



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»
РТУ МИРЭА

ДИСЦИПЛИНА	Программирование управляемых технических систем (укажите полное наименование дисциплины без сокращений)
ИНСТИТУТ	ПТИП (укажите название учебного института)
КАФЕДРА	Индустриального программирования (укажите полное наименование кафедры)
ВИД УЧЕБНОГО МАТЕРИАЛА	Материалы для практических/семинарских занятий (в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	Закожурников Сергей Сергеевич (фио)
СЕМЕСТР	2, 2025 (указать семестр обучения, учебный год)

Практическая работа

Изучение интерфейса программы OpenHAB

OpenHAB осуществляет электронную связь с интеллектуальными и не очень интеллектуальными устройствами, выполняет определяемые пользователем действия и предоставляет веб-страницам определяемую пользователем информацию, а также определяемые пользователем инструменты для взаимодействия со всеми устройствами. Для этого openHAB сегментирует и разделяет определенные функции и операции. В следующей таблице 1 дано общее описание наиболее важных концепций.

Таблица 1

Концепции	Значение
Привязки	Являются компонентом openHAB, который обеспечивает интерфейс для электронного взаимодействия с устройствами.
Вещи	Являются первым сгенерированным openHAB представлением ваших устройств
Каналы	Представляют собой соединение openHAB (программное обеспечение) между «Вещами» и «Предметами»
Предметы	Представляют собой сгенерированное openHAB представление информации об устройствах
Правила	Которые выполняют автоматические действия (в простейшей форме: если произойдет «это», openHAB сделает «это»)
Карта сайта	Это созданный openHAB пользовательский интерфейс (веб-сайт), который представляет информацию и позволяет взаимодействовать

Каналы

Каналы — это логическая связь между Вещью и Предметом. Каналы происходят из определения Вещей и определяют, как ваша Вещь может взаимодействовать с Предметом (и наоборот). Вы будете создавать каналы при определении вашей Вещи.

Во время определения вашей вещи вы определите канал, к которому будет привязана ваша вещь. Эти два шага гарантируют, что

openHAB сможет передавать информацию от вещи к предмету (и наоборот).

Привязки

Привязки — это программные пакеты, которые устанавливаются пользователем в openHAB. Основная цель Bindings — установить соединение между вашим устройством и вашей вещью. Привязки взаимодействуют с вашим устройством и переводят все команды в openHAB и обратно между вашим устройством и вашей вещью.

Привязки представлены в разделе дополнений (открывает новое окно) этого веб-сайта. Здесь находится доступный для поиска список из нескольких сотен привязок для поддержки максимально возможного количества устройств. Новые привязки регулярно добавляются по мере того, как разработчики интегрируют в openHAB больше устройств.

Для каждой привязки предоставляются подробные инструкции и примеры, которые включают руководство по настройке (если таковые имеются) самой привязки, определение вещей, поддерживаемых этой привязкой, и каналов, которые эти вещи предоставляют. В большинстве случаев описание также содержит полностью проработанный пример, включающий определение Вещей и их Каналов, Предметов, связанных с этими Каналами, и использование этих Предметов в карте сайта.

Правила

OpenHAB имеет высокоуровневый интегрированный, лёгкий, но мощный движок выполнения правил. «Правила» используются для автоматизации процессов. Каждое правило запускается по какому-либо событию и выполняет сценарий, исполняющий задачи разного рода, такие как управление освещением, изменение значений элементов, математические вычисления или запускает таймер.

Правила расположены в папке \$OPENHAB_CONF/rules. Демонстрационный комплект включает в себя файл demo.rules, в котором есть пара примеров и он может стать хорошей отправной точкой. Файл может содержать множество правил. Все правила, сохранённые в одном файле, будут выполняться в общем контексте, то есть они могут общаться и обмениваться переменными друг с другом. Поэтому имеет смысл иметь разные файлы правил для разных вариантов использования или категорий.

Определение на основе пользовательского интерфейса

Правила могут создаваться и редактироваться через пользовательский интерфейс UI. Вы можете найти редактор просматривая раздел Settings -> Rules. Нажмите на иконку «+» чтобы добавить правило, описать его название и на что оно должно срабатывать.

В нашем примере правило запускается при запуске openHAB для определения окружения. Если добавить второй триггер в раздел When, то будут учитываться срабатывания обоих триггеров (логическое ИЛИ). Если добавить дополнительное действие в Then, то оно будет выполняться последовательно. Если же добавить дополнительное условие в раздел «But only if», то оно будет работать совместно с остальными (рис. 1).

The screenshot shows the configuration interface for a rule in openHAB. At the top, the 'Unique ID' is 86e99503b8 and the 'Name' is 'Тестовое правило'. Below this is a 'Description' field. A 'Reorder' button is located on the right. The interface is divided into three main sections: 'When', 'Then', and 'But only if'. The 'When' section contains one trigger: 'When the system has reached start level 100', with a sub-description 'This triggers the rule if a given start level is reached by the system.' and an 'Add Trigger' button. The 'Then' section contains one action: 'execute a given script', with a sub-description 'Allows the execution of a user-defined script.' and an 'Add Action' button. The 'But only if' section contains one condition: 'it is a weekend', with a sub-description 'Checks if the current day is on the weekend.' and an 'Add Condition' button. Below these sections is a 'Tags' section with a 'Semantic Class' dropdown set to 'None' and an 'Add tag' button. At the bottom is a 'Remove Rule' button.

Unique ID	86e99503b8
Name	Тестовое правило
Description	

Reorder

When

- When the system has reached start level 100
This triggers the rule if a given start level is reached by the system.
- Add Trigger

Then

- execute a given script
Allows the execution of a user-defined script.
- Add Action

But only if

- it is a weekend
Checks if the current day is on the weekend.
- Add Condition

Tags

Semantic Class: None

Add tag

Remove Rule

Рисунок 1 – Меню тестового правила

Если нажать Show All, то вариантов станет заметно больше. За кнопкой. Show All зачастую скрываются очень полезные вещи. Например, при выборе триггера по элементу становятся доступны все элементы, а не только входящие в семантическую модель, как в базовом варианте выбора.

Нажмите Add Action и выберите Run Script. Выберите Rule DSL и вводите правила по образцу описанному в данном руководстве (рис. 2).

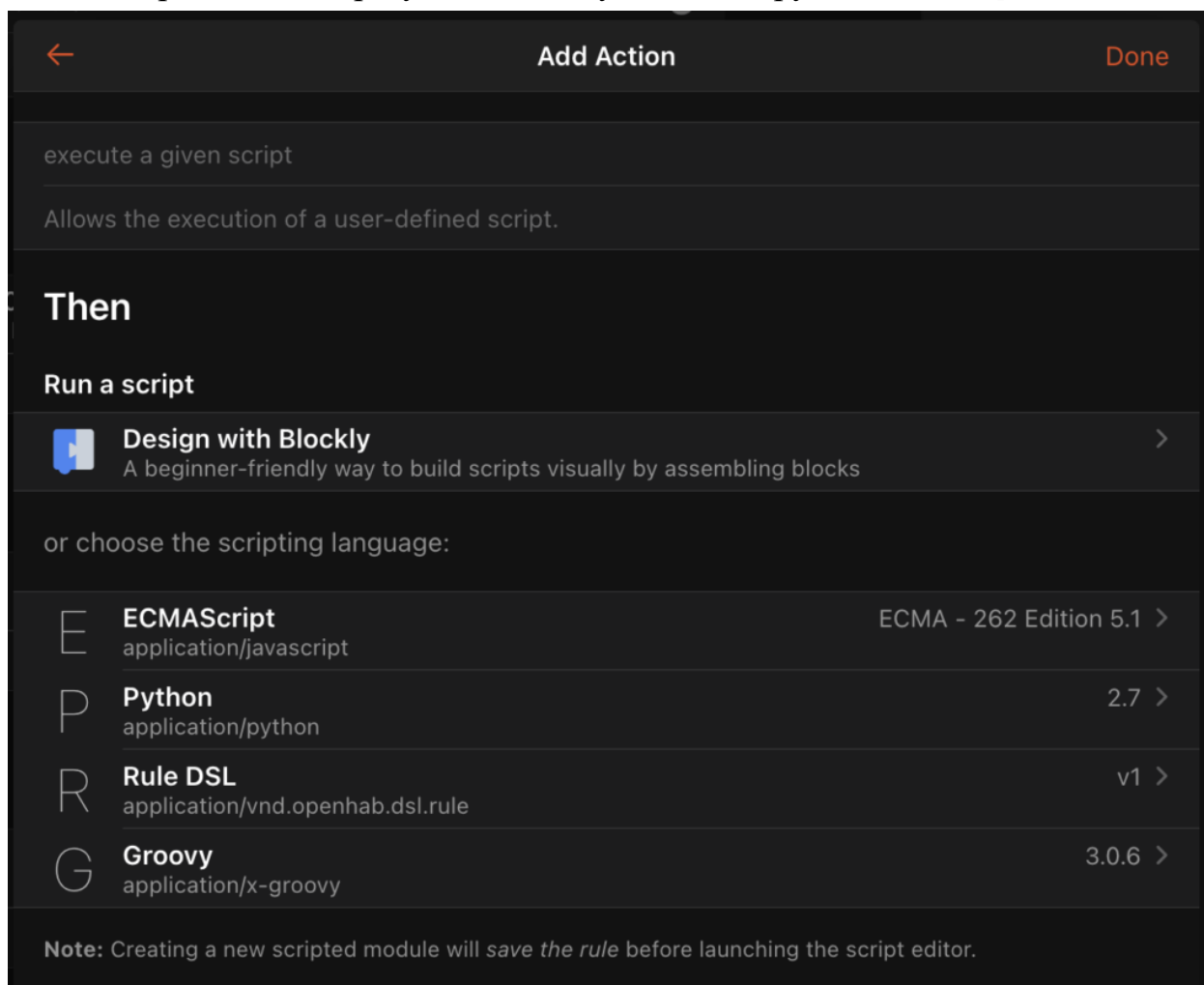


Рисунок 2 – Меню добавления скрипта

Поддержка средств разработки (IDE)

Расширение openHAB VS Code Extension предлагает поддержку построения правил. Он включает в себя проверку синтаксиса и раскраску, проверку с маркерами ошибок, подсказки в контенте (Ctrl+Space), включая шаблоны и т. д. Это делает создание правил очень легким! Посетите страницу редакторов для получения дополнительной информации и дополнительных возможностей редактора.

Синтаксис

Синтаксис правил основан на Xbase и в результате он во многом схож с Xtend, который также построен поверх Xbase. Для получения дополнительной информации можно сослаться на документацию к Xtend.

Файл правил — это текстовый файл следующей структуры:

1. Импорт
2. Описание переменных
3. Правила

Раздел **«Импорт»** содержит инструкцию импорта, как и в Java. Как и в Java, они делают импортированные типы доступными без необходимости использовать для них полное имя. Например: `import java.net.URI`. Несколько модулей импортированы по умолчанию, так что классы из них не требуется импортировать отдельно, как показано в таблице 2.

Таблица 2

```
org.openhab.core.items
org.openhab.core.persistence
org.openhab.core.library.types
org.openhab.core.library.items
org.openhab.model.script.actions
```

Раздел **«Описание переменных»** можно использовать для объявления переменных, которые должны быть доступны для всех правил в этом файле. Вы можете объявить переменные с начальными значениями или без них и изменяемыми или доступными только для чтения, как показано в таблице 3.

Таблица 3

```
// a variable with an initial value. Note that the variable type is
// automatically inferred
var counter = 0

// a read-only value, again the type is automatically inferred
val msg = "Здесь был Вася"

// an uninitialized variable where we have to provide the type (as it cannot
// be inferred from an initial value)
var Number x
```

Раздел **Правила** содержит список правил. Каждое правило имеет следующий синтаксис, показанный в таблице 3.

Таблица 3

```
rule "<RULE_NAME>"
when
    <TRIGGER_CONDITION> [or <TRIGGER_CONDITION2> [or ...]]
then
    <SCRIPT_BLOCK>
end
```

<RULE_NAME> — Каждое правило должно иметь уникальное имя (помещённое в кавычки). Рекомендуется использовать осмысленные названия правил.

<TRIGGER_CONDITION> — Иницилирующее событие, при котором выполняется логика правила. Правило выполняется в ответ на одно или несколько условий триггера. Несколько условий разделяются ключевым словом `or`. Пожалуйста, смотрите ниже различные возможные триггеры.

<SCRIPT_BLOCK> — Содержит логику, которая должна быть

выполнена при выполнении условия триггера, подробности о его синтаксисе см. в разделе сценария.

Задание:

1. Добавить вещь в программе OpenHAB
2. Создать привязку
3. Создать правило
4. Создать канал связи между вещью и предметом.