

Методические указания по выполнению практических работ № 1-4

«Основы концепции Интернета вещей»

по дисциплине «Технологические основы Интернета вещей»

Оглавление

Практическая работа №1 – Знакомство с оборудованием	2
Основные правила техники безопасности при работе со стендом и оборудованием в его составе	2
Компоненты в составе стенда	3
Начало работы	5
Знакомство с базовым функционалом оборудования и предустановленные сценарии.....	5
Выключение стенда.....	6
Задание практической работы №1	6
Дополнительное задание практической работы №1.....	7
Практическая работа №2 – Настройка удаленного подключения.....	7
Веб-интерфейс стенда.....	7
Работа с веб-интерфейсом	8
Разделы интерфейса	9
Задание практической работы №2	15
Дополнительное задание практической работы №2.....	15
Практическая работа №3 – Обработка событий в системах Интернета вещей.....	15
Правила-скрипты в WirenBoard.....	15
Создание и редактирование правил	16
Задание практической работы №3	26
Дополнительное задание практической работы №3.....	26
Практическая работа №4 – Основы электротехники в системах Интернета вещей	27
Задание практической работы №4	27
Дополнительное задание практической работы №4.....	28
Требования к отчету по ПР №1-4:	29
Литература для изучения:.....	29

ВНИМАНИЕ! ВНУТРИ ЛАБОРАТОРНОГО СТЕНДА НАХОДИТСЯ ВЫСОКОЕ НАПРЯЖЕНИЕ 230V!

Практическая работа №1 – Знакомство с оборудованием

Внимательно ознакомьтесь с правилами техники безопасности и распишитесь в журнале техники безопасности. Без подписи в журнале студент НЕ ДОПУСКАЕТСЯ к работе со стендом (чемоданом) и оборудованию в составе стенда.

Правила техники безопасности представлены **отдельным документом**, основные его положения приведены ниже:

Основные правила техники безопасности при работе со стендом и оборудованием в его составе

Оборудование в составе стенда (чемодана) находится под высоким напряжением 230v и содержит движущиеся части.

Категорически запрещено:

1. Включать стенд в работу без разрешения преподавателя;
2. Проносить в аудиторию с оборудованием: еду, любые жидкости, верхнюю одежду;
3. Находиться в аудитории в верхней одежде;
4. Захламлять рабочее место, ставить сумки на столы с оборудованием;
5. Прикасаться руками или проводящими электрический ток предметами к любым контактам оборудования в составе стенда;
6. Самостоятельно подключать и отключать любое оборудование в составе стенда;
7. Самостоятельно подключать к стенду внешние устройства;
8. Переносить и закрывать подключенный к электросети стенд;
9. Прикасаться руками к движущимся частям оборудования: шаровому крану (!), лопастям вентилятора (!!!), – даже если в настоящий момент они неподвижны.

Перед работой со стендом необходимо:

1. Снять с кистей рук и запястий любые металлические украшения и украшения, которые могут зацепиться за подвижные части оборудования в составе стенда;
2. Надежно заправить волосы (убрать их в хвост) во избежание контакта с подвижными частями оборудования;
3. Снять/убрать галстуки и любые аксессуары, также способные зацепиться за оборудование.

В случае возникновения любых неисправностей необходимо немедленно сообщить о них преподавателю и не пытаться разрешить их самостоятельно!

Компоненты в составе стенда

Работа на практических занятиях проводится с использованием стенда (чемодана WB-demo-kit v.2), содержащего типовый набор оборудования в «умном доме».

Стенд состоит из компонентов, расположение которых можно увидеть на Рис. 1-2:

№	Название	№	Название	№	Название
1	Датчик температуры 1-wire DS18B20	13	Электросчетчик "Милур 307"	25	Индикатор 2 (вентилятор)
2	Датчик температуры 1-wire DS18B20	14	Автомат питания набора (L1)	26	Индикатор 3 (контактор)
3	Преобразователь 1-Wire — Modbus RTU WB-M1W2	15	Автомат питания вентилятора (L2)	27	Кнопка 1 (подача воды, сброс аварии по протечке)
4	Устройство ИК-управления WB-MIR	16	Автомат питания контактора (L3)	28	Кнопка 2 (вентилятор)
5	Настенный комбинированный датчик WB-MSW v.3	17	Трансформатор тока 25 А (L1)	29	Кнопка 3 (контактор)
6	RGB лента в профиле	18	Трансформатор тока 25 А (L2)	30	Импульсный счетчик расхода воды с имитацией потока
7	Блок питания HDR-30-24	19	Трансформатор тока 25 А (L3)	31	Шаровой кран с электроприводом
8	Контроллер Wiren Board 6 с модулем резервного питания для Wiren Board 6 WBMZ2-BATTERY	20	Контактор 220 В	32	Вентилятор
9	Модуль ввода-вывода WBIO-DO-R10A-8	21	Модуль реле 3-канальный WB-MR3		
10	Модуль обнаружения протечек WB-MWAC	22	Многоканальный измеритель WB-MAP12H		
11	Диммер светодиодных лент на DIN-рейку WB-MRGBW-D	23	Датчик протечки		
12	Комбинированный датчик WB-MS	24	Индикатор 1 (протечка)		



Рисунок 1 – Компоненты, расположенные на верхней крышке WB-demo-kit

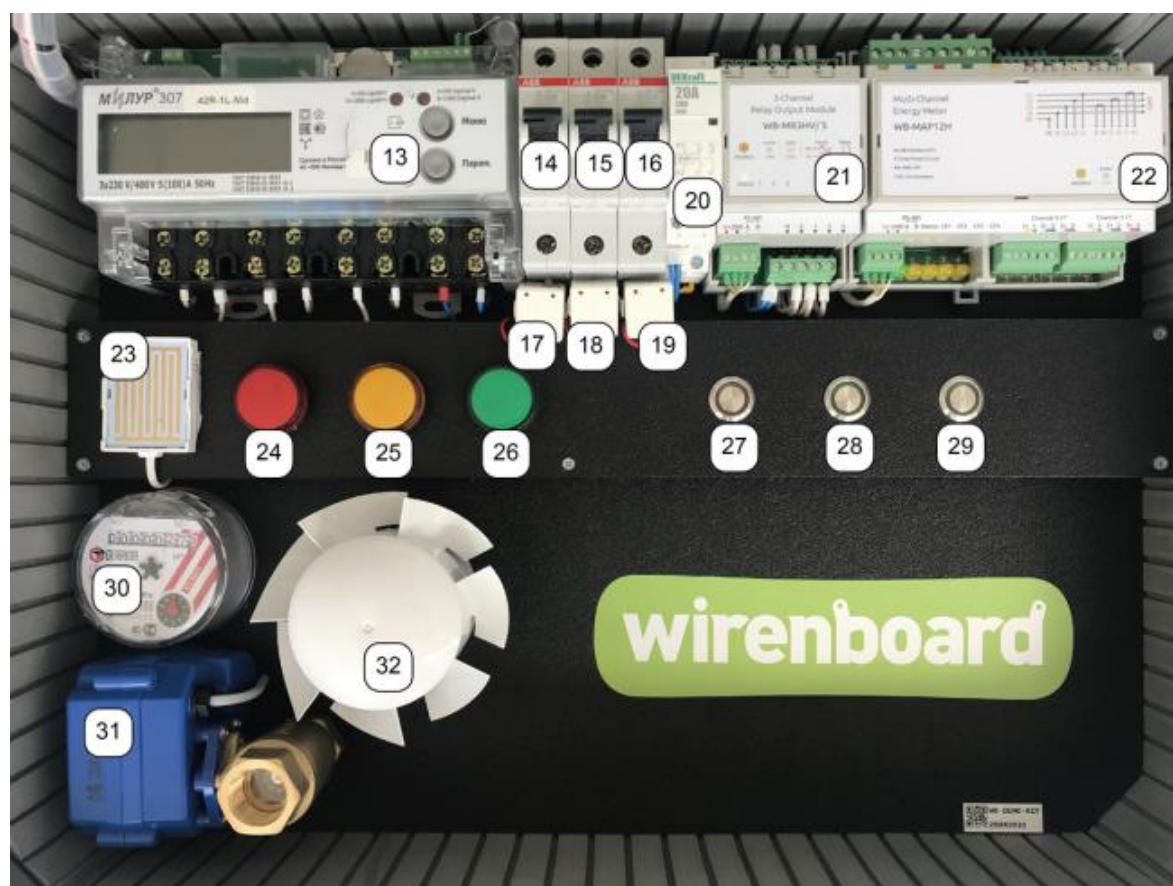


Рисунок 2 - Компоненты, расположенные на нижней крышке WB-demo-kit

Начало работы

Перед началом работы убедитесь, что компоненты не имеют видимых механических повреждений. Помните, что внутри присутствует опасное для жизни напряжение: не прикасайтесь к металлическим контактам устройств!

Подключение стенда к электросети производится преподавателем.

Убедившись, что стенд подключен к электросети, включите автоматы в порядке слева-направо, т.е. 14, 15, 16.

Включите контроллер (8), нажав на кнопку на корпусе. Когда индикатор контроллера начнет мигать зеленым светом, контроллер будет готов к работе.

Знакомство с базовым функционалом оборудования и преднастроенные сценарии

Стенд Wb-demo-kit имеет следующие категории преднастроенного функционала (сценарии):

- Энергопотребление и контроль питания
- Управление внешними силовыми устройствами
- Мониторинг качества воздуха и управление вентиляцией
- Мониторинг водоснабжения и протечек

С использованием Веб-интерфейса становится возможной демонстрация и другого функционала, например:

- Технический учет энергопотребления
- Управление ИК-устройствами (WB-MIR или WB-MSW v3)
- Контроль температуры

Энергопотребление и контроль питания

1. Наличие сетевого напряжения

Выключите автомат (14). Через несколько секунд индикатор контроллера (8) несколько раз часто моргнет красным и раздастся предупреждающий звуковой сигнал. Это означает, что сетевое напряжение отсутствует. Часть модулей продолжит питаться от встроенного в контроллер аккумуляторного модуля. Синий индикатор на блоке питания (7) погаснет примерно через 30 секунд.

2. Контроль повышенного энергопотребления

Включите вентилятор кнопкой (28). Загорится зеленая подсветка кнопки. Через некоторое время загорится желтый индикатор (25) – это означает, что счетчик (22) детектирует энергопотребление на фазе, к которой подключен вентилятор. Не касаясь лопастей вентилятора, остановите его. Через несколько секунд счетчик (22) определит повышенное энергопотребление застопоренного вентилятора и контроллер отключит его. Погаснет зеленая подсветка кнопки (28), а затем – желтый индикатор (25).

3. Контроль автоматов

Отключите автоматы (15) и (16). Через несколько секунд начнет мигать подсветка кнопок (28) и (29), что означает, что напряжение на выходах автоматов пропало. Включите автоматы снова – подсветка кнопок перестанет мигать.

Управление внешними силовыми устройствами

4. Нажмите кнопку (29). Подсветка кнопки загорится зеленым, при этом сработает контактор (20). Через некоторое время загорится индикатор (26), что означает обнаружение энергопотребления на соответствующей фазе счетчиком (22). Нажмите кнопку (29) – контактор выключится, подсветка кнопки погаснет, а через несколько секунд погаснет и индикатор энергопотребления (26).

Мониторинг качества воздуха

5. При допустимом уровне концентрации CO₂ в помещении индикатор датчика (5) мигает зеленым светом. Если несколько раз на него энергично подуть, то через 15-20 секунд индикатор начнет мигать красным, что свидетельствует о превышении концентрации CO₂. При достижении нормальной концентрации датчик снова будет мигать зеленым.

Мониторинг водоснабжения и протечек

6. Работа модуля защиты от протечек

Нажмите кнопку (27). Откроется шаровой кран (31), а счетчик (30) начнет вращаться, имитируя поток воды в системе водоснабжения. Прикоснитесь с небольшим усилием слегка влажным пальцем или смоченной салфеткой к датчику протечки (лить воду на датчик ЗАПРЕЩЕНО). Шаровой кран перекроет поток воды, счетчик перестанет вращаться, загорится красный индикатор протечки (24), подсветка кнопки (27) начнет мигать, а модуль обнаружения протечек (10) будет выдавать непрерывный звуковой сигнал (на самом модуле будет гореть индикатор Alarm). Для сброса аварийной ситуации ("протечка устранена") снова нажмите кнопку (27).

Кнопкой 27 можно открывать и закрывать шаровой кран с электроприводом, последовательно нажимая на нее.

Выключение стенда

Для выключения оборудования сначала выключите контроллер (8), после – автоматы в порядке справа-налево (т.е. 16, 15, 14). Питание к стенду подключает и отключает только преподаватель!

Задание практической работы №1

Все практические работы выполняются в подгруппе из 4 человек, состав которой не может меняться в течение семестра.

По ходу выполнения этой и следующих частей ПР необходимо фиксировать результат выполнения заданий в виде снимков экрана или фотографий, которые будут приложены к отчету.

1. Включите стенд
2. Запустите все преднастроенные сценарии (6 сценариев)
3. Выключите стенд

Дополнительное задание практической работы №1

Примечание 1. Все дополнительные задания не являются обязательными при сдаче отчета по практическим работам, однако эти задания оцениваются дополнительными баллами.

Ознакомьтесь с рынком IoT-решений и подготовьте краткое описание одной из следующих отраслей Интернета вещей и её составляющие:

1. Умный город
2. Интеллектуальные транспортные системы
3. Умное страхование
4. Умный дом
5. Умный магазин
6. Цифровое производство
7. Система маркировки и прослеживаемости товаров
8. Носимая электроника
9. Подключенный работник (Connected worker)
10. Умное ЖКХ
11. Умное сельское хозяйство
12. Цифровые двойники
13. Системы контроля и управления дорожным движением
14. Безопасный город
15. Системы контроля за обращением с отходами

Примечание 2. Варианты заданий не могут повторяться более 2 раз в одной учебной группе.

В отчете необходимо:

1. Привести сценарии применения технологий Интернета вещей в данной отрасли
2. Привести примеры применения технологий сбора, передачи и обработки данных, использующихся в описанных решениях
3. Сформулировать выводы о перспективах развития данной отрасли и возможных учебных проектов в ней.

Практическая работа №2 – Настройка удаленного подключения

Веб-интерфейс стенда

Контроллер Wiren Board имеет встроенный веб-интерфейс. Через интерфейс можно:

- следить за состоянием контроллера и подключённых устройств и управлять ими;
- подключать устройства к контроллеру;
- настраивать контроллер и обновлять его ПО;
- писать правила на встроенном движке;
- настраивать SMS и email-уведомления;
- смотреть на графике историю значений (например, температуры).

Веб-интерфейс работает непосредственно на Wiren Board. В качестве веб-сервера работает nginx, сайт взаимодействует с MQTT через WebSocket.

Чтобы зайти в веб-интерфейс, подключитесь к локальной сети, в которой находится стенд, и введите в адресную строку браузера IP-адрес контроллера (адрес указан на крышке стенда).

Работа с веб-интерфейсом

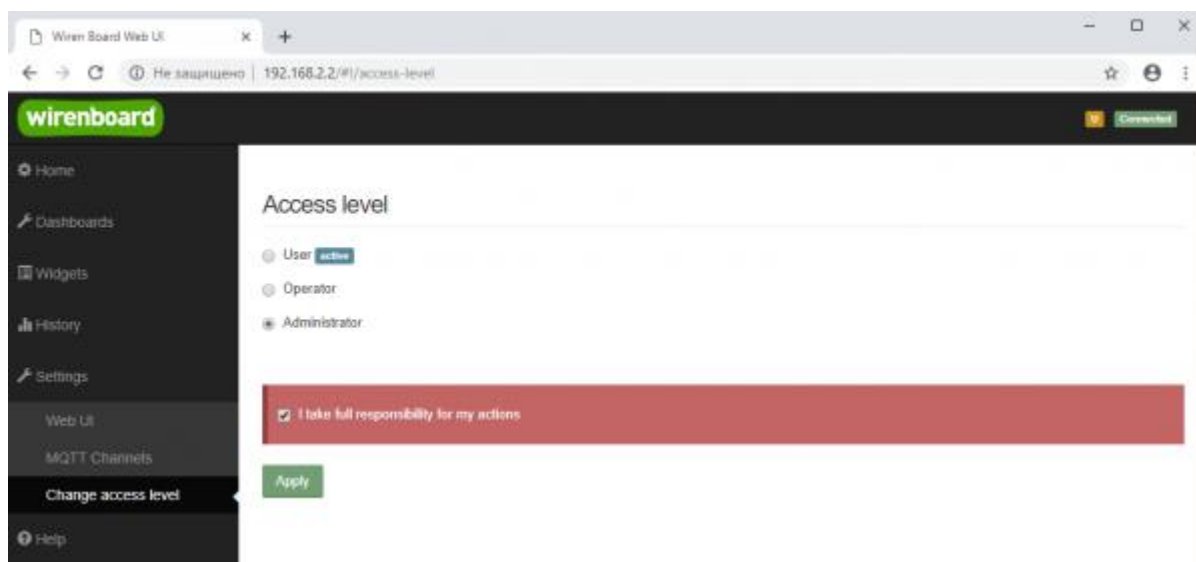


Рисунок 3 – Выбор уровня доступа

Для начала работы с веб-интерфейсом выберите уровень доступа. Для этого на вкладке **Settings** -> **Change access level** -> **Access Level** выберите один из уровней и нажмите кнопку **Apply**.

Уровни доступа призваны защитить пользователя от ошибок при регулярной работе с контроллером. Важно понимать, что это не полноценное разграничение прав, а способ защитить себя от необдуманных действий. Новых пользователей создавать нельзя.

Доступны следующие уровни:

- **User** — дашборды, виджеты, история, базовые настройки.
- **Operator** — права уровня **User** и раздел **Devices**.
- **Administrator** — полный доступ ко всем функциям.

В дальнейшем предполагается, что все действия выполняются пользователем **Administrator**.

Чтобы получить уровень доступа **Administrator**, на вкладке **Access Level** выберите опцию **Administrator**, подтвердите выбор, установив флажок в поле **"I take full responsibility for my actions"**, и нажмите кнопку **Apply** (как показано на рисунке "Выбор уровня доступа").

Разделы интерфейса

Home (Главная страница)

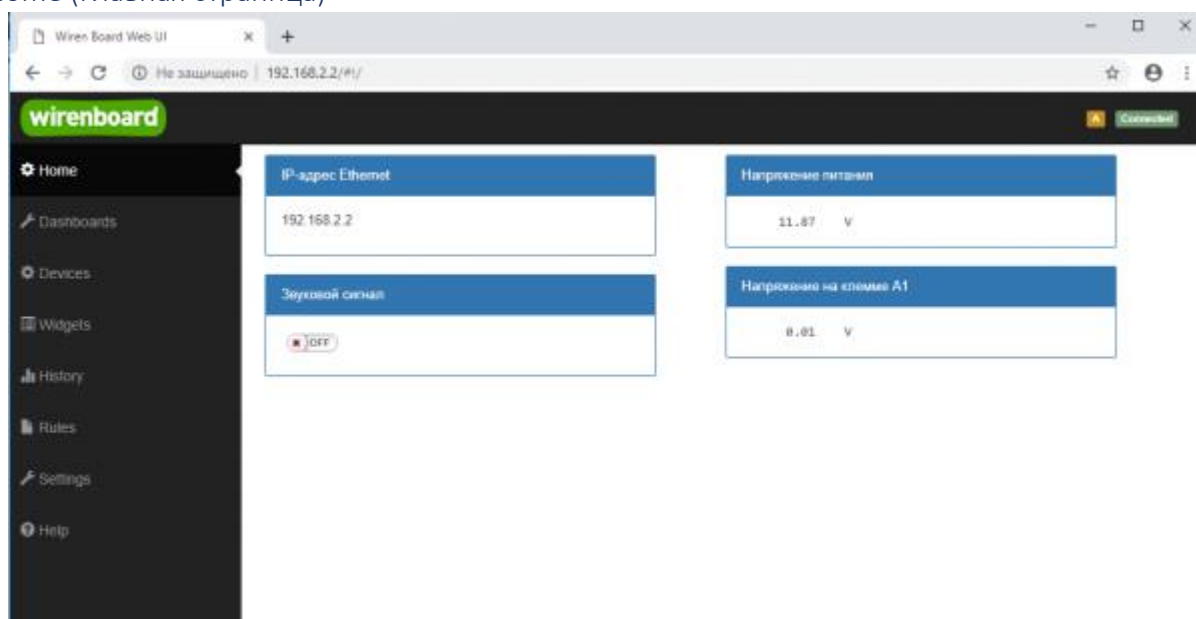


Рисунок 4 – Home - главная страница

Это главная страница пользователя. На неё выводятся элементы интерфейса - так называемые "виджеты" (widget). Это могут быть показания датчиков (например, датчика температуры), кнопки включения света, управления подключёнными реле. Набор виджетов на главной странице полностью настраивается пользователем в меню **Settings -> Web UI -> Common Info**, где можно выбрать панель, которая будет отображаться во вкладке **Home** по умолчанию.

Dashboards (Панели)

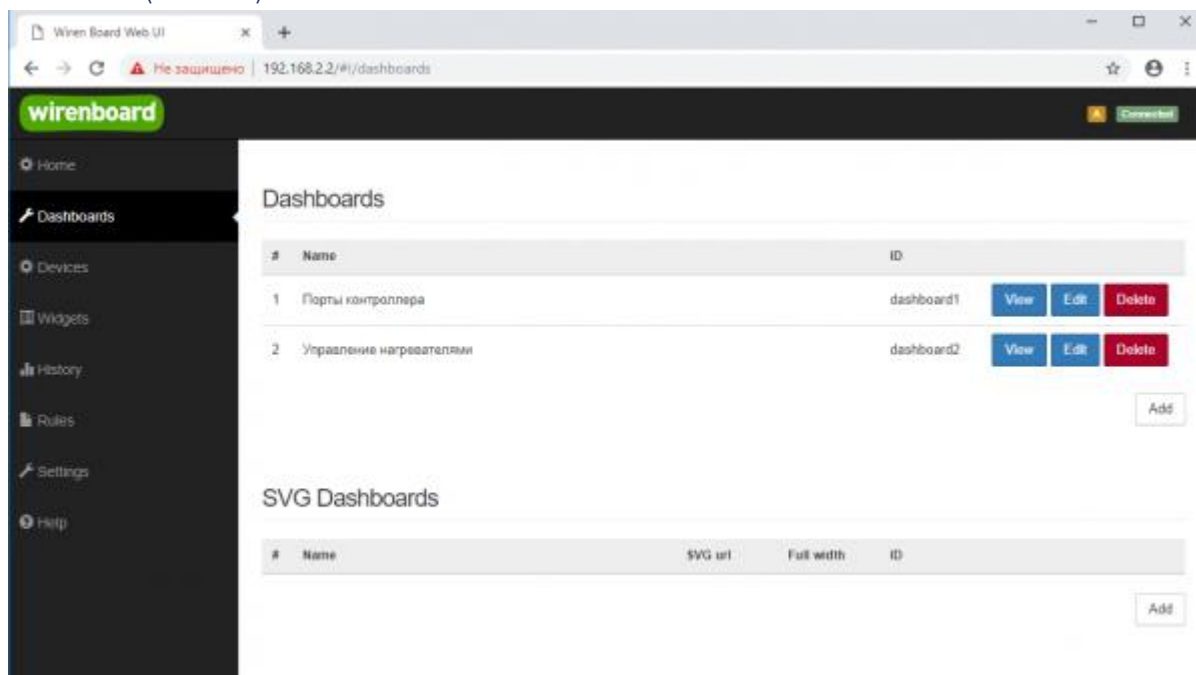


Рисунок 5 – Dashboards - список панелей

Виджеты со схожим назначением можно группировать в панели, где на одном экране находятся все необходимые кнопки, настройки и показатели датчиков. Например, можно

объединить виджеты включения подогрева, кондиционера, отображения температуры и влажности. В разделе **Dashboards** можно увидеть все созданные панели. Раздел Home тоже отображает одну из панелей, выбранную в настройках.

Дважды щелкнув по вкладке **Dashboards** на боковой панели, можно раскрыть список всех панелей, созданных в веб-интерфейсе.

Кроме текстовых панелей с виджетами, можно создавать [интерактивные SVG-панели](#) (SVG Dashboards).

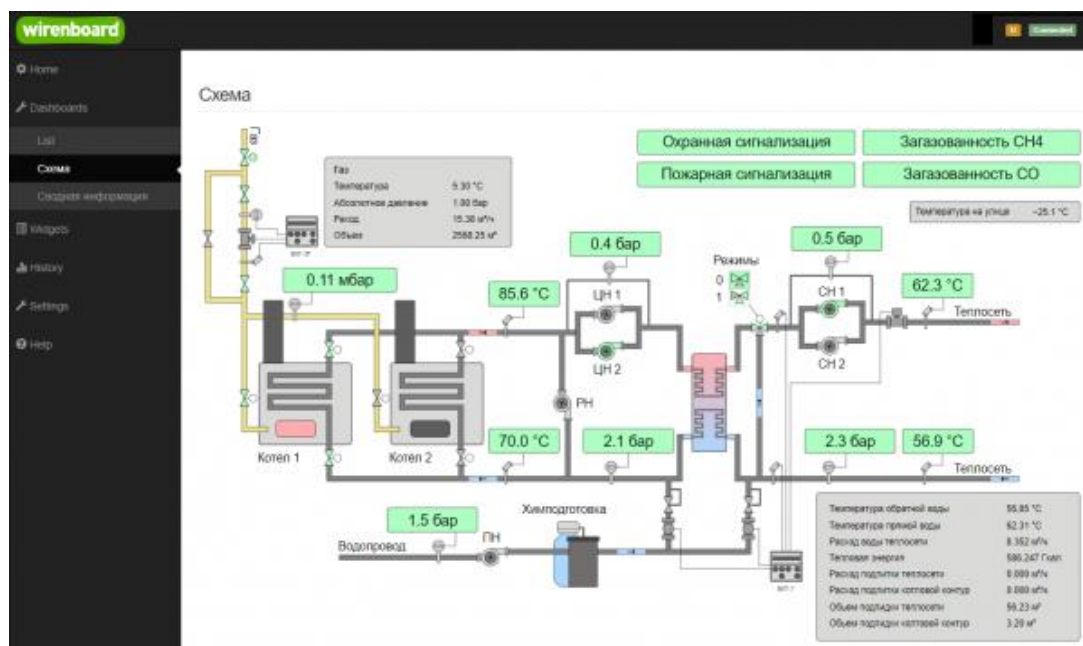


Рисунок 6 – Пример SVG-панели

Devices (Устройства)

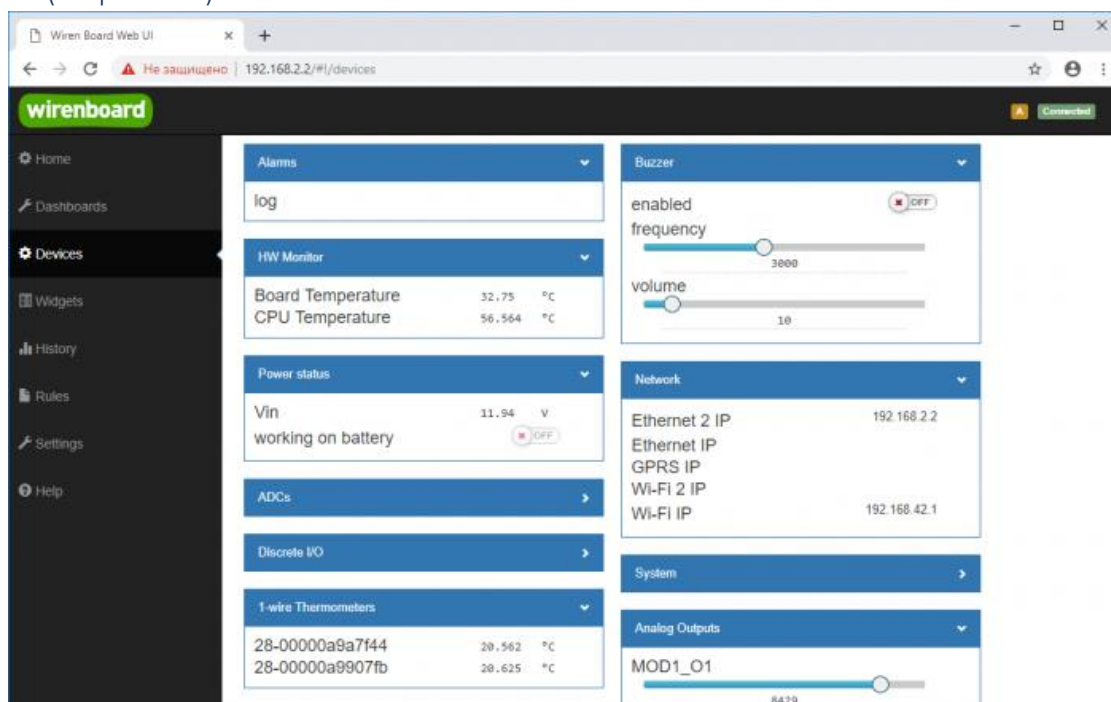


Рисунок 7 – Devices - список всех аппаратных функций контроллера и подключённых устройств

На странице **Devices** отображаются все аппаратные возможности контроллера (состояние входов, выходов, напряжение питания), а также подключённых датчиков и устройств. Если вы подключили к контроллеру внешний модуль, все его меняющиеся значения будут отображены тут.

Каждый элемент устройства (показание значения напряжения, сетевой адрес, кнопка управления реле, флажок состояния входа и т.п.) -- называется "контрол". Несколько контролов могут быть объединены в один виджет. Подробнее смотрите в разделе Widgets (Виджеты).

Подключаемые устройства (Modbus-модули, боковые и внутренние модули) **не** определяются контроллером автоматически. Чтобы на этой странице появились аппаратные возможности подключённых устройств (например, внешних модулей реле), сначала нужно настроить их через раздел Configs (Конфигурирование).

Удалить отключенные/неработающие устройства из веб-интерфейса можно с помощью кнопки **Delete** в виде значка с изображением мусорной корзины, в верхней строке плитки устройства. Кнопка появляется, когда указатель мыши находится над плиткой устройства.

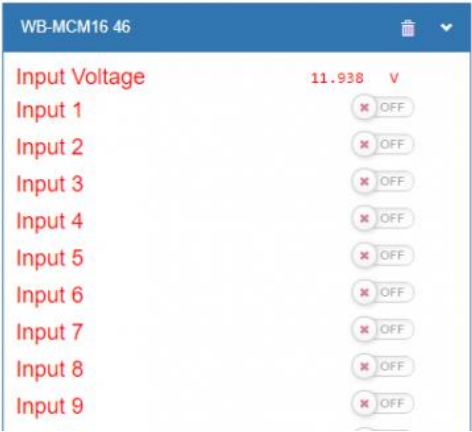


Рисунок 8 – Удаление отключенного устройства

Widgets (Виджеты)

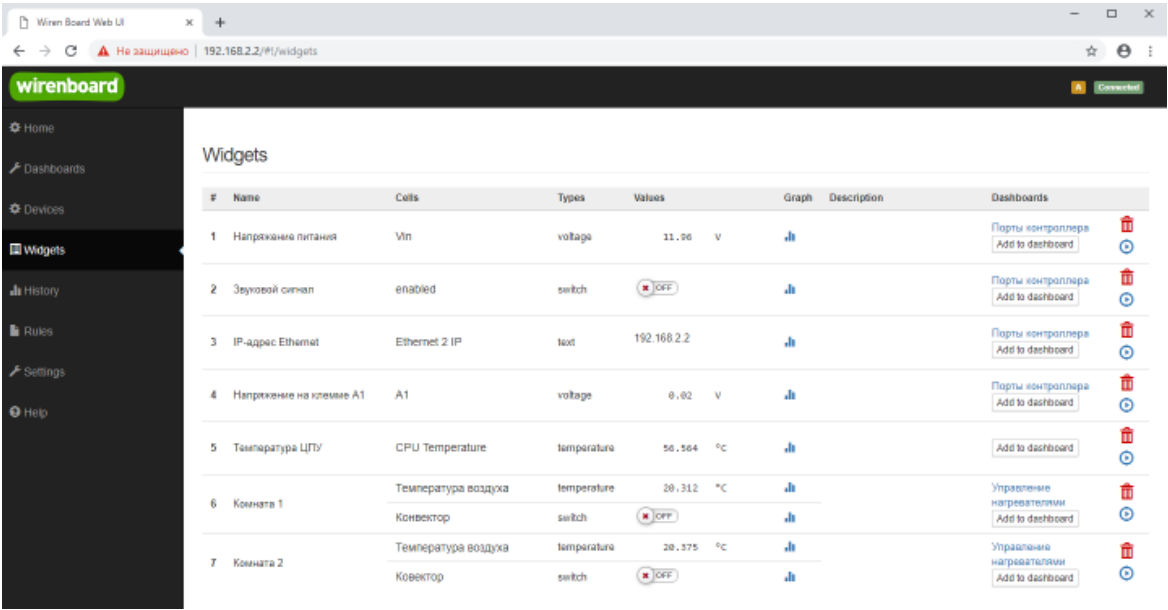


Рисунок 9 – Widgets - страница управления виджетами

Виджеты - комбинированные элементы интерфейса контроллера, включающие в себя набор контролов, то есть аппаратных параметров контроллера и подключённых к нему устройств - тех, что отображаются на странице Devices (Устройства).

На странице Widgets представлен список всех виджетов, созданных в системе. Сами виджеты создаются в настройках панелей, на этой странице ими можно только управлять: просматривать, удалять и добавлять к существующим панелям-дашбордам.

Пример создания виджетов.

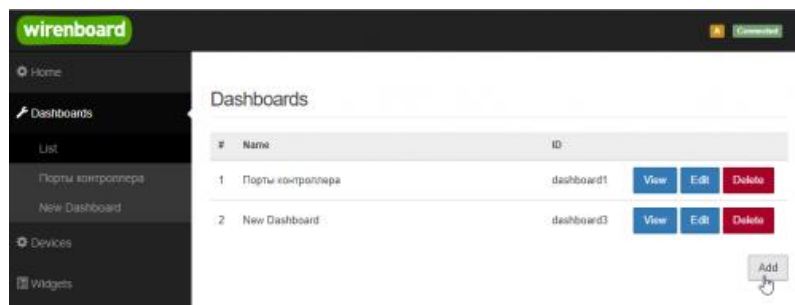


Рисунок 10 – Создать новую панель

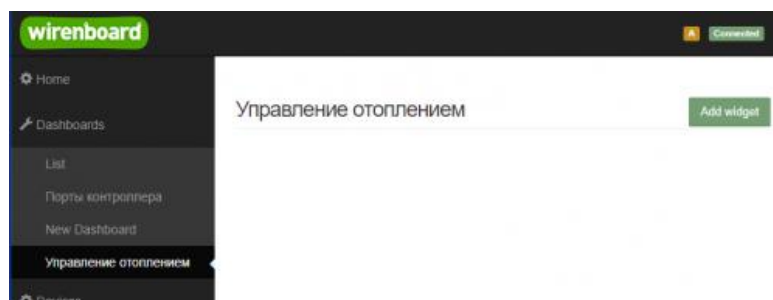


Рисунок 11 – Создать новый виджет

Для примера создадим два виджета с показаниями температуры и переключателями управления отопительными конвекторами для двух комнат "Комната 1" и "Комната 2".

1. На боковой панели щелкнем на вкладке **Dashboards**, в раскрывшемся списке выберем элемент **List** и на открывшейся странице нажмем кнопку **Add**.
2. В поле **Name** напишем название новой панели, "Управление отоплением" и нажмем кнопку **Save**.
3. В списке на странице **Dashboards** щелкнем по кнопке **View** напротив новой панели "Управление отоплением".
4. В открывшемся окне с названием панели щелкнем по кнопке **Add widget** в правом верхнем углу окна (см. Рис. "Создать новый виджет").
5. В заголовке виджета укажем название, в нашем случае "Комната 1", в списке **Add control...** выберем контрол, соответствующий термометру в первой комнате, еще раз в этом списке выберем реле, которое будет включать нагреватель.
6. В поле **Name** виджета можно задать осмысленные названия для контролов, например: "Температура" и "Обогреватель". Снимите флажок **Compact mode**, чтобы эти названия контролов отображались в виджете.
7. В поле **Widget description** можно написать назначение виджета.
8. Аналогично создадим виджет для управления отоплением в комнате 2.
9. Для внесения изменений подведите курсор к заголовку виджета и нажмите кнопку **Edit widget**, внесите изменения и нажмите кнопку **Save**.

Управление отоплением

Комната 1

☒ Compact mode

Find controls of type any

Add control...

- WB-MR3 27 / Input 0
- WB-MR3 27 / Input 1 counter
- WB-MR3 27 / Input 2 counter
- WB-MR3 27 / Input 3 counter
- WB-MR3 27 / Input 0 counter
- WB-MR3 27 / Supply voltage
- 1-wire Thermometers / 28-00000a9a7f44**
- 1-wire Thermometers / 28-00000a9907fb
- Rule Engine Settings / Rule debugging

Рисунок 13 – Добавить новый контрол в виджет

Управление отоплением

Комната 1

☒ Compact mode

Actions	Name	Type
	Температура	temperature
<input type="checkbox"/> Invert	Обогреватель	switch

Find controls of type any

Add control...

Управление обогревателем в комнате 1

Рисунок 14 – Пример создания виджета

Управление отоплением

Комната 1

Температура 21.5 °C

Обогреватель ☒

Комната 2

Температура 21.562 °C

Обогреватель ☒

Рисунок 15 – Пример готовых виджетов на панели

History (История показаний)

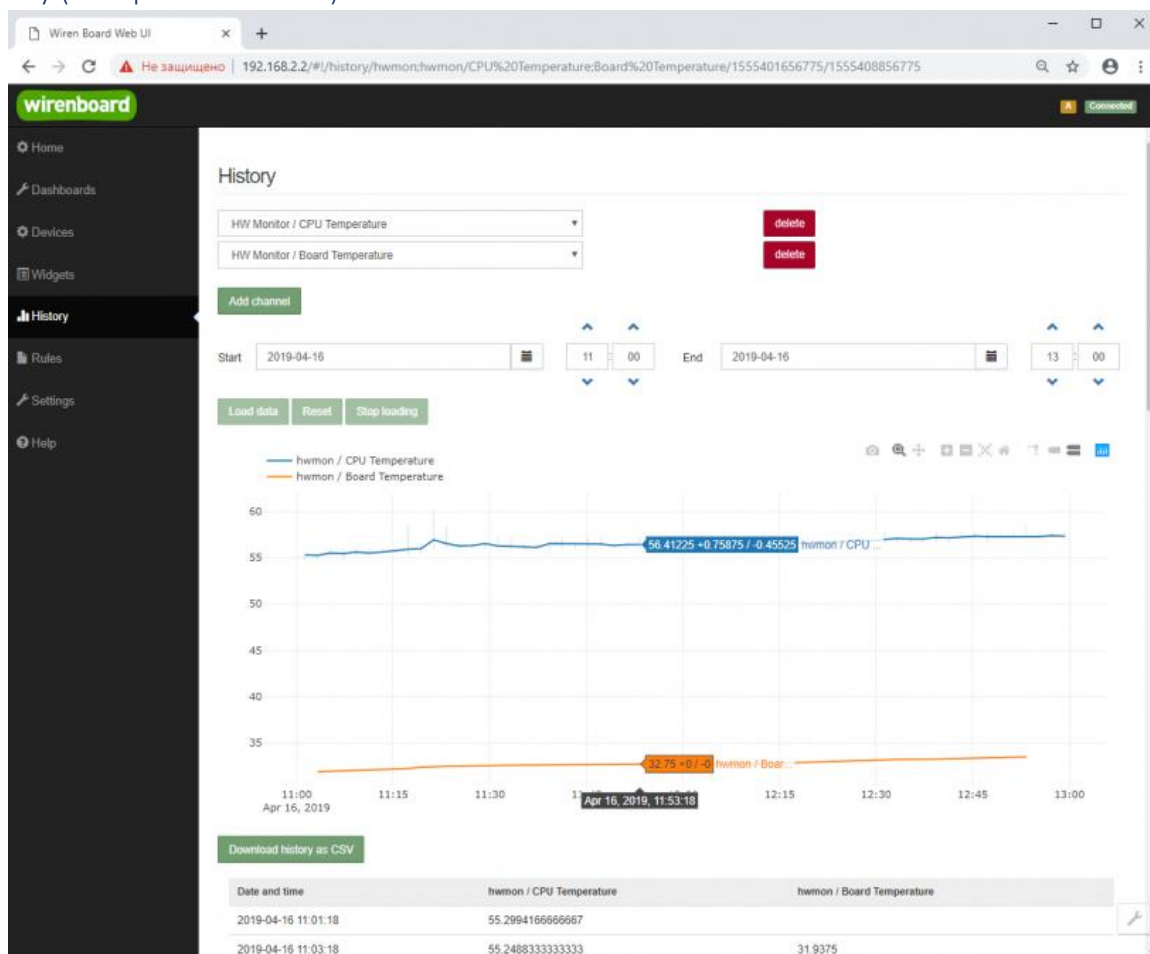


Рисунок 16 – Пример отображения исторических данных

На странице *History* можно просмотреть историю изменения значений аппаратных ресурсов (например, датчиков температуры, напряжения, показаний счётчиков). История представляется одновременно в виде графика и таблицы значений с метками времени.

- Возможности просмотра исторических данных:
- Указание интервала времени для отображения данных
- Добавление и удаление нескольких показателей (кнопки Add channel и delete) на график
- Просмотр данных в виде графика и в виде таблицы
- Загрузка данных за выбранный период в csv-формате.

При наведении указателя мыши на область изображения становятся доступными дополнительные функции (кнопки в верхней правой части графика):

- Сохранение графика в формате .png
- Переключение между режимами Zoom (увеличения/уменьшения отрезка данных и масштаба с помощью выделения нужной области указателем мыши) и панорамирования Pan (перемещения области видимости с зажатой левой кнопкой мыши)

- Уменьшение и увеличение отображаемого временного интервала (Zoom in и Zoom out)
- Автоматический выбор масштаба графика по обеим осям
- Возвращение масштаба осей к исходному
- Включение/выключение указателя координат

Задание практической работы №2

1. Включите стенд
2. Подключитесь к стенду через веб интерфейс
3. Запустите преднастроенные сценарии еще раз, отследить изменение параметров в веб-интерфейсе
4. Зафиксируйте изменения параметров в веб-интерфейсе, возникшие при выполнении сценариев
5. Выключите стенд

Дополнительное задание практической работы №2

Придумайте и утвердите с преподавателем тему проекта – IoT системы, разработкой которой вы будете заниматься на протяжении всего семестра.

Разработка IoT системы и защита проекта происходит в той же подгруппе, в которой и защита обязательных практических работ.

Приведите описание выбранной темы проекта. Описание должно содержать:

1. Название проекта
2. Отрасль Интернета вещей, к которой относится тема проекта
3. Актуальность проекта
4. Предварительный перечень оборудования, необходимого для работы IoT системы
5. Какие данные возможно собрать от физических «вещей» для данной области?

Практическая работа №3 – Обработка событий в системах Интернета вещей

Правила-скрипты в WirenBoard

Для контроллера WirenBoard можно писать правила, например: "Если температура датчика меньше 18°C, включи нагреватель".

Правила можно создать и редактировать на странице **Rules**. Правила пишутся на простом языке, похожем на JavaScript и позволяют создавать правила ("включай свет с 10:00 до 18:00") или виртуальные устройства (например, кнопка в интерфейсе, которая включает и отключает всё освещение в здании вместе).

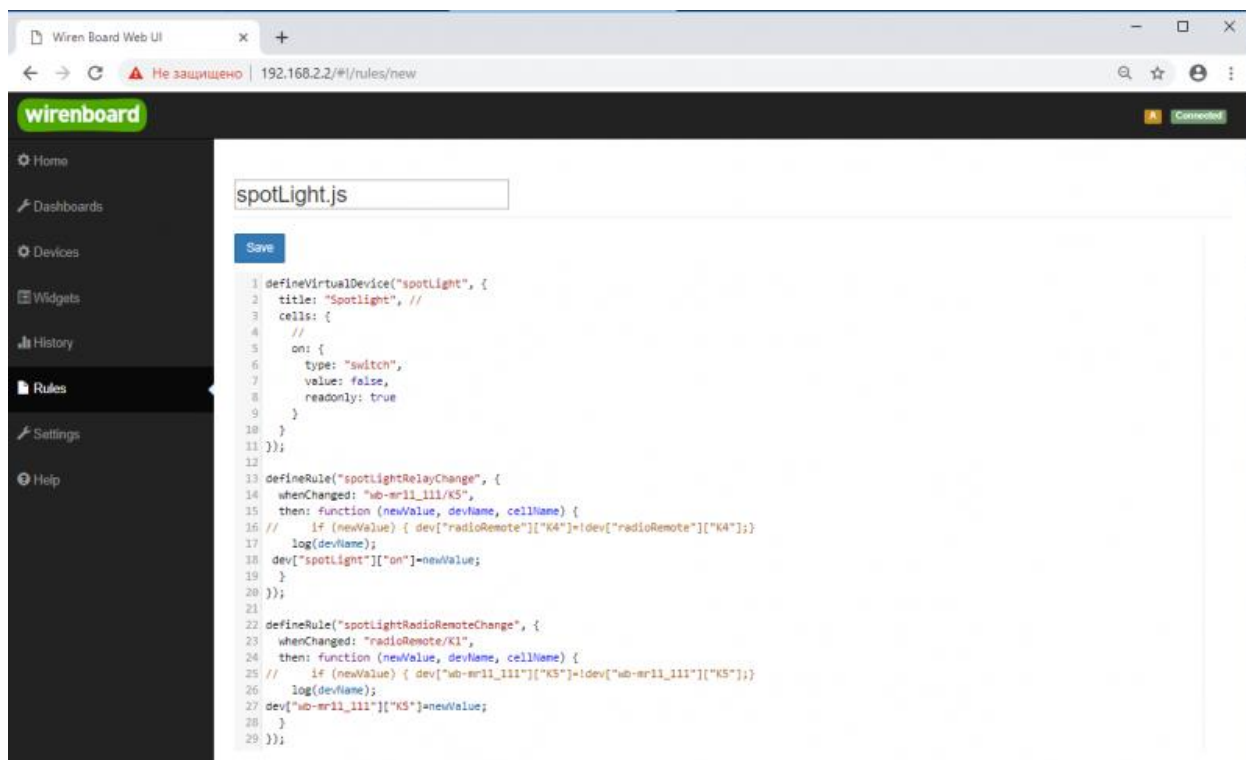


Рисунок 17 – Скрипт, открытый для просмотра и редактирования

Создание и редактирование правил

Список файлов с правилами находится на странице *Rules* веб-интерфейса. Для редактирования правила нажмите на название файла.

Для создания нового правила, нажмите на *New...*, вверху введите название (латинские буквы и цифры, в качестве расширения укажите *.js*), в основное поле введите текст скрипта, и нажмите *Save* вверху.

Правило начинает сразу работать после сохранения, если в нём нет ошибок.

В одном файле можно хранить неограниченное количество правил. Но обычно в одном файле хранятся правила с близкими функциями.

Пример правила

Для начала разберём простое правило "при превышении температуры - выключи обогреватель". Температуру получаем с датчика 1-Wire, обогреватель подключён к Реле 1 внешнего релейного модуля WB-MRM2.

```
defineRule("heater_control", { //название правила - "контроль обогревателя",
    может быть произвольным

    whenChanged: "wb-w1/28-0115a48fcfff", //при изменении состояния датчика 1-
    Wire с идентификатором 28-0115a48fcfff

    then: function (newValue, devName, cellName) { //выполняй следующие
    действия

        if ( newValue > 30) { //если температура датчика больше 30 градусов

            dev["wb-mrm2_130"]["Relay 1"] = false; //установи Реле 1 модуля WB-MRM2
            с адресом 130 в состояние "выключено"

        } else {
```

```

dev["wb-mrm2_130"]["Relay 1"] = true; //установи Реле 1 модуля WB-MRM2
с адресом 130 в состояние "включено"

    }

}

});

```

1. Первая строка - кодовое слово *defineRule* и название правила
2. Вторая строка - кодовое слово для определения, когда выполняется правило, - *whenChanged* - "при изменении параметра", далее название параметра, при изменении которого запустится правило - температура с датчика 1-Wire. Название параметра записывается в виде "Device/Control", где названия *Device* и *Control* для каждого параметра можно найти на странице *Settings* веб-интерфейса, в таблице *MQTT Channels*.
3. Третья строка - начало функции, которая будет исполняться
4. Четвертая строка - условие - "если значение температуры больше порогового, то ...". Значение параметра записывается в виде *dev[Device][Control]* - заметьте, оно отличается от вида записи параметра, при изменении которого запускается правило, потому что там речь идёт о *параметре*, а здесь - о *значении* того же параметра.
5. После выставляются значения для реле в каждом случае - *false* - "выключено", *true* - "включено". Названия *Device* и *Control* для реле смотрим всё в той же таблице *MQTT Channels*, на странице *Settings* веб-интерфейса.

Ознакомиться с другими примерами правил можно здесь:
https://wirenboard.com/wiki/Rule_Examples

Типы правил *defineRule*

whenChanged

Правило срабатывает при любых изменениях значений параметров или функций В примере кнопка подключена ко входу A1 контроллера. При нажатии на кнопку срабатывает реле 1 устройства MRM2-mini, при отжатии реле возвращается в исходное состояние.

```

defineRule("test_rule", { //имя правила test_rule
    whenChanged: "wb-gpio/A1_IN",
    then: function (newValue, devName, cellName) {
        dev["wb-mrm2-mini_2"]["Relay 1"] = newValue;
    }
});

```

asSoonAs

Правила, задаваемые при помощи *asSoonAs*, срабатывают в случае, когда значение, возвращаемое функцией, заданной в *asSoonAs*, становится истинным при том, что при предыдущем просмотре данного правила оно было ложным.

```

defineRule({
    asSoonAs: function () {

```

```

        return dev["wb-gpio"]["A2_IN"];
    },
    then: function (newValue, devName, cellName) {
        dev["wb-mrm2-mini_2"]["Relay 2"] = true;
    }
});

```

При нажатии на кнопку, подключенную к входу A2, включаем реле, а при нажатии на кнопку, подключенную к входу A3 — выключаем.

when

Правила, задаваемые при помощи when срабатывают при каждом просмотре, при котором функция, заданная в when, возвращает истинное значение. При срабатывании правила выполняется функция, заданная в свойстве when.

```

defineVirtualDevice("test_button2", {
    title: "test_relay2",
    cells: {
        "switch_on": {
            type: "pushbutton",
            value: false,
        },
        "switch_off": {
            type: "pushbutton",
            value: false,
        }
    }
});

defineRule({
    when: function () {
        return dev["test_button2"]["switch_on"];
    },
    then: function (newValue, devName, cellName) {
        dev["wb-mrm2-mini_2"]["Relay 2"] = true;
    }
});

defineRule({
    when: function () {
        return dev["test_button2"]["switch_off"];
    }
});

```

```

    },
    then: function (newValue, devName, cellName) {
        dev["wb-mrm2-mini_2"]["Relay 2"] = false;
    }
});

```

При нажатии на кнопку switch_on, включаем реле, а при нажатии на кнопку,switch_off — выключаем.

cron-правила

Отдельный тип правил - cron-правила. Такие правила задаются следующим образом:

```

defineRule("crontest_hourly", {
    when: cron("@hourly"),
    then: function () {
        log("@hourly rule fired");
    }
})

```

Вместо @hourly здесь можно задать любое выражение, допустимое в стандартном crontab, например, 00 00 20 * * (секунды минуты часы выполнять правило каждый день в 20:00). Помимо стандартных выражений допускается использование ряда расширений, см. описание формата выражений используемой cron-библиотеки.

Массивы

Массив определяется короткой записью:

```

fruits = ["One", "Two", "Three"];

```

Обращение к элементу:

```

fruits[0]; //One

```

Таймеры

setTimeout();

С помощью данного таймера можно отсрочить выполнение правила на определенное время. В примере после нажатия кнопки включается пищалка и затем выключается спустя 2 секунды.

```

defineVirtualDevice("test_buzzer", {
    title: "Test Buzzer",
    cells: {
        enabled: {
            type: "pushbutton",
            value: false

```

```

    }
  }
});

defineRule({
  whenChanged: "test_buzzer/enabled",
  then: function (newValue, devName, cellName) {
    dev["buzzer"]["enabled"] = true;
    setTimeout(function () {
      dev["buzzer"]["enabled"] = false;
    }, 2000);
  }
});

```

setInterval();

```

defineVirtualDevice("test_buzzer", {
  title: "Test Buzzer",
  cells: {
    enabled: {
      type: "pushbutton",
      value: false
    }
  }
});

var test_interval = null;

defineRule({
  whenChanged: "test_buzzer/enabled",
  then: function (newValue, devName, cellName) {
    var n = 0;
    if (dev["test_buzzer"]["enabled"]){
      test_interval = setInterval(function () {
        dev["buzzer"]["enabled"] = !dev["buzzer"]["enabled"];
        n = n+1;
        if (n >= 10){
          clearInterval(test_interval);
        }
      }
    }
  }
});

```



```

    }, 500);

}

});

```

startTimer();

Запускает однократный таймер. При срабатывании таймера происходит просмотр правил, при этом `timers.<name>.firing` для этого таймера становится истинным на время этого просмотра.

```

defineVirtualDevice("test_buzzer", {
  title: "Test Buzzer",
  cells: {
    enabled: {
      type: "switch",
      value: false
    }
  }
});

defineRule("1",{
  asSoonAs: function () {
    return dev["test_buzzer"]["enabled"] ;
  },
  then: function () {
    startTimer("one_second", 1000);
    dev["buzzer"]["enabled"] = true; //включаем пищалку
  }
});

defineRule("2",{
  when: function () { return timers.one_second.firing; },
  then: function () {
    dev["buzzer"]["enabled"] = false; //выключаем пищалку
    dev["test_buzzer"]["enabled"] = false;
  }
});

```

startTicker();

Запускает периодический таймер с указанным интервалом. В примере правило 1 запускает таймер с интервалом 1 сек. Вызывая срабатывание правила 2. Метод stop() приводит к остановке таймера.

```
defineVirtualDevice("test_buzzer", {
  title: "Test Buzzer",
  cells: {
    enabled: {
      type: "switch",
      value: false
    }
  }
});

defineRule("1",{
  asSoonAs: function () {
    return dev["test_buzzer"]["enabled"] ;
  },
  then: function () {
    startTicker("one_second", 1000);
  }
});

defineRule("2",{
  when: function () { return timers.one_second.firing; },
  then: function () {
    dev["buzzer"]["enabled"] = !dev["buzzer"]["enabled"];
    if (dev["test_buzzer"]["enabled"] == false){
      timers.one_second.stop();
    }
  }
});
```

Виртуальные устройства defineVirtualDevice

Правила делятся на два типа: непосредственно правила (defineRule) и виртуальные устройства (defineVirtualDevice).

Виртуальные устройства — это появляющиеся в веб-интерфейсе новые элементы управления - например, кнопка-выключатель, которая на самом деле выключает два устройства одновременно. Она не привязана напрямую ни к какому физическому устройству, а действия при её нажатии определяются написанным вами скриптом. При

написании скрипта вы можете создать виртуальное устройство для включения/выключения тех или иных управляющих алгоритмов и установки их параметров.

```
defineVirtualDevice("wb-1", {
  title: "Отопление",
  cells: {
    "Температура": {
      type: "range",
      max: 24,
      value: 20,
    },
    "Вкл.": {
      type: "switch",
      value: false,
    }
  }
});
```

Типы устройств (type):

switch — переключатель. Может принимать значения true или false. По-умолчанию доступен для изменения пользователем.

wo-switch — переключатель, аналог switch. Может принимать значения true или false. По-умолчанию не доступен для изменения пользователем.

pushbutton — кнопка. Может принимать значения true. По-умолчанию доступна для нажатия.

range — ползунок. Может принимать значения от 0 до max. По-умолчанию доступен для изменения пользователем.

rgb — специальный тип для задания цвета. Кодировка 3 числами от 0 до 255, разделенными точкой с запятой. Например „255;0;0“ задает красный цвет. По-умолчанию доступен для изменения пользователем.

alarm — индикатор. Может принимать значения true или false.

text — текстовое поле. По-умолчанию не доступно для редактирования пользователем.

value — значение с плавающей точкой. По-умолчанию не доступно для редактирования пользователем.

Так же существуют еще 14 специальных типов. Все они аналогичны value, но имеют соответствующие подписи в интерфейсе.

Type	meta/type	units	value format
Temperature	temperature	°C	float

Relative humidity	rel_humidity	%, RH	float, 0 - 100
Atmospheric pressure	atmospheric_pressure	millibar (100 Pa)	float
Precipitation rate (rainfall rate)	rainfall	mm per hour	float
Wind speed	wind_speed	m/s	float
Power	power	watt	float
Power consumption	power_consumption	kWh	float
Voltage	voltage	volts	float
Water flow	water_flow	m ³ / hour	float
Water total consumption	water_consumption	m ³	float
Resistance	resistance	resistance	float
Gas concentration	concentration	ppm	float (unsigned)
Heat power	heat_power	Gcal / hour	float
Heat energy	heat_energy	Gcal	float

value:

Обязательное поле. При создании устройства в первый раз его значение будет установлено в значение по-умолчанию. В дальнейшем значения сохраняются в специальное хранилище в постоянной памяти и восстанавливаются при загрузке сценария.

forceDefault:

Если необходимо каждый раз при перезагрузке скрипта восстанавливать строго определённое значение (т.е. не восстанавливать предыдущее сохранённое), нужно добавить в описание контрола поле **forceDefault: true**

max:

Для параметра типа **range** может задавать его максимально допустимое значение.

readonly:

Когда задано истинное значение — устройство становится не доступным для редактирования пользователем. Если надо предоставить пользователю возможность редактировать значение, следует добавить в описание **readonly: false**

Сообщения в лог

В зависимости от функции сообщение классифицируется как отладочное (debug), информационное (info), предупреждение (warning) или сообщение об ошибке (error).

```

defineVirtualDevice("test_buzzer", {
  title: "Test Buzzer",
  cells: {
    enabled: {
      type: "pushbutton",
      value: false
    }
  }
});

defineRule({
  whenChanged: "test_buzzer/enabled",
  then: function (newValue, devName, cellName) {
    log.info('info');
    log.debug('debug');
    log.error('error');
    log.warning('warning');
    log('log!'); //сокращение для log.info();
    debug('deb!'); //сокращение для log.debug();
  }
});

```

Выполнение произвольной команды runShellCommand();

```

defineVirtualDevice("test_button", {
  title: "Test Button",
  cells: {
    enabled: {
      type: "pushbutton",
      value: false
    }
  }
});

defineRule({
  whenChanged: "test_button/enabled",
  then: function (newValue, devName, cellName) {
    runShellCommand("mosquitto_pub -t
'/devices/test_button/controls/enabled/meta/readonly' -r -m 1");
    setTimeout(function () {

```

```

        runShellCommand("mosquitto_pub -t
'/devices/test_button/controls/enabled/meta/readonly' -r -m 0");
    }, 5000);
}
});

```

Задание практической работы №3

1. Разработайте сценарии для обработки событий согласно вариантам и приведите в отчете его листинг с комментариями:

№ варианта	Сценарии
1	<ol style="list-style-type: none"> 1. Включение и выключение воды по датчику движения 2. Включение и выключение диодной ленты по кнопке
2	<ol style="list-style-type: none"> 1. Включение и выключения вентилятора по концентрации CO₂ 2. Включение, выключение и изменение звукового сигнала по кнопке (или нескольким кнопкам)
3	<ol style="list-style-type: none"> 1. Изменение цвета диодной ленты по концентрации CO₂ (зеленый цвет – концентрация в норме, красный – повышена) 2. Включение и выключения световых индикаторов по кнопкам
4	<ol style="list-style-type: none"> 1. Включение и выключение вентилятора по температуре 2. Открытие и закрытие шарового крана при одновременном нажатии кнопок 25 и 26
5	<ol style="list-style-type: none"> 1. Включение и выключение вентилятора по датчику движения 2. Включение и выключения индикации зеленым и красным светом комбинированного датчика 5 по кнопкам
6	<ol style="list-style-type: none"> 1. Включение и выключение звукового сигнала по датчику силы тока 2. Включение и изменение цвета диодной ленты по кнопкам
7	<ol style="list-style-type: none"> 1. Включение и выключение диодной ленты по датчику силы тока 2. Включение и выключение вентилятора при одновременном нажатии кнопок 27 и 29

2. Не забудьте выгрузить сценарий на личное устройство, чтобы избежать его потери

Дополнительное задание практической работы №3

1. Уточните портрет пользователя проекта
2. В каких ситуациях и каким образом он взаимодействует с оборудованием, сайтом, приложениями?
3. Отрадите сценарии использования проекта словесно и посредством UML use-case диаграммы. Например:



Рисунок 18 – Пример UML use-case диаграммы

Практическая работа №4 – Основы электротехники в системах Интернета вещей

Задание практической работы №4

1. Отобразить события из вариантов Практической работы №3 на UML диаграммах последовательности (Sequence diagram). Диаграммы последовательности должны включать в себя полное взаимодействие всех составляющих компонентов: от пользователя до исполнительных устройств.
2. Отобразить (выделить цветом) на схеме (Рис. 19) подключение компонентов, задействованных в обработке событий.

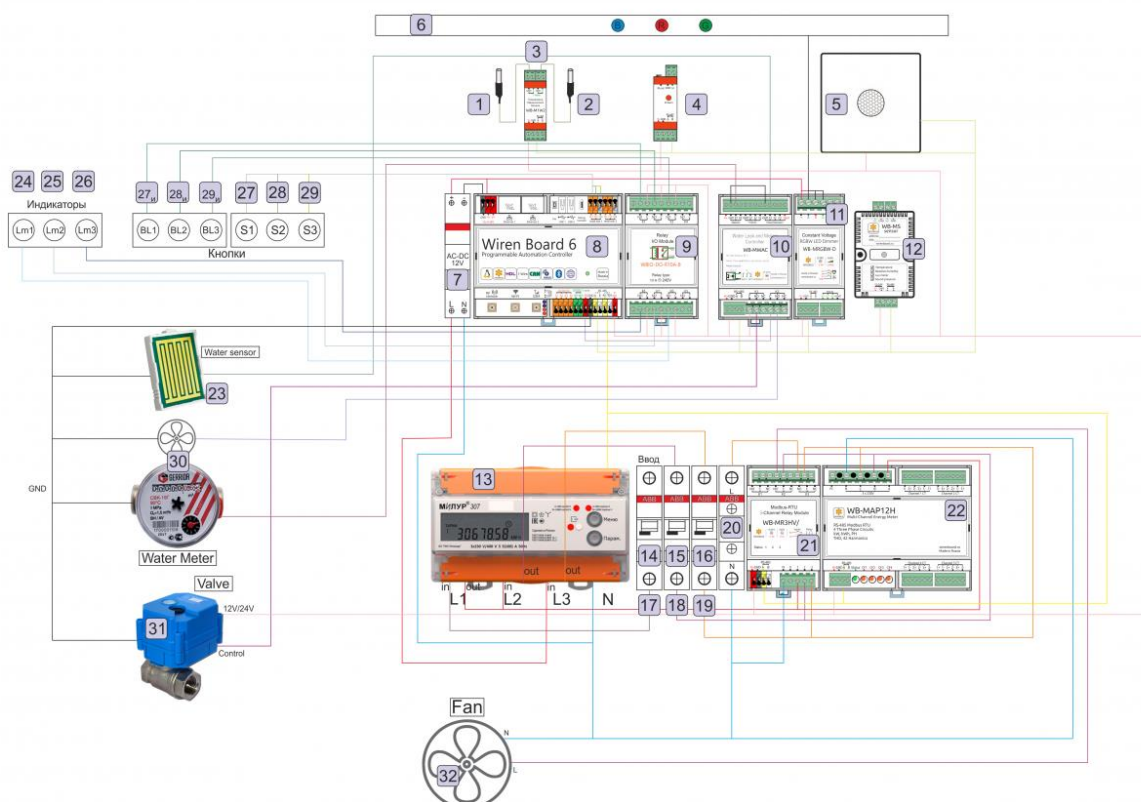


Рисунок 19 – Схема подключения устройств в демонстрационном комплекте. Цвета соединений на схеме условные и не совпадают с цветами проводов в устройстве.

Дополнительное задание практической работы №4

1. Сформируйте требования к проекту
2. Определите архитектуру проекта и отобразите в виде схемы. Архитектура проекта должна включать в себя следующие подсистемы: физическое устройство (датчики, актуаторы, контроллеры или их совокупность), облако или IoT платформа, Пользовательский интерфейс пользователя (веб-интерфейс, десктоп или мобильное приложение)
3. Составить список оборудования для реализации проекта

Требования к отчету по ПР №1-4:

По итогу выполнения практических работ №1-4 необходимо оформить единый отчёт, включающий:

1. Титульный лист;
2. Оглавление;
3. Процесс и результат выполнения основных заданий (скриншоты и описание) ПР №1;
4. Процесс и результат выполнения основных заданий (скриншоты и описание) ПР №2;
5. Процесс и результат выполнения основных заданий (скриншоты и описание) ПР №3;
6. Процесс и результат выполнения основных заданий (скриншоты и описание) ПР №4;
7. Отчет о проекте – результаты выполнения всех дополнительных заданий ПР №1-4;
8. Выводы о проделанной работе.

Отчет по практическим работам необходимо загрузить в СДО (в случае каких-либо технических проблем отчет необходимо выслать на почту преподавателя в домене *mirea.ru*)

Литература для изучения:

1. Документация на чемодан: https://wirenboard.com/wiki/Wb-demo-kit_v.2
2. Веб-интерфейс WirenBoard: [https://wirenboard.com/wiki/Wiren Board Web Interface](https://wirenboard.com/wiki/Wiren_Board_Web_Interface)
3. Утилита для извлечения исторических данных из внутренней базы данных: <https://wirenboard.com/wiki/Wb-mqtt-db-cli>
4. Правила в WirenBoard: [https://wirenboard.com/wiki/How to write rules](https://wirenboard.com/wiki/How_to_write_rules)
5. Написание скриптов WirenBoard для начинающих: <https://wirenboard.com/wiki/Wb-jscript>
6. Полное описание движка правил WirenBoard: <https://github.com/wirenboard/wb-rules>
7. Некоторые понятия области Интернета вещей: <https://iot.ru/wiki/>