"alert_binary_value": "",
"alert_binary_text": "",
"alert_color": "#ee434d",
"alert_min_regid": "-1",
"alert_min_delta": "",
"alert_max_regid": "-1",
"alert_max_delta": "",
"alert_min_val": "",
"alert_max_val": "",
"timeformat": "Y-m-d H:i:s",
"log_tolerance": "",
"precision_digits": "",
"min_log_interval": "",
"max_log_interval": "",
"max_graph_interval": "",
"max_allowed_value": "",
"min_allowed_value": "",
"rw_mode": "",
"min_reasonable_value": "",
"max_reasonable_value": "",
"save_graph_data": "0",
"graph_interval": "1",
"graph_color": "#ff0000",
"disabled": "0",
"nocloud": "",
"delta": "0",
"delay": "0"
}
]
```

Пример запроса информации о регистре номер 2:

**URL:** http://192.168.1.1/api/registers/2

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "id": "2",
  "plcid": "2",
  "title": "Рецепт",
  "addr": "VW100",
  "modbusAddr": "",
  "type": "2",
  "dictid": "1",
  "value_format": "0",
  "multiplicator": "",
  "shift": "",
  "save_to_log": "0",
  "measureUnits": "",
  "has_normal": "0",
  "normal_regid": "0",
  "normal_regid2": "0",
  "normal_dynamic_value": "0",
  "normal_binary_value": "",
  "normal_binary_text": "",
  "normal_color": "#32d355",
  "normal_min_regid": "-1",
  "normal_min_delta": "",
  "normal_max_regid": "-1",
  "normal_max_delta": "",
  "normal_min_val": "",
  "normal_max_val": "",
  "has_disabled": "0",
  "disabled_regid": "0",
  "disabled_regid2": "0",
  "disabled_dynamic_value": "0",
  "disabled_binary_value": "",
  "disabled_binary_text": "",
  "disabled_color": "#cccccc",
  "disabled_min_regid": "-1",
  "disabled_min_delta": "",
  "disabled_max_regid": "-1",
```

```
"disabled_max_delta": "",
"disabled_min_val": "",
"disabled_max_val": "",
"has_warning": "0",
"warning_regid": "0",
"warning_regid2": "0",
"warning_dynamic_value": "0",
"warning_binary_value": "",
"warning_binary_text": "",
"warning_color": "#f0b416",
"warning_min_regid": "-1",
"warning_min_delta": "",
"warning_max_regid": "-1",
"warning_max_delta": "",
"warning_min_val": "",
"warning_max_val": "",
"has_alert": "0",
"alert_regid": "0",
"alert_regid2": "0",
"alert_dynamic_value": "0",
"alert_binary_value": "",
"alert_binary_text": "",
"alert_color": "#ee434d",
"alert_min_regid": "-1",
"alert_min_delta": "",
"alert_max_regid": "-1",
"alert_max_delta": "",
"alert_min_val": "",
"alert_max_val": "",
"timeformat": "Y-m-d H:i:s",
"log_tolerance": "",
"precision_digits": "",
"min_log_interval": "",
"max_log_interval": "",
"max_graph_interval": "",
"max_allowed_value": "",
"min_allowed_value": "",
"rw_mode": "",
"min_reasonable_value": "",
"max_reasonable_value": "",
"save_graph_data": "0",
"graph_interval": "1",
"graph_color": "#ff0000",
"disabled": "0",
"nocloud": "",
"delta": "0",
"delay": "0"
```

```
}

```

Атрибуты регистров:

<b>id</b>	int	Уникальный идентификатор регистра
<b>title</b>	string	Название регистра
<b>plcid</b>	string	Идентификатор соединения по которому будеи происходить работа с этим регистром
<b>addr</b>	string	Адрес регистра
<b>modbusAddr</b>	string	Адрес регистра для встроенного Modbus TCP сервера
<b>type</b>	int	Идентификатор типа данных (см. ниже)
<b>value_format</b>	int	Идентификатор формата данных (см. ниже)
<b>dictid</b>	int	Идентификатор словаря
<b>multiplicator</b>	double	Множитель для математических преобразования значения
<b>shift</b>	double	Постоянная для математических преобразования значения
<b>save_to_log</b>	int	0 = не сохранять значения в лог, 1 = периодически сохранять значения в лог
<b>measureUnits</b>	string	Единицы измерения (метры, вольты и т.п.)
<b>has_normal</b>	int	1 = регистр имеет нормальное состояние, 0 = нормального состояния нет
<b>normal_regid</b>	int	Идентификатор регистра для сравнения (для регистров типа бит)
<b>normal_regid2</b>	int	Идентификатор второго регистра для сравнения (для регистров типа бит)
<b>normal_dynamic_value</b>	int	1 = сравнивать значение регистра со значениями из других регистров, 0 = сравнивать значение регистра с константами
<b>normal_binary_value</b>	int	Бинарное значение (1 или 0) которое включает нормальное состояние регистра
<b>normal_binary_text</b>	string	Текст для отображения нормального состояния
<b>normal_color</b>	string	Цвет нормального состояния
<b>normal_min_regid</b>	int	Идентификатор регистра с минимальным значением при диначическом сравнении
<b>normal_min_delta</b>	string	Дельта с минимальным значением при диначическом сравнении
<b>normal_max_regid</b>	int	Идентификатор регистра с максимальным значением при диначическом сравнении
<b>normal_max_delta</b>	string	Дельта с максимальным значением при диначическом сравнении
<b>normal_min_val</b>	double	Минимальное значение при обычном сравнении
<b>normal_max_val</b>	double	Максимальное значение при обычном сравнении
<b>has_disabled</b>	int	1 = регистр имеет выключенное состояние, 0 = выключенного состояния нет
<b>disabled_regid</b>	int	Идентификатор регистра для сравнения (для регистров типа бит)
<b>disabled_regid2</b>	int	Идентификатор второго регистра для сравнения (для регистров типа бит)
<b>disabled_dynamic_value</b>	int	1 = сравнивать значение регистра со значениями из других регистров, 0 = сравнивать значение регистра с константами
<b>disabled_binary_value</b>	int	Бинарное значение (1 или 0) которое включает выключенное состояние регистра
<b>disabled_binary_text</b>	string	Текст для отображения выключенного состояния
<b>disabled_color</b>	string	Цвет выключенного состояния
<b>disabled_min_regid</b>	int	Идентификатор регистра с минимальным значением при диначическом сравнении
<b>disabled_min_delta</b>	string	Дельта с минимальным значением при диначическом сравнении
<b>disabled_max_regid</b>	int	Идентификатор регистра с максимальным значением при диначическом сравнении

<b>disabled_max_delta</b>	string	Дельта с максимальным значением при динамическом сравнении
<b>disabled_min_val</b>	double	Минимальное значение при обычном сравнении
<b>disabled_max_val</b>	double	Максимальное значение при обычном сравнении
<b>has_warning</b>	int	1 = регистр имеет состояние предупреждения, 0 = состояния предупреждения нет
<b>warning_regid</b>	int	Идентификатор регистра для сравнения (для регистров типа бит)
<b>warning_regid2</b>	int	Идентификатор второго регистра для сравнения (для регистров типа бит)
<b>warning_dynamic_value</b>	int	1 = сравнивать значение регистра со значениями из других регистров, 0 = сравнивать значение регистра с константами
<b>warning_binary_value</b>	int	Бинарное значение (1 или 0) которое включает состояние предупреждения регистра
<b>warning_binary_text</b>	string	Текст для отображения состояния предупреждения
<b>warning_color</b>	string	Цвет состояния предупреждения
<b>warning_min_regid</b>	int	Идентификатор регистра с минимальным значением при динамическом сравнении
<b>warning_min_delta</b>	string	Дельта с минимальным значением при динамическом сравнении
<b>warning_max_regid</b>	int	Идентификатор регистра с максимальным значением при динамическом сравнении
<b>warning_max_delta</b>	string	Дельта с максимальным значением при динамическом сравнении
<b>warning_min_val</b>	double	Минимальное значение при обычном сравнении
<b>warning_max_val</b>	double	Максимальное значение при обычном сравнении
<b>has_alert</b>	int	1 = регистр имеет состояние тревоги, 0 = состояния тревоги нет
<b>alert_regid</b>	int	Идентификатор регистра для сравнения (для регистров типа бит)
<b>alert_regid2</b>	int	Идентификатор второго регистра для сравнения (для регистров типа бит)
<b>alert_dynamic_value</b>	int	1 = сравнивать значение регистра со значениями из других регистров, 0 = сравнивать значение регистра с константами
<b>alert_binary_value</b>	int	Бинарное значение (1 или 0) которое включает состояние тревоги регистра
<b>alert_binary_text</b>	string	Текст для отображения состояния тревоги
<b>alert_color</b>	string	Цвет состояния тревоги
<b>alert_min_regid</b>	int	Идентификатор регистра с минимальным значением при динамическом сравнении
<b>alert_min_delta</b>	string	Дельта с минимальным значением при динамическом сравнении
<b>alert_max_regid</b>	int	Идентификатор регистра с максимальным значением при динамическом сравнении
<b>alert_max_delta</b>	string	Дельта с максимальным значением при динамическом сравнении
<b>alert_min_val</b>	double	Минимальное значение при обычном сравнении
<b>alert_max_val</b>	double	Максимальное значение при обычном сравнении
<b>timeformat</b>	string	Формат вывода даты и времени для регистров с типом UnixTime
<b>log_tolerance</b>	double	Максимально пороговое изменение значения регистра при превышении которого значение будет записано в лог
<b>precision_digits</b>	int	Количество знаков после запятой для дробных чисел
<b>min_log_interval</b>	int	Записывать значения в лог не реже чем каждые X секунд
<b>max_log_interval</b>	int	Записывать значения в лог не чаще чем каждые X секунд
<b>max_graph_interval</b>	int	Соединять точки на графике если нет данных между ними менее чем X секунд
<b>max_allowed_value</b>	double	Максимально допустимое значение которое разрешено вводить оператору
<b>min_allowed_value</b>	double	Минимально допустимое значение которое разрешено вводить оператору

<b>rw_mode</b>	string	Не используется
<b>min_reasonable_value</b>	double	Минимально достоверное значение. При выходе за этот рубеж значение регистра будет сброшено в ноль.
<b>max_reasonable_value</b>	double	Максимально достоверное значение. При выходе за этот рубеж значение регистра будет сброшено в ноль
<b>save_graph_data</b>	int	Сохранять данные для построения графиков.
<b>graph_interval</b>	int	Точность данных для графиков. 0 = максимальная точность (в т.ч. брать данные из лога), 1 = каждую минуту, 2 = каждые 2 минуты, 5 = каждые 5 минут, 15 = каждые 15 минут, 60 = каждые 60 минут
<b>graph_color</b>	string	Цвет линии на графике
<b>disabled</b>	int	0 = регистр обрабатывается, 1 = регистр отключен
<b>nocloud</b>	int	0 = отправлять данные для графиков в облако, 1 = не отправлять данные графиков в облако
<b>delta</b>	int	0 = нормальный режим чтения, 1 = читать разницу между текущим и предыдущим значением, 2 = разница между текущим и предыдущим значением разделенная на время прошедшее между опросами этих значений (моментальная скорость)
<b>delay</b>	int	Период опроса регистра в миллисекундах. Не может быть меньше чем системная скорость опроса. 0 = скорость по умолчанию.

---

#### Идентификаторы типов данных:

- 0 = Bit
- 1 = Byte
- 2 = Word
- 3 = Double Word
- 4 = Unix Time

---

#### Идентификаторы форматов хранения данных:

- 0 = Unsigned, Integer
  - 1 = Signed, Integer
  - 2 = Unsigned, Fixed Point Float (0 ... +1)
  - 3 = Signed, Fixed Point Float (-1 ... +1)
  - 4 = Two's complement
  - 5 = Signed, Floating Point, 32 Bits, IEEE 754
  - 6 = Signed, Floating Point, 24 Bits
  - 7 = Signed, Floating Point, 32 Bits, IEEE 754, Reverse Word Order
-

# API - Получение лога регистров

---

Для получения записей из лога регистров необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/register-log`.

В заголовках запроса необходимо передать API key, границы интересующего временного интервала а также список идентификаторов регистров. X-WH-START и X-WH-END передаются в формате UnixTime. ID регистров передаются списком разделенным запятой в заголовке X-WH-REG-IDS. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
X-WH-START: 1422595501
X-WH-END: 1422599101
X-WH-REG-IDS: 8,10
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса записей из лога регистров:

**URL:** `http://192.168.1.1/api/register-log`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
X-WH-START: 1422595501
X-WH-END: 1422599101
X-WH-REG-IDS: 8,10
```

Пример ответа:

```
{
  "1422598266": [
    {
      "r": 8,
      "v": "6",
      "s": "-1"
    },
    {
      "r": 10,
      "v": "",
      "s": ""
    }
  ],
  "1422598271": [
    {
```





```
{
  "r": 8,
  "v": "10",
  "s": "-1"
},
{
  "r": 10,
  "v": "",
  "s": ""
}
],
"1422598295": [
  {
    "r": 8,
    "v": "15",
    "s": "-1"
  },
  {
    "r": 10,
    "v": "",
    "s": ""
  }
],
"1422598300": [
  {
    "r": 8,
    "v": "0",
    "s": "-1"
  },
  {
    "r": 10,
    "v": "",
    "s": ""
  }
],
"1422598305": [
  {
    "r": 8,
    "v": "5",
    "s": "-1"
  },
  {
    "r": 10,
    "v": "",
    "s": ""
  }
],
```

```
"1422598310": [  
  {  
    "r": 8,  
    "v": "10",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
],  
"1422598315": [  
  {  
    "r": 8,  
    "v": "15",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
],  
"1422598320": [  
  {  
    "r": 8,  
    "v": "0",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
],  
"1422598325": [  
  {  
    "r": 8,  
    "v": "5",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
]
```

```
],  
"1422598330": [  
  {  
    "r": 8,  
    "v": "10",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
],  
"1422598335": [  
  {  
    "r": 8,  
    "v": "15",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
],  
"1422598340": [  
  {  
    "r": 8,  
    "v": "0",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
],  
"1422598345": [  
  {  
    "r": 8,  
    "v": "5",  
    "s": "-1"  
  },  
  {  
    "r": 10,  
    "v": "",  
    "s": ""  
  }  
]
```

```

    }
  ],
  и т.д.
]

```

Формат возвращаемых данных следующий:

Возвращается массив, где индексом является время в формате UnixTime. Элементами массива есть еще один массив объектов.

У каждого объекта есть три свойства: r, v, s.

**r** - это ID регистра. Например, 10

**v** - это значение регистра в указанный момент времени или пустая строка если в этот момент времени нет данных для этого регистра.

**s** - это состояние (state) регистра, которое соответствовало ему в указанный момент времени. Unknown = -1, Disabled = 0, Normal = 1, Warning = 2, Alert = 3 или же пустая строка если значения регистра нет.

## API - Получение текущих значений регистров

Для получения текущих значений всех регистров необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/register-values`.

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```

Accept: application/json
Content-Type: application/json

```

Пример запроса текущих значений:

**URL:** `http://192.168.1.1/api/register-values`

**Method:** GET

Заголовки:

```

Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7

```

Пример ответа:

```

{
  "1": {
    "r": "1",
    "v": "-1",
    "s": "i"
  },
  "2": {
    "r": "2",

```

```
    "v": "-1",
    "s": "i"
  },
  "3": {
    "r": "3",
    "v": "-1",
    "s": "i"
  },
  "4": {
    "r": "4",
    "v": "1422600498",
    "s": "u"
  },
  "5": {
    "r": "5",
    "v": "1422600498",
    "s": "u"
  },
  "6": {
    "r": "6",
    "v": "-1",
    "s": "i"
  },
  "7": {
    "r": "7",
    "v": "-1",
    "s": "i"
  },
  "8": {
    "r": "8",
    "v": "18",
    "s": "u"
  },
  "9": {
    "r": "9",
    "v": "0",
    "s": "u"
  }
}
```

Формат возвращаемых данных следующий:

Возвращается объект, где названием свойств является время ID регистров. А самими свойствами являются еще одни объекты.

У каждого объекта есть три свойства: r, v, s.

**r** - это ID регистра. Например, 10

**v** - это значение регистра в текущий момент времени или "-1" если значение не получилось прочесть.

**s** - это состояние (state) регистра, которое соответствовало ему в указанный момент времени. Unknown = u,

Disabled = d, Normal = n, Warning = w, Alert = a, Incorrect = i.

## API - Запись нового значения в регистр

---

Для записи нового значения в регистр необходимо выполнить PUT-запрос на URL вида `http://192.168.1.1/api/register-values/23`. Здесь **23** - это ID регистра для которого необходимо изменить значение.

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

В теле запроса необходимо передать json-объект с единственным свойством "value":

```
{"value": "12"}
```

---

Пример записи значения 15 в регистр с ID = 5:

**URL:** `http://192.168.1.1/api/register-values/5`

**Method:** PUT

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Тело запроса:

```
{"value": "15"}
```

Если запрос валидный то API вернет HTTP-код 200.

---

# API - Список словарей

---

Для получения списка всех словарей необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/dictionaries`.

Для получения информации о конкретном словаре необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/dictionaries/1`. Здесь **1** - это ID нужного словаря.

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса списка всех словарей:

**URL:** `http://192.168.1.1/api/dictionaries`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "title": "Рецепты",
    "value": "1,Green Tea<br>2,Black Tea<br>3,Lemon Grass"
  },
  {
    "id": "2",
    "title": "Режим подогрева",
    "value": "0,Выкл.<br>1,Вкл.<br>2,Авария"
  }
]
```



---

Пример запроса информации о словаре номер 2:

**URL:** http://192.168.1.1/api/dictionaries/2

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "id": "2",
  "title": "Режим подогрева",
  "value": "0, Выкл.<br>1, Вкл.<br>2, Авария"
}
```

---

Атрибуты словаря:

**id** int Уникальный идентификатор словаря

**title** string Название словаря

**value** string Содержание словаря. Записи словаря разделены тегом <br>. Каждая запись содержит пару ключ-значение разделенные запятой. Например: 1,Green Tea. Это значит что при значении регистра = 1 будет отображаться строка "Green Tea"

---

# API - Список трендов

---

Для получения списка всех трендов необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/trends`.

Для получения информации о конкретном тренде необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/trends/1`. Здесь **1** - это ID нужного тренда.

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса списка всех трендов:

**URL:** `http://192.168.1.1/api/trends`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "regsid": "16",
    "title": "Горелка",
    "height": "350",
    "min_val": "0",
    "max_val": ""
  },
  {
    "id": "2",
    "regsid": "11,12,13",
    "title": "Токи А, В, С",
    "height": "350",
    "min_val": "0",
    "max_val": "1500"
  }
]
```

---

Пример запроса информации о тренде номер 2:

**URL:** http://192.168.1.1/api/trends/2

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "id": "2",
  "regsid": "11,12,13",
  "title": "Токи А, В, С",
  "height": "350",
  "min_val": "0",
  "max_val": "1500"
}
```

---

Атрибуты тренда:

<b>id</b>	int	Уникальный идентификатор тренда
<b>title</b>	string	Название тренда
<b>regsid</b>	string	Список идентификаторов регистров которые необходимо отображать на тренде. Регистры разделены запятыми.
<b>height</b>	int	Высота тренда в пикселях
<b>min_val</b>	double	Минимальное значение оси Y
<b>max_val</b>	double	Максимальное значение оси Y

---

# API - Список графиков

---

Для получения списка всех графиков необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/graphs`.

Для получения информации о конкретном графике необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/graphs/1`. **Здесь 1 - это ID нужного графика.**

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса списка всех графиков:

**URL:** `http://192.168.1.1/api/graphs`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "regsid": "10,11",
    "title": "Горелка 1 и 2",
    "height": "350",
    "min_val": "0",
    "max_val": ""
  },
  {
    "id": "2",
    "regsid": "8",
    "title": "Температура в печи",
    "height": "350",
    "min_val": "0",
    "max_val": "200"
  }
]
```

---

Пример запроса информации о графике номер 2:

**URL:** http://192.168.1.1/api/graphs/2

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "id": "2",
  "regsid": "8",
  "title": "Температура в печи",
  "height": "350",
  "min_val": "0",
  "max_val": "200"
}
```

---

Атрибуты графика:

<b>id</b>	int	Уникальный идентификатор графика
<b>title</b>	string	Название графика
<b>regsid</b>	string	Список идентификаторов регистров которые необходимо отображать на графике. Регистры разделены запятыми.
<b>height</b>	int	Высота графика в пикселях
<b>min_val</b>	double	Минимальное значение оси Y
<b>max_val</b>	double	Максимальное значение оси Y

---

# API - Список изображений

---

Для получения списка всех изображений необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/images`.

Для получения информации о конкретном изображении необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/images/1`. Здесь **1** - это **ID** нужного изображения.

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса списка всех изображений:

**URL:** `http://192.168.1.1/api/images`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "title": "Фон"
  },
  {
    "id": "2",
    "title": "Пламя"
  }
]
```

---

Пример запроса информации о изображений номер 2:

**URL:** `http://192.168.1.1/api/images/2`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

---

```
{
  "id": "2",
  "title": "Пламя"
}
```

Атрибуты изображения:

<b>id</b>	int	Уникальный идентификатор изображения
<b>title</b>	string	Название изображения

## API - Получение данных для графика

Для получения данных из лога графиков необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/graph-data/2`. **Здесь 2 - это ID нужного графика.**

В заголовках запроса необходимо передать API key, границы интересующего временного интервала (X-WH-START, X-WH-END) и количество значений (X-WH-SLICES). X-WH-START и X-WH-END передаются в формате UnixTime. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
X-WH-START: 1422797440
X-WH-END: 1422997405
X-WH-SLICES: 300
```

X-WH-SLICES - это количество значений. Временной интервал X-WH-START - X-WH-END будет разбит на X-WH-SLICES одинаковых промежутков и система возьмет наиболее детализированные значения для каждого из них.

X-WH-SLICES нужно выбирать исходя из необходимой детализации графика. Так, если необходимо построить график шириной 800 пикселей то нет смысла выбирать более 800 значений из лога. А иногда будет достаточно и 200-400 значений в зависимости от ширины линии на графике.

Для ускорения работы система может хранить предварительно усредненные значения регистров для различных интервалов (1 минута, 2 минуты, 5 минут, 15 минут, 1 час). В настройке каждого регистра можно указать необходимую детализацию хранимых данных. В зависимости от выбранного интервала времени и количества значений система будет пытаться взять данные из максимально подходящей таблицы усредненных значений. или же из лога регистров (если он включен для заданного регистра)

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

Пример запроса данных о графике номер 1:

**URL:** http://192.168.1.1/api/graph-data/1

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
X-WH-START: 1422854081
X-WH-END: 1422856235
X-WH-SLICES: 20
```

Пример ответа:

```
[
  {
    "8": "NaN;NaN;NaN",
    "10": "NaN;NaN;NaN",
    "16": "NaN;NaN;NaN",
    "x": 1422861065000
  },
  {
    "8": "NaN;NaN;NaN",
    "10": "NaN;NaN;NaN",
    "16": "NaN;NaN;NaN",
    "x": 1422861173000
  },
  {
    "8": "NaN;NaN;NaN",
    "10": "NaN;NaN;NaN",
    "16": "NaN;NaN;NaN",
    "x": 1422861281000
  },
  {
    "8": "NaN;NaN;NaN",
    "10": "NaN;NaN;NaN",
    "16": "NaN;NaN;NaN",
    "x": 1422861388000
  },
  {
    "8": "0.000000;9.292714;19.000000",
    "10": "NaN;NaN;NaN",
    "16": "0.000000;0.674894;1.000000",
    "x": 1422861496000
  },
  {
    "8": "0.000000;9.489933;19.000000",
```



```
"10": "NaN;NaN;NaN",
"16": "0.000000;0.742101;1.000000",
"x": 1422861604000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.747650;1.176471",
  "x": 1422861711000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.748471;1.176471",
  "x": 1422861819000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.746659;1.176701",
  "x": 1422861927000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.755068;1.176932",
  "x": 1422862034000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.741844;1.000000",
  "x": 1422862142000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.740291;1.000000",
  "x": 1422862250000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.756320;1.176701",
  "x": 1422862358000
},
{
  
```

```
"8": "0.000000;9.500000;19.000000",
"10": "NaN;NaN;NaN",
"16": "0.000000;0.746004;1.176471",
"x": 1422862465000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.754233;1.176932",
  "x": 1422862573000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.740343;1.000000",
  "x": 1422862681000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.751393;1.176932",
  "x": 1422862788000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.750868;1.176932",
  "x": 1422862896000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.745956;1.176701",
  "x": 1422863004000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.758306;1.176932",
  "x": 1422863111000
},
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.741780;1.000000",
  "x": 1422863219000
},
},
```

```
{
  "8": "0.000000;9.500000;19.000000",
  "10": "NaN;NaN;NaN",
  "16": "0.000000;0.741780;1.000000",
  "x": 1422863327000
},
{
  "8": "NaN;NaN;NaN",
  "10": "NaN;NaN;NaN",
  "16": "NaN;NaN;NaN",
  "x": 1422863435000
},
{
  "8": "NaN;NaN;NaN",
  "10": "NaN;NaN;NaN",
  "16": "NaN;NaN;NaN",
  "x": 1422863542000
}
]
```

Структура возвращаемых данных следующая. Возвращается массив объектов. В каждом объекте есть атрибут "x" - это временная метка данной записи. Формат времени - UnixTime умноженный на 1000 (размерность - миллисекунды).

Также в этом объекте есть атрибуты со значениями для каждого регистра. Имя атрибута - это ID регистра. Значение атрибута - это строка где через точку с запятой объединены минимальное, среднее и максимальное значение регистра на этом интервале. Например, "0.000000;0.755068;1.176932" означает что регистр принимал значения от 0 до 1.176932, при этом его среднее арифметическое было 0.755068. Если же в данном промежутке нет данных о значении данного регистра то система будет возвращать "NaN;NaN;NaN".

# API - Список событий

Для получения списка всех событий необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/events`.

Для получения информации о конкретном событии необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/events/1`. **Здесь 1 - это ID нужного события.**

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

Пример запроса списка всех событий:

**URL:** `http://192.168.1.1/api/events`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "2",
    "title": "Интервал",
    "hasDuration": "1",
    "start": "<xml
xmlns=\"http://www.w3.org/1999/xhtml\"><block type=\"wh_rules\"
id=\"1\" inline=\"false\" deletable=\"false\" x=\"-499\"
y=\"14\"><value name=\"IF\"><block type=\"logic_compare\"
id=\"2\" inline=\"true\"><field
name=\"OP\">EQ</field><value name=\"A\"><block
type=\"wh_reg\" id=\"3\"><field
name=\"REG\">8</field></block></value><value
name=\"B\"><block type=\"math_number\" id=\"4\"><field
name=\"NUM\">5</field></block></value></block></xml>\",
    "end": "<xml xmlns=\"http://www.w3.org/1999/xhtml\"><block
type=\"wh_rules\" id=\"1\" inline=\"false\" deletable=\"false\"
x=\"-596\" y=\"16\"><value name=\"IF\"><block
type=\"logic_compare\" id=\"2\" inline=\"true\"><field
name=\"OP\">EQ</field><value name=\"A\"><block
type=\"wh_reg\" id=\"3\"><field
```

```

name=\ "REG\">8</field></block></value><value
name=\ "B\"><block type=\ "math_number\" id=\ "4\"><field
name=\ "NUM\">10</field></block></value></block></value></block></xml>",
  "hasEndCondition": "1",
  "saveToLog": "1",
  "report_title": "Отчет \ "интервал\" ",
  "report_menu_title": "Интервал",
  "extra_regs": "-1^last^^Y-m-d H:i:s|2^last^^Y-m-d H:i:s",
  "color": "#ff0000",
  "savetologtype": "0",
  "loginterval": "",
  "parentid": "0",
  "showontimeline": "0",
  "description": ""
},
{
  "id": "3",
  "title": "Нажата кнопка A",
  "hasDuration": "1",
  "start": "<xml
xmlns=\ "http://www.w3.org/1999/xhtml\"><block type=\ "wh_rules\"
id=\ "1\" inline=\ "false\" deletable=\ "false\" x=\ "-613\"
y=\ "10\"><value name=\ "IF\"><block type=\ "logic_compare\"
id=\ "2\" inline=\ "true\"><field
name=\ "OP\">EQ</field><value name=\ "A\"><block
type=\ "wh_reg\" id=\ "4\"><field
name=\ "REG\">2</field></block></value><value
name=\ "B\"><block type=\ "math_number\" id=\ "11\"><field
name=\ "NUM\">1</field></block></value></block></value></block></xml>",
  "end": "<xml xmlns=\ "http://www.w3.org/1999/xhtml\"><block
type=\ "wh_rules\" id=\ "1\" inline=\ "false\" deletable=\ "false\"
x=\ "10\" y=\ "10\"><value name=\ "IF\"><block
type=\ "logic_compare\" id=\ "2\" inline=\ "true\"><field
name=\ "OP\">EQ</field></block></value></block></xml>",
  "hasEndCondition": "0",
  "saveToLog": "1",
  "report_title": "Нажатие кнопки A",
  "report_menu_title": "Кнопка A",
  "extra_regs": "-1^last^^Y-m-d H:i:s|-2^last^^Y-m-d H:i:s",
  "color": "#0084ff",
  "savetologtype": "0",
  "loginterval": "",
  "parentid": "0",
  "showontimeline": "1",
  "description": "Событие срабатывает при нажатии на кнопку A"
}
]

```

Пример запроса информации о событии номер 2:

**URL:** http://192.168.1.1/api/events/2

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "id": "2",
  "title": "Интервал",
  "hasDuration": "1",
  "start": "<xml xmlns=\"http://www.w3.org/1999/xhtml\"><block
type=\"wh_rules\" id=\"1\" inline=\"false\" deletable=\"false\"
x=\"-499\" y=\"14\"><value name=\"IF\"><block
type=\"logic_compare\" id=\"2\" inline=\"true\"><field
name=\"OP\">EQ</field><value name=\"A\"><block
type=\"wh_reg\" id=\"3\"><field
name=\"REG\">8</field></block></value><value
name=\"B\"><block type=\"math_number\" id=\"4\"><field
name=\"NUM\">5</field></block></value></block></value></block></xml>",
  "end": "<xml xmlns=\"http://www.w3.org/1999/xhtml\"><block
type=\"wh_rules\" id=\"1\" inline=\"false\" deletable=\"false\"
x=\"-596\" y=\"16\"><value name=\"IF\"><block
type=\"logic_compare\" id=\"2\" inline=\"true\"><field
name=\"OP\">EQ</field><value name=\"A\"><block
type=\"wh_reg\" id=\"3\"><field
name=\"REG\">8</field></block></value><value
name=\"B\"><block type=\"math_number\" id=\"4\"><field
name=\"NUM\">10</field></block></value></block></value></block></xml>",
  "hasEndCondition": "1",
  "saveToLog": "1",
  "report_title": "Отчет \"интервал\"",
  "report_menu_title": "Интервал",
  "extra_regs": "-1^last^^Y-m-d H:i:s|2^last^^Y-m-d H:i:s",
  "color": "#ff0000",
  "savetologtype": "0",
  "loginterval": "",
  "parentid": "0",
  "showontimeline": "0",
  "description": ""
}
```

---

Атрибуты события:

<b>id</b>	int	Уникальный идентификатор события
<b>title</b>	string	Название события
<b>hasDuration</b>	string	0 = событие срабатывает при каждом выполнении условия, 1 = событие имеет продолжительность во времени
<b>start</b>	string	Условие начала события
<b>end</b>	string	Условие окончания события
<b>hasEndCondition</b>	int	0 = у события нет условия его окончания, 1 = у события есть условие его окончания
<b>saveToLog</b>	int	0 = не сохранять событие в лог, 1 = сохранять событие в лог
<b>report_title</b>	string	Заголовок отчета о событии
<b>report_menu_title</b>	string	Заголовок пункта меню для отчета о событии
<b>extra_regs</b>	string	Список дополнительных регистров для сохранения в лог события.
<b>color</b>	string	Цвет для отображения события на TimeLine
<b>savetologtype</b>	int	0 = сохранять данные в лог один раз для события, 1 = сохранять данные в лог каждые X секунд
<b>loginterval</b>	int	Интервал сохранения данных в лог, секунды
<b>parentid</b>	int	Идентификатор родительского события. Дочернее событие может выполняться только когда выполняется родительское событие.
<b>showontimeline</b>	int	0 = не отображать событие на TimeLine, 1 = отображать событие на TimeLine
<b>description</b>	string	Комментарии с описание события. Предназначены для операторов и пользователей системы.

---

Формат поля extra\_regs:

Информация о дополнительных регистрах собрана в одну строку. Запись о каждом регистре разделена символом водопровода "|".

Запись о каждом регистре состоит из 4-х полей, разделенных символом "^". Поля в записи о регистре такие: идентификатор регистра, тип записываемого значения, название регистра, формат времени и даты.

Идентификатор регистра может быть положительным и отрицательным. Положительные идентификаторы означают ID регистров. Отрицательные идентификаторы - это специальные значения которые относятся к самому событию.

-1 = Start time - время когда началось событие

-2 = End time - время когда событие закончилось

-3 = Last update time - последнее время когда событие было обновлено в базе данных. Длинные события система перезаписывает в базу данных каждые 5 секунд что бы в случае любого сбоя или выключения питания информация о событии не пропала.

Тип записываемого значения:

first = первое значение регистра за время выполнения события

last = последнее значение регистра за время выполнения события

min = минимальное значение регистра за время выполнения события

max = максимальное значение регистра за время выполнения события

avg = среднее значение регистра за время выполнения события

Название регистра по умолчанию берется из информации о регистре. Если же в записи указано не пустое название то именно оно и будет отображаться в качестве названия этого сохраненного регистра в событии.

Формат времени и даты имеет смысл только для регистров со временем. Формат указывается в формате [функции date<sup>[1]</sup>] языка PHP.

---

Пример:

```
-1^last^Y-m-d H:i:s|2^last^Рецепт^Y-m-d H:i:s
```

Здесь описаны 2 дополнительные регистра:

```
-1^last^Y-m-d H:i:s
```

и

```
2^last^Y-m-d H:i:s
```

-1^last^Y-m-d H:i:s означает: Сохранять время начала события, отображать его в формате Y-m-d H:i:s

2^last^Рецепт^Y-m-d H:i:s означает: Сохранять последнее значение регистра с ID=2, его название в отчете будет Рецепт, и если это регистр не типа unixtime то формат времени и даты не будет использован

## API - Получение данных для события

---

Для получения данных из лога событий необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/event-data/2`. **Здесь 2 - это ID нужного события.**

В заголовках запроса необходимо передать API key и границы интересующего временного интервала. X-WH-START и X-WH-END передаются в формате UnixTime. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
X-WH-START: 1422797440
X-WH-END: 1422997405
```

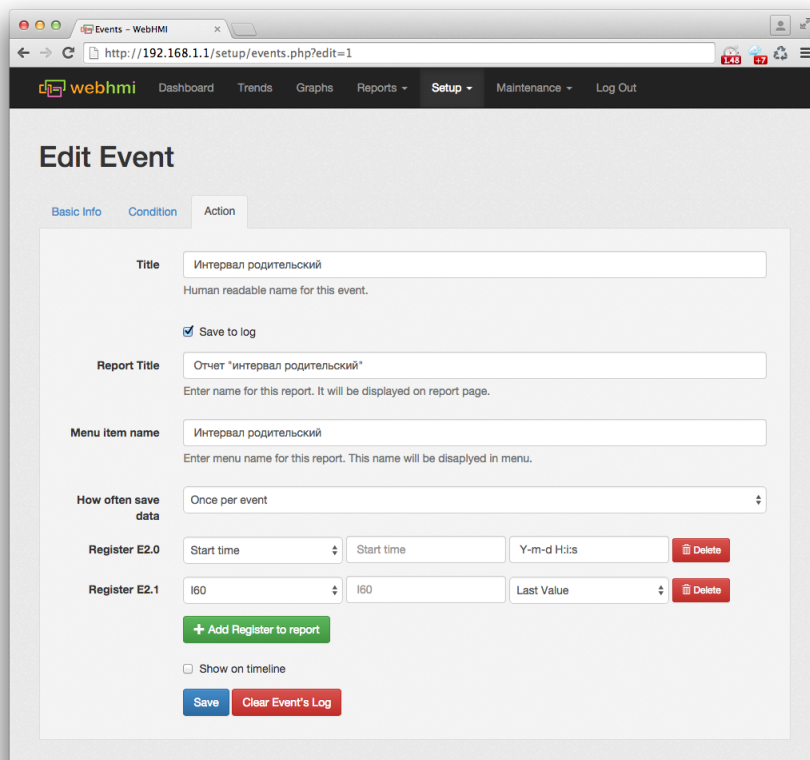
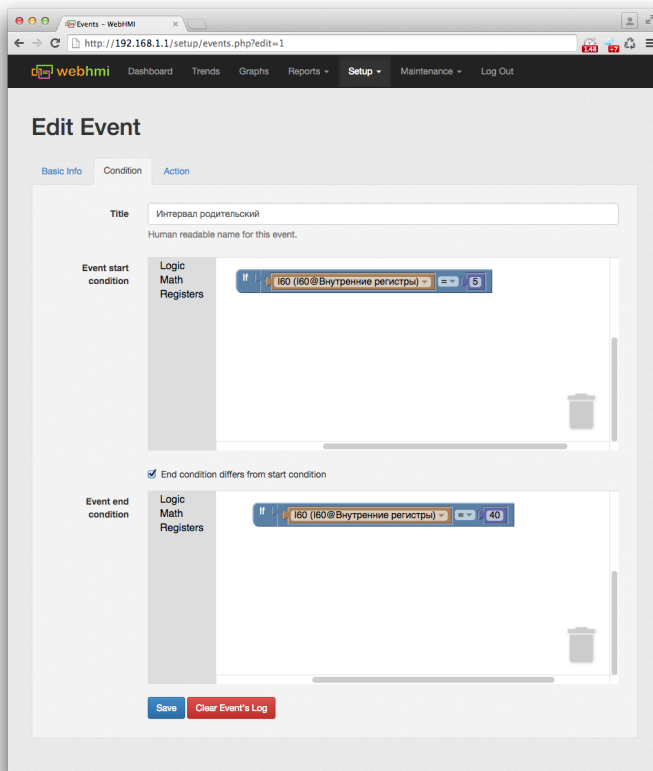
Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

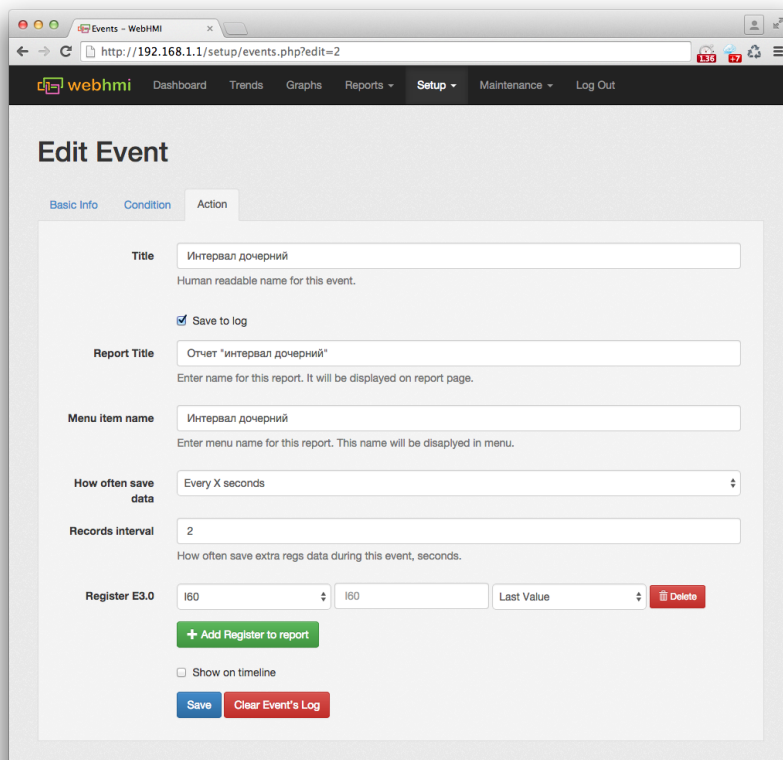
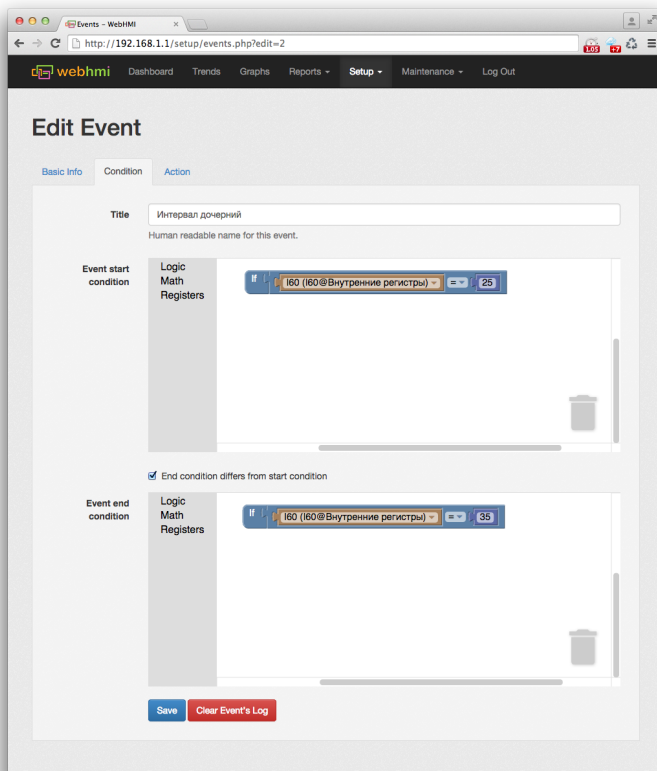
---

Рассмотрим пример из двух событий. Событие номер 1 является родительским. Оно начинается на 5-й секунде каждой минуты и заканчивается на 40-й секунде каждой минуты. В лог записываются данные один раз за все событие. В лог записывается время начала события и последнее значение интервального регистра I60. Скриншоты иллюстрируют настройку этого события.





Второе событие является дочерним событием для первого. Поэтому оно может выполняться только когда выполняется первое событие. Помимо этого условием оно ограничено так что начинает выполняться на 25-й секунде каждой минуты и заканчивается на 35-й. Данные для него записываются в лог каждые 2 секунды. В лог пишется текущее значение регистра I60. Скриншоты иллюстрируют настройку этого события.



Пример запроса данных о событии номер 1:

**URL:** <http://192.168.1.1/api/event-data/1>

**Method:** GET

**Заголовки:**

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
X-WH-START: 1422797440
X-WH-END: 1422997405
```

**Пример ответа:**

```
[
  {
    "rowid": "3050",
    "event_id": "1",
    "start_time": "1422797405",
    "end_time": "1422797440",
    "extra_regs": [
      {
        "E1.0": 1422797405
      },
      {
        "E1.1": 39
      }
    ]
  },
  {
    "rowid": "3052",
    "event_id": "1",
    "start_time": "1422797465",
    "end_time": "1422797500",
    "extra_regs": [
      {
        "E1.0": 1422797465
      },
      {
        "E1.1": 39
      }
    ]
  },
  {
    "rowid": "3054",
    "event_id": "1",
    "start_time": "1422797525",
    "end_time": "1422797560",
    "extra_regs": [
      {
        "E1.0": 1422797525
      },
      {
```

```
        "E1.1": 39
      }
    ]
  },
  {
    "rowid": "3056",
    "event_id": "1",
    "start_time": "1422797585",
    "end_time": "1422797680",
    "extra_regs": [
      {
        "E1.0": 1422797585
      },
      {
        "E1.1": 39
      }
    ]
  },
  {
    "rowid": "3058",
    "event_id": "1",
    "start_time": "1422797705",
    "end_time": "1422797740",
    "extra_regs": [
      {
        "E1.0": 1422797705
      },
      {
        "E1.1": 39
      }
    ]
  },
  {
    "rowid": "3060",
    "event_id": "1",
    "start_time": "1422797765",
    "end_time": "1422797800",
    "extra_regs": [
      {
        "E1.0": 1422797765
      },
      {
        "E1.1": 39
      }
    ]
  },
  {
```

```
"rowid": "3062",
"event_id": "1",
"start_time": "1422797825",
"end_time": "0",
"extra_regs": [
  {
    "E1.0": 1422797825
  },
  {
    "E1.1": 39
  }
]
}
```

Атрибуты данных события:

<b>rowid</b>	int	Уникальный идентификатор события в базе данных
<b>event_id</b>	int	ID события
<b>start_time</b>	int	Время начала события
<b>end_time</b>	int	Время окончания события. Если событие еще не окончилось то end_time = 0
<b>extra_regs</b>	object	Объект с данными о дополнительных регистрах которые сохраняются в логе событий. Ключ-значение.

Формат имени дополнительных регистров:

"E" + ID события + "." + порядковый номер регистра в событии начиная с нуля

Пример запроса данных для события номер 2:

**URL:** <http://192.168.1.1/api/event-data/2>

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
X-WH-START: 1422804505
X-WH-END: 1422800200
```

Пример ответа:

```
[
  {
    "rowid": "3201",
    "event_id": "2",
    "start_time": "1422804505",
    "end_time": "1422804515",
    "extra_regs": [
      {
        "E2.0": 25
      }
    ]
  }
]
```

```
    }
  ]
},
{
  "rowid": "3201",
  "event_id": "2",
  "start_time": "1422804505",
  "end_time": "1422804515",
  "extra_regs": [
    {
      "E2.0": 27
    }
  ]
},
{
  "rowid": "3201",
  "event_id": "2",
  "start_time": "1422804505",
  "end_time": "1422804515",
  "extra_regs": [
    {
      "E2.0": 29
    }
  ]
},
{
  "rowid": "3201",
  "event_id": "2",
  "start_time": "1422804505",
  "end_time": "1422804515",
  "extra_regs": [
    {
      "E2.0": 31
    }
  ]
},
{
  "rowid": "3201",
  "event_id": "2",
  "start_time": "1422804505",
  "end_time": "1422804515",
  "extra_regs": [
    {
      "E2.0": 33
    }
  ]
}
```

```
] ]
```

## API - Получение данных о локальном времени

Для получения информации о локальном времени необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/timeinfo`.

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

Пример:

**URL:** `http://192.168.1.1/api/timeinfo`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "timestamp": 1422861826,
  "timezone": "EET",
  "gmtoffset": 7200,
  "isdst": 0
}
```

Атрибуты даты и времени:

<b>timestamp</b>	unixtime	Текущее время в формате unixtime
<b>timezone</b>	string	Название временной зоны
<b>gmtoffset</b>	int	Смещение местного времени от UTC в секундах
<b>isdst</b>	int	0 = зимнее время, 1 = летнее время

# API - Список панелей

---

Для получения списка всех панелей необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/dashboards`.

Для получения информации о конкретной панели необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/dashboards/12`. Здесь **12** - это ID нужной панели.

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса списка панелей:

**URL:** `http://192.168.1.1/api/dashboards`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "title": "Упаковка",
    "hide_title": "0",
    "width": "900",
    "height": "450"
  },
  {
    "id": "2",
    "title": "Расход воды",
    "hide_title": "1",
    "width": "1024",
    "height": "800"
  }
]
```



---

Пример запроса информации об соединений номер 1:

**URL:** http://192.168.1.1/api/dashboards/1

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "id": "1",
  "title": "Упаковка",
  "hide_title": "0",
  "width": "900",
  "height": "450"
}
```

---

Атрибуты панелей:

<b>id</b>	int	Уникальный идентификатор панели
<b>title</b>	string	Название панели
<b>hide_title</b>	int	0 = отображать название панели, 1 = спрятать название панели
<b>width</b>	int	Ширина рабочей области панели в пикселях
<b>height</b>	int	Высота рабочей области панели в пикселях

---

# API - Список блоков панелей

---

Для получения списка всех блоков панелей необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/blocks`.

Для получения информации о конкретном блоке необходимо выполнить GET-запрос на URL вида `http://192.168.1.1/api/blocks/12`. **Здесь 12 - это ID нужного блока.**

В заголовках запроса необходимо передать API key. Например:

```
X-WH-APIKEY: F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Также в заголовках необходимо указать желаемый формат обмена данными. Сейчас поддерживается работа только с форматом JSON [17].

```
Accept: application/json
Content-Type: application/json
```

---

Пример запроса списка всех блоков:

**URL:** `http://192.168.1.1/api/blocks`

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
[
  {
    "id": "1",
    "dashboard_id": "1",
    "zindex": "1",
    "pos_left": "0",
    "pos_top": "0",
    "pos_width": "896",
    "pos_height": "445",
    "block_type": "Indicator",
    "title": "Фон",
    "url": "",
    "imageid": "1",
    "label": "",
    "align": "",
    "fontsize": "",
    "regid": "",
    "is_button": "false",
    "hide_value": "false",
    "start_value": "",
    "end_value": ""
```

```
    "colorize": "false"
  },
  {
    "id": "2",
    "dashboard_id": "1",
    "zindex": "3",
    "pos_left": "714",
    "pos_top": "277",
    "pos_width": "85",
    "pos_height": "25",
    "block_type": "TextValue",
    "title": "",
    "url": "",
    "imageid": "",
    "label": "",
    "align": "center",
    "fontsize": "4",
    "regid": "3",
    "is_button": "false",
    "hide_value": "false",
    "start_value": "",
    "end_value": "",
    "colorize": "false"
  },
  {
    "id": "3",
    "dashboard_id": "1",
    "zindex": "3",
    "pos_left": "450",
    "pos_top": "277",
    "pos_width": "110",
    "pos_height": "25",
    "block_type": "TextValue",
    "title": "",
    "url": "",
    "imageid": "",
    "label": "",
    "align": "left",
    "fontsize": "4",
    "regid": "5",
    "is_button": "",
    "hide_value": "",
    "start_value": "",
    "end_value": ""
  }
]
```

Пример запроса информации о блоке номер 1:

**URL:** http://192.168.1.1/api/blocks/1

**Method:** GET

Заголовки:

```
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip, deflate, sdch
X-WH-APIKEY:F3C74230818DA487BB2017CE5D0290F4DABCAFD7
```

Пример ответа:

```
{
  "id": "1",
  "dashboard_id": "1",
  "zindex": "1",
  "pos_left": "0",
  "pos_top": "0",
  "pos_width": "896",
  "pos_height": "445",
  "block_type": "Indicator",
  "title": "Фон",
  "url": "",
  "imageid": "1",
  "label": "",
  "align": "",
  "fontsize": "",
  "regid": "",
  "is_button": "false",
  "hide_value": "false",
  "start_value": "",
  "end_value": "",
  "colorize": "false"
}
```

Атрибуты блоков:

<b>id</b>	int	Уникальный идентификатор блока
<b>dashboard_id</b>	string	Идентификатор панели на которой отображается блок
<b>zindex</b>	int	Z-index блока (номер "слоя")
<b>pos_left</b>	int	Положение на панели от левого края в пикселях
<b>pos_top</b>	int	Положение на панели от верхнего края в пикселях
<b>pos_width</b>	int	Высота блока в пикселях
<b>pos_height</b>	int	Ширина блока в пикселях
<b>block_type</b>	string	Тип блока. Варианты: Indicator, TextValue, Gauge
<b>title</b>	string	Название блока
<b>url</b>	string	URL для ссылок

<b>imageid</b>	int	ID изображения которое нужно отображать
<b>label</b>	string	Текст который нужно выводить в блоке
<b>align</b>	string	Выравнивание текста. Варианты: left, right, center
<b>fontsize</b>	int	Размер шрифта, 1-5
<b>regid</b>	int	ID регистра из которого брать значение
<b>is_button</b>	int	0 = пользователь не может изменять значение, 1 = пользователь может изменять значение
<b>hide_value</b>	int	0 = отображать числовое значение, 1 = не отображать числовое значение
<b>start_value</b>	double	Начальное значение для Gauge
<b>end_value</b>	double	Конечное значение для Gauge
<b>colorize</b>	int	0 = отображать блок черным цветом, 1 = раскрашивать блок в соответствии со статусом регистра

## Пример доступа к данным из Excel

Ниже приведен простой пример запроса из Excel к WebHMI API.

Скачать пример

```
Const URL As String = "http://192.168.0.1/api/event-data/1"
Sub xmlHttp()

    Dim xmlHttp As Object
    Set xmlHttp = CreateObject("MSXML2.ServerXMLHTTP.6.0")
    xmlHttp.Open "GET", URL, False
    xmlHttp.setRequestHeader "Content-Type", "text/xml"
    xmlHttp.setRequestHeader "Accept", "application/json"
    xmlHttp.setRequestHeader "Host", "192.168.0.1"
    xmlHttp.setRequestHeader "Cookie", " "
    xmlHttp.setRequestHeader "X-WH-APIKEY", "6E51E728896794EBF406E2F070BE7AFBE49E90D4"
    xmlHttp.setRequestHeader "X-WH-START", "1388948941"
    xmlHttp.setRequestHeader "X-WH-END", "1399208143"
    xmlHttp.send

    Dim JSON As New JSON

    Dim p As Object
    Set p = JSON.parse(xmlHttp.ResponseText)

    i = 2
    j = 1
    For Each Item In p ' rows
        If (j = 1) Then
            Cells(i, j).NumberFormat = "yyyy-mm-dd hh:mm:ss"
        End If
        If (j = 2) Then
```

```
Cells(i, j).NumberFormat = "#.#" & "$#,##0.00_); [Red] ($#,##0.00) "
End If
For Each Item2 In Item 'columns
    For Each Item3 In Item2
        If (j = 1 And i > 1) Then
            Cells(i, j) = (Item2(Item3) / 86400) + 25569
        Else
            Cells(i, j) = Item2(Item3)
        End If
    Next
    Cells(1, j) = Item3
Next
j = j + 1
Next
i = i + 1
j = 1

Next

End Sub
```

## Пример доступа к данным из C/C++

---

Пример кода на языке C для выполнения запроса к API WebHMI.

```
#include <cstdlib>
#include <iostream>
#include <curl/curl.h>
#include <curl/easy.h>
#include <fstream>

using namespace std;
size_t write_func(void *ptr, size_t size, size_t nmemb, void *userdata)
{
    cout.write( (char * ) ptr, size*nmemb); // just output response
    to screen
    return size * nmemb;
}

int main(int argc, char** argv) {

    CURL *curl;
    CURLcode res;

    char errorBuffer[CURL_ERROR_SIZE];
```

```
// Create the GET request
struct curl_slist *headers = NULL;
headers = curl_slist_append(headers, "X-WH-APIKEY:
F3C74230818DA487BB2017CE5D0290F4DABCAFD7"); // API Key. If API Key is
wrong API will return 401 status code.
headers = curl_slist_append(headers, "X-WH-START: 1391165350"); //
Start time for report. Unixtime
headers = curl_slist_append(headers, "X-WH-END: 1391186951"); //
End time for report. Unixtime
headers = curl_slist_append(headers, "Accept: application/json");
// Currently we support JSON ONLY, required header
headers = curl_slist_append(headers, "Content-Type:
application/json"); // Currently we support JSON ONLY, required header

// Init CURL
curl = curl_easy_init();

if(curl) {
    curl_easy_setopt(curl, CURLOPT_URL,
"http://192.168.0.1/api/event-data/1"); // 1 = ID of report
    curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
    curl_easy_setopt(curl, CURLOPT_ERRORBUFFER, errorBuffer);
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_func);

    // Attempt to Connect the Server
    res = curl_easy_perform(curl);

    if (res == CURLE_OK) {
        long http_code = 0;
        curl_easy_getinfo (curl, CURLINFO_RESPONSE_CODE,
&http_code);
        if (http_code == 200 && res !=
CURLE_ABORTED_BY_CALLBACK) {
            cout << endl << "Success!"<< endl;
        } else {
            cout << endl << "Error! Code: " <<
http_code << endl;
        }
    } else {
        cout << "Connection Failed! Error: " <<
errorBuffer << endl;
    }

    // Close the connection
    curl_easy_cleanup(curl);
}
```

```
    return 0;  
}
```



# Типовые решения

## Подсчет количества упаковок

На пищевом производстве имеется упаковочная линия. Требовалось обеспечить ведение отчетов, содержащих количество упакованной продукции по видам выпускаемой продукции (рецептам). Данные отчеты должны быть доступны из локальной сети предприятия и из интернет.

Контроллер управления упаковочной линией - Siemens S7-200. Отчет должен содержать:

- Название рецепта
- Дату и время начала упаковки
- Дату и время окончания упаковки
- Количество упакованной продукции

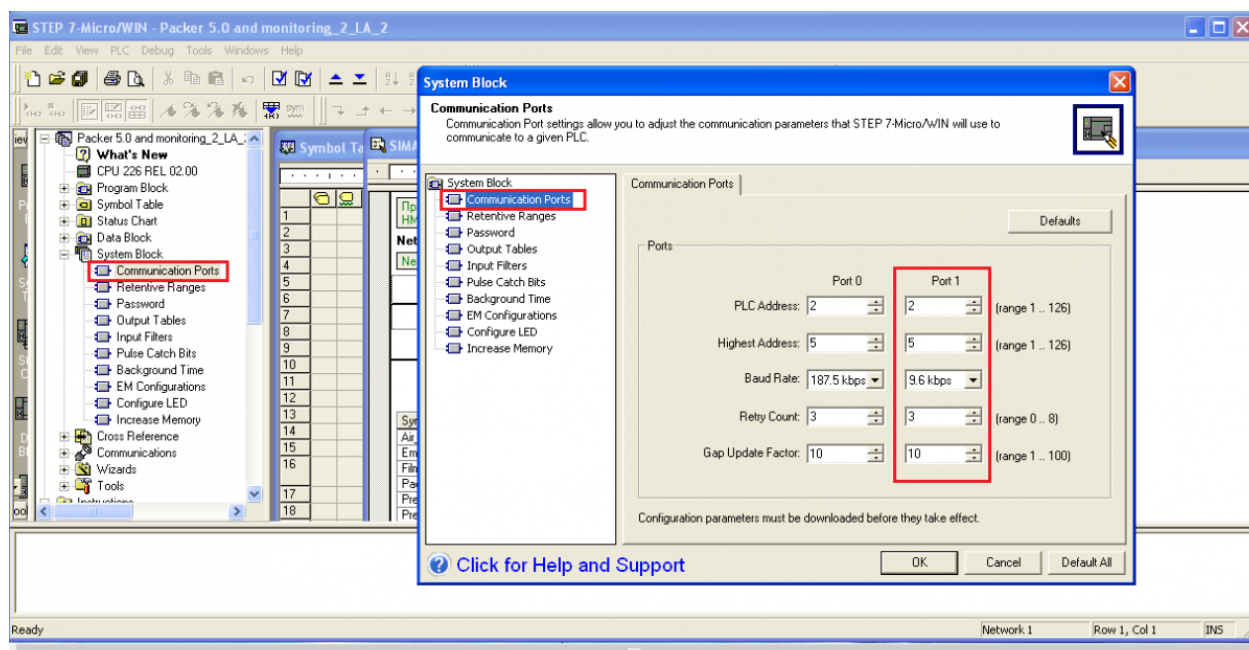
Для решения задачи очень подходил механизм событий <sup>[1]</sup>, имеющийся в WebHMI. Необходимо было определить "временное окно" во время которого происходит конкретная упаковка и записать данные даты/времени, установленного в этот момент рецепта, и максимального значения счетчика продукции на протяжении этого промежутка. После анализа листинга программы контроллера, который имелся в распоряжении службы эксплуатации, данные параметры были определены по таблице символов, содержащих комментарии к программе:

**сигнал "временного окна" идет упаковка** Адрес (в адресном пространстве) контроллера S7 200 **M3.4**, тип данных - бит

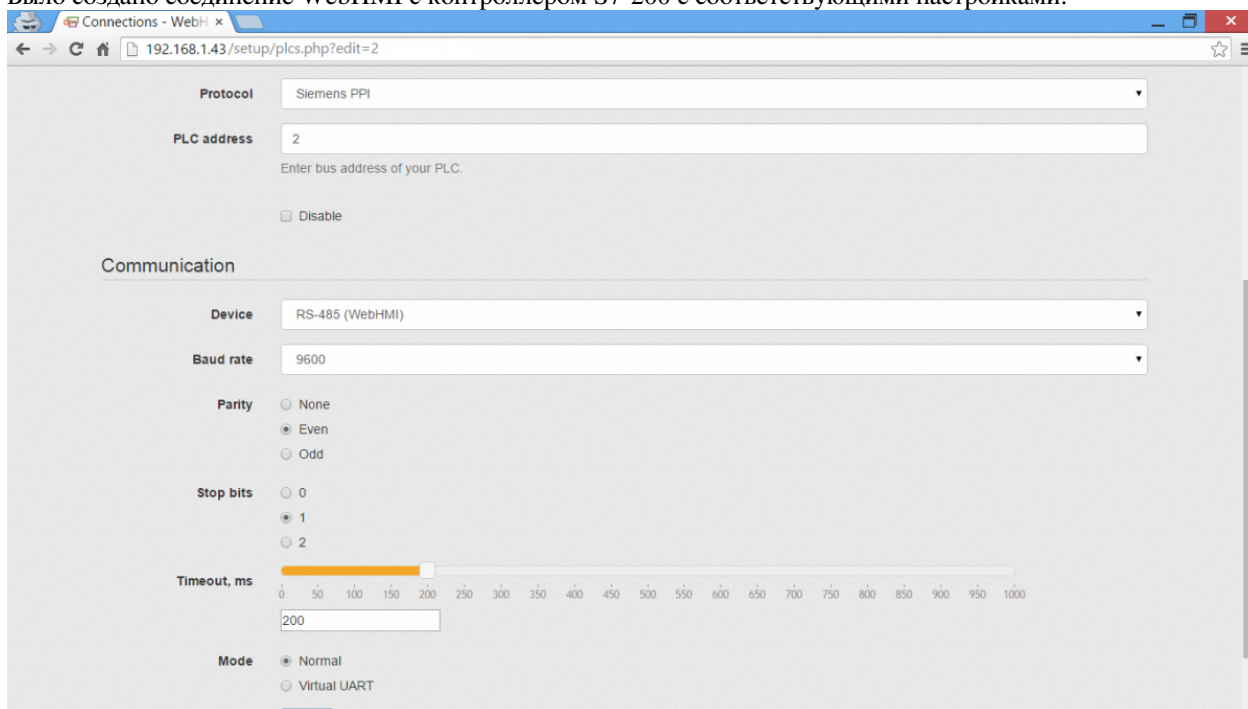
**номер рецепта**, соответственно которому высвечивалось название на имеющейся панели оператора, Адрес **VW540**, тип данных слово.

**счетчик упаковок** Адрес **C2**, тип данных двойное слово.

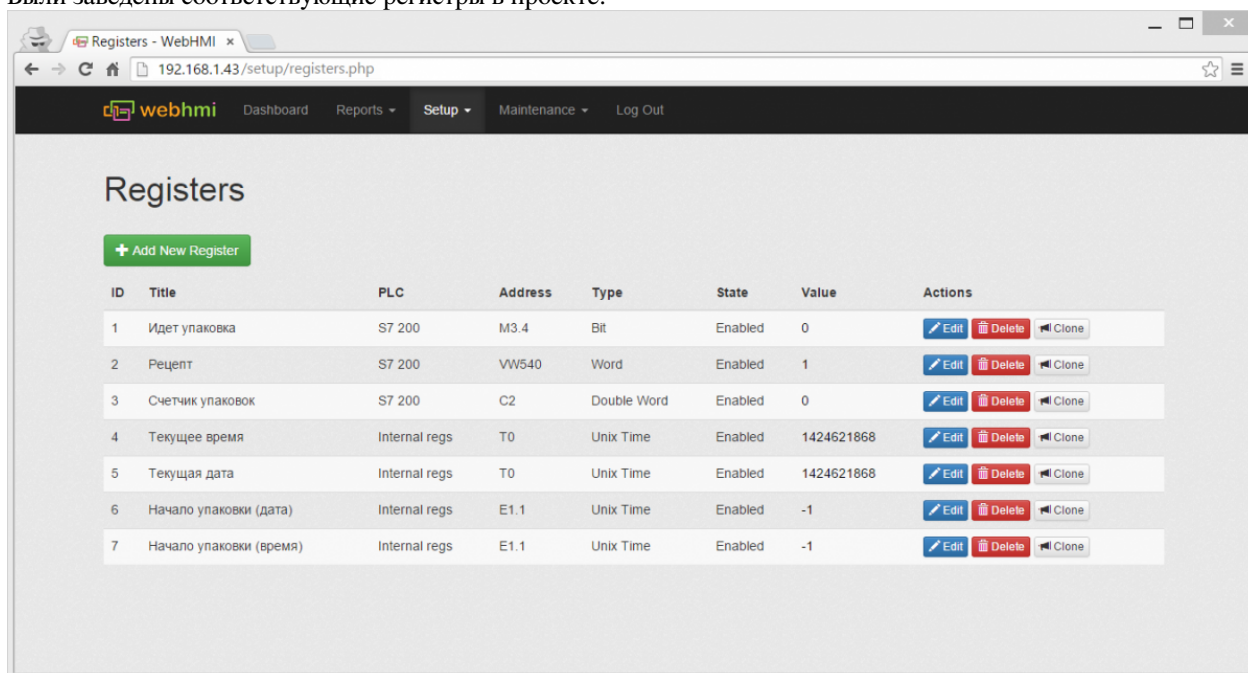
Параметры связи были определены из проекта контроллера, раздела **Control block**.



Было создано соединение WebHMI с контроллером S7-200 с соответствующими настройками:

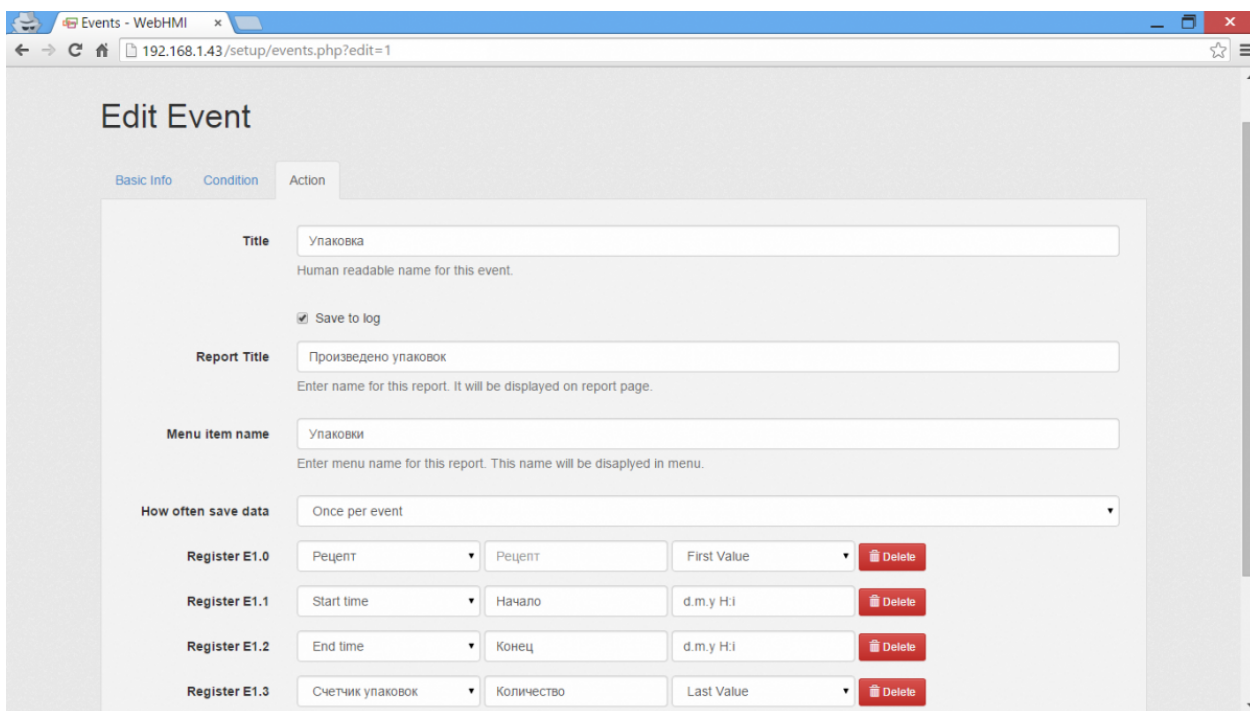


Были заведены соответствующие регистры в проекте:

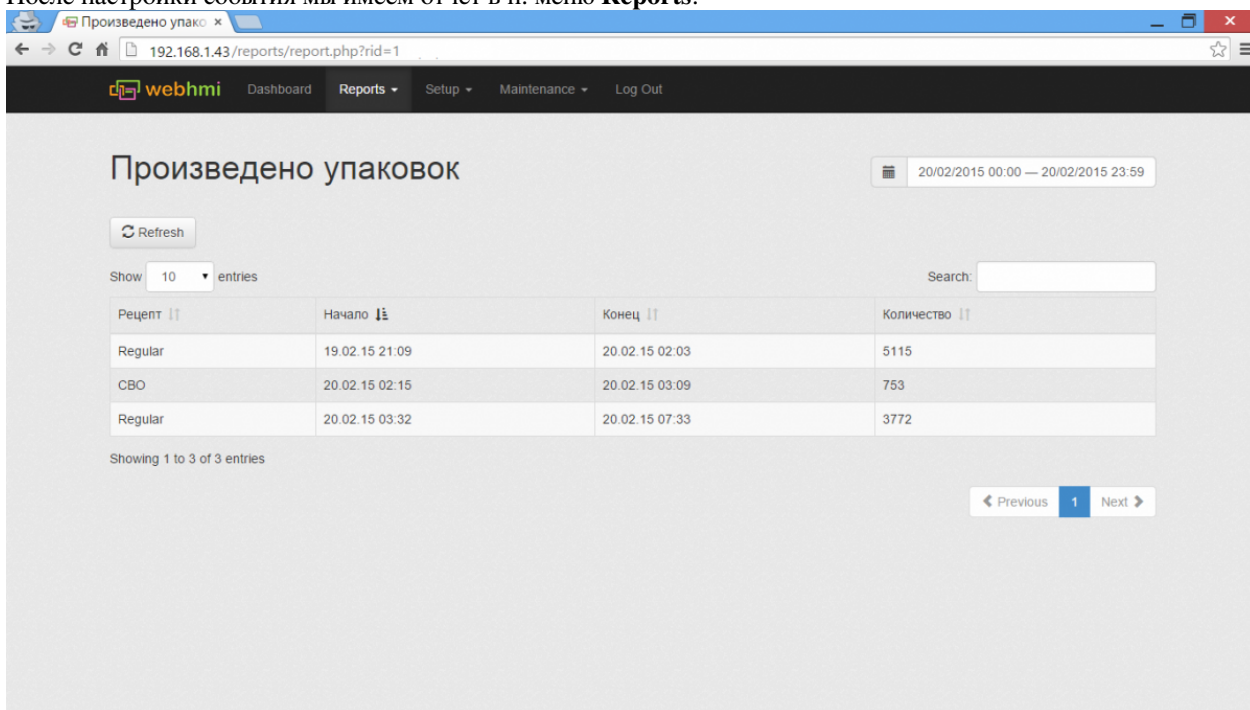


Регистры №4,5 были заведены для отображения текущей даты и времени на приборной панели. Регистры условий №6,7 - для отображения даты начала и окончания упаковки (из события).

Событие "Упаковка" является длящимся событием пока  $M3.4 = 1$  и настроено следующим образом:



После настройки события мы имеем отчет в п. меню **Reports**:



Приборная панель создана с применением только лишь элементов текста <sup>[2]</sup>, и примитива "прямоугольник" в графическом он-лайн редакторе Pixlr Editor.

The screenshot shows a web browser window with the URL `192.168.1.43/dashboard.php?id=1`. The dashboard header includes the 'webhmi' logo and navigation links for 'Dashboard', 'Reports', 'Setup', 'Maintenance', and 'Log Out'. The main content area is titled 'Упаковка' and contains a table with the following data:

Упаковка			
Regular			
	Дата	Время	Упаковок
Начало	22.02.15	18:02	
Текущее	22.02.15	18:54	20