

## Разработка сегмента умного дома

*Чуриков Илья Игоревич*

*Российский экономический университет им. Плеханова*

*студент*

### Аннотация

В статье представлено описание применения современных технологий Интернета вещей в повседневной жизни человека. Статья ознакомит читателя с современными технологиями на рынке Интернета вещей и докажет, что современные технологии могут быть использованы даже в самых обыденных занятиях. Здесь рассмотрено подключение, регистрация и запуск сегмента Умного дома.

**Ключевые слова:** Интернет вещей, Умный дом, распределенная система, повседневный быт

## The development of smart home segment

*Churikov Ilya Igorevich*

*Plekhanov Russian University of Economics*

*student*

### Abstract

This research article presents a description of using modern technologies of Internet of things in people's everyday life. This article will acquaint readers with different technologies on the Internet of things market and will prove that these technologies can be used even in ordinary routine. Connection, setting up and activation of smart house segment are studied in this article.

**Keywords:** Internet of Things, Smart home, distributed system, everyday life

### Введение

Интернет вещей все плотнее и плотнее входит в нашу повседневную жизнь. Тема «умного дома» в настоящее время является довольно актуальной. Практически любой пользователь что-то слышал или даже видел в действии работу умного дома. Умный дом - это новый сегмент рынка ИТ-технологий.

Единый стандарт умного дома еще не выведен, как и нет инструкции, которой стоит следовать, поэтому построить свой умный дом может любой пользователь так, как ему захочется. По этой причине я не смог пройти мимо этой увлекательной, на мой взгляд, темы и углубился в ее изучение.

«Умный дом» — это не только распределенная система с искусственным интеллектом, которая общается с Вами, разогревает Вам обед пока вы едете домой, следит за вашим домом во время отсутствия жильцов [1].

«Умный дом» — это абсолютно любая система автоматизации, которая в какой-то мере облегчает Ваш повседневный быт. При этом не имеет значения что она делает: автоматически включает свет на улице в темное время суток, поливает цветы по определенным дням или кормит домашних животных, в случае Вашей задержки допоздна. Если в доме что-то срабатывает по какому-либо алгоритму, автоматически, значит эту систему уже можно назвать «Умным домом», только с ограниченным функционалом. В начальном виде система должна понимать ситуации, которые происходят в доме в тот или иной момент, и реагировать на них.

В процессе курсовой работы мы познакомимся с современным решением для «умного дома», соберем, опробуем и протестируем возможности такой распределенной системы.

### **Программная часть**

Для реализации нашей распределенной системы, мы будем использовать свободную платформу OpenHAB. OpenHAB расшифровывается как «Open Home Automation Bus», что в переводе «открытая платформа домашней автоматизации». Этот проект нацелен на создание универсальной платформы для подключения к ней разных сегментов умного дома от разных производителей. OpenHAB это программное обеспечение, которое позволяет интегрировать разные автоматизационные системы в одно единственное готовое решение.

Область умных домов развивается уже довольно долгое время. За этот промежуток времени появилось большое количество различных вендоров «умного» оборудования, поставляющих на рынок все от автоматических жалюзи до лампочек, управляемых со смартфона. Все устройства работают на разных протоколах (проводных и беспроводных).

В итоге, конечный пользователь, который хочет автоматизировать свое жилище, должен делать выбор между продуктами от различных вендоров. В итоге ему приходится использовать разные системы для контроля разных устройств. Кроме того, создание полноценного интернета вещей становится сложно реализуемым.

OpenHAB же реализует единую шину, т.е. помогает собрать все устройства с разными протоколами в единую сеть, обособляя пользователя от каждого конкретного протокола. Таким образом, можно пользоваться единственным средством управления (скажем, приложением на смартфоне) и реализовать сколь угодно сложную логику взаимосвязи между всеми «умными» устройствами.

OpenHAB — это некий сервер, работающий на любом компьютере под любой ОС. Вся установка - это распаковка дистрибутивов и запуск Java машины. Далее следует настройка и отладка элементов.

OpenHAB можно представить как конструктор. Базирующийся на технологии OSGi, он конфигурирует каждый отдельный плагин (Binding) «налету», не перезагружая сервер.

Биндинг в openHAB — это то самое связующее звено между конкретным протоколом устройства и единой системой openHAB. Пользователь может добавить в свою версию openHAB только те плагины, которые необходимы именно ему.

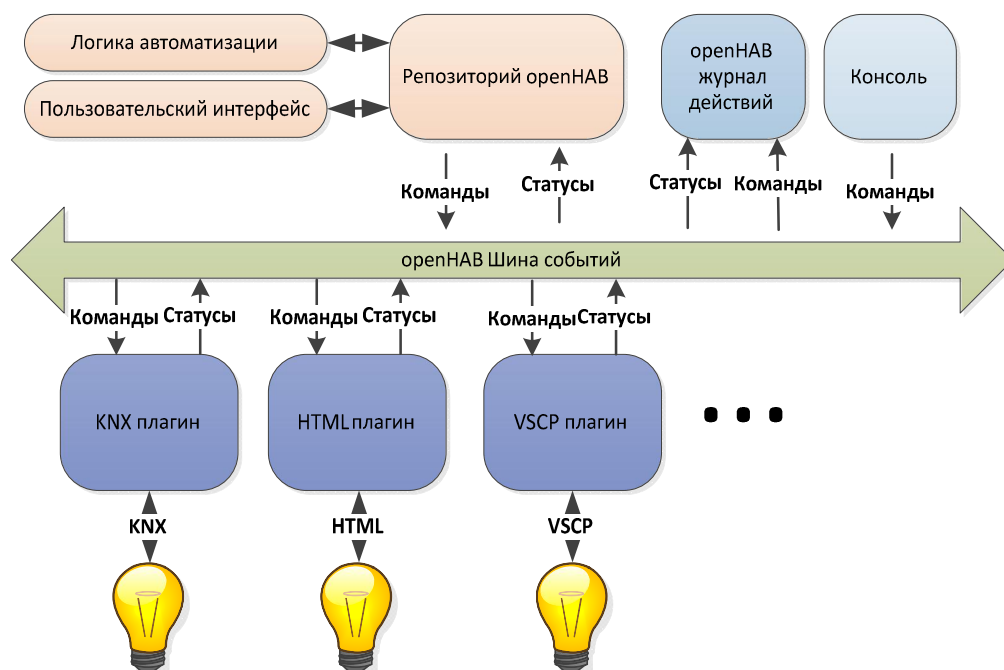


Рисунок 1 - Схема openHAB

После настройки взаимосвязей между каждым «умным» девайсом и соответствующим ему биндингом система сможет общаться с ним через простые команды, независимых от конкретного протокола. Например, для включения/выключения есть команды ON/OFF.

Далее, используя только такие простые команды, можно будет создавать очень сложные алгоритмы взаимодействия и программировать свой умный дом.

Для настройки используются следующие параметры[2]:

- Items — глобально доступный объект, обычно это внешнее устройство или функция устройства. Могут быть разных типов, например, Switch для выключателя, Dimmer для диммера. Есть также String и Number, это Items которые содержат строку и число.
- Rules — некие правила, которые выполняются при наступлении какого-либо события или по определенному графику.
- Sitemaps — перечисление items с настройками отображения в пользовательском интерфейсе.

Один из плюсов openHAB в том, что пользовательские интерфейсы унифицированы практически на всех устройствах.

### Физическая часть

Для автоматизации полива используем следующие компоненты:

- Компьютер с поднятым и настроенным сервером openHAB

- Siemens LOGO! - устройство выполняет роль «умного» реле
- Raspberry PI – мини компьютер, выполняющий преобразование протокола Siemens logo в HTTP
- 4 электромагнитных клапана, открывающихся по команде Siemens LOGO!

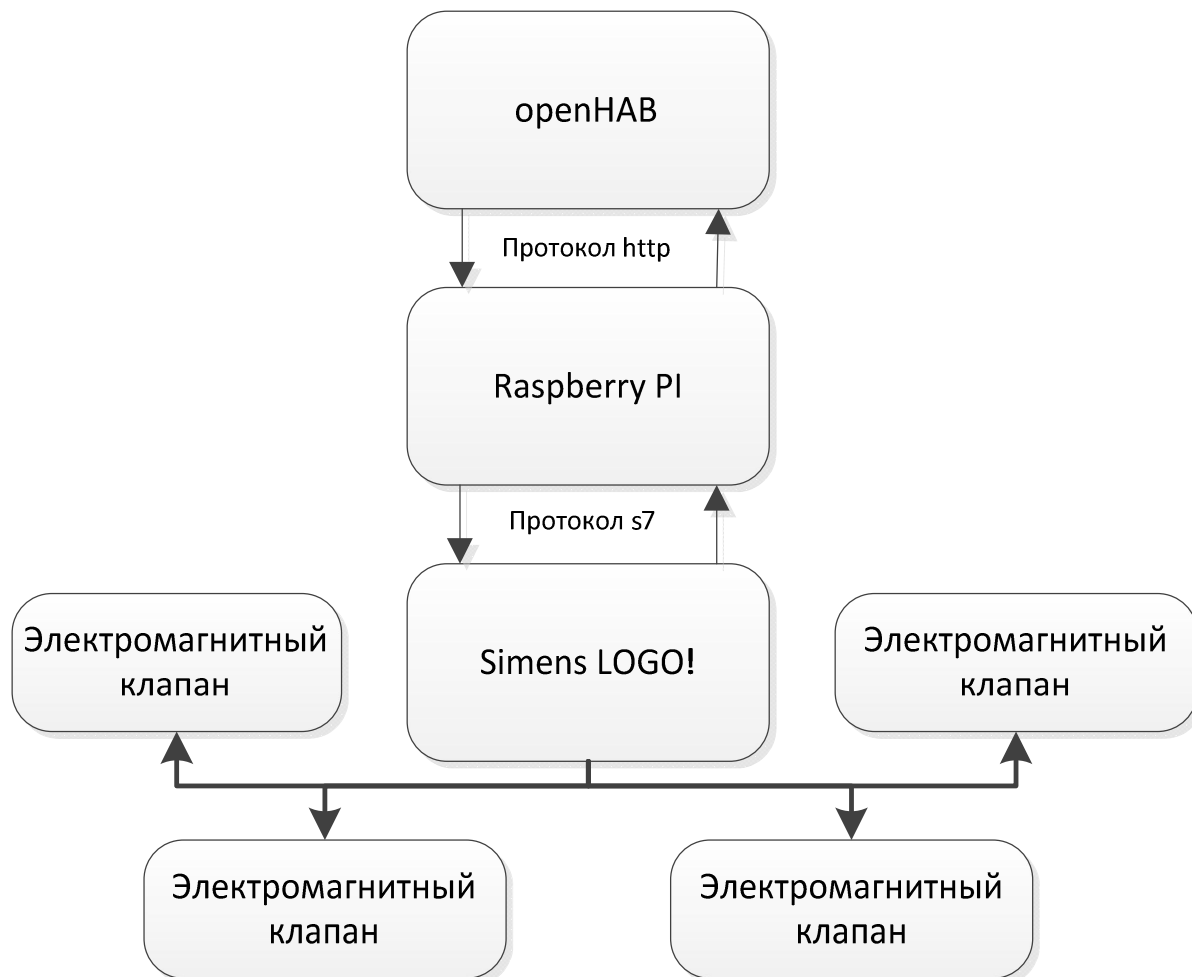


Рисунок 2 - Схема автоматического полива

### Конфигурация

OpenHAB поднимаем на постоянно включенном компьютере под ос Linux. Напомню, что сервер можно поднять на любой ОС и любом компьютере. Был выбран домашний компьютер, потому что на нем осуществить это оказалось проще всего.

Для связи с остальными частями нашей распределенной системы используем http биндинг. Для полноценной работы openHAB нужны 3 файла в папке configuration: items, rules, sitemap. Начнем разбирать по порядку. Пример items:

Группа POLIV- глобальная группа для удобства управления в будущем.

2 подгруппы POLIV\_SW - переключатель полива и POLIV\_ST - показатель статуса (сокращения введены пользователем от английской switch и status)

Далее описываем по образцу: тип, название, «Название для значка», (Группа1, Группа2), {конфигурация биндинга}.

У нас:

- тип - switch (переключатель с двумя положениями)
- название - Valve1 (введено для удобства)
- лейбл - Полив 1(2, 3, 4) (так будет отображаться в UI)
- группы - POLIV, POLIV\_ST для статусного айтема и POLIV, POLIV\_SW для айтема переключения
- Конфигурация. Разберем для переключателя: скрипт ищет по имени Valve1 переменные в программе на Raspberry, найдя переменную, он рассматривает атрибуты и с помощью запроса GET по команде ON/OFF ставит по определенному адресу в атрибут соответственно логическую единицу либо нуль.

```

1 Group POLIV
2 Group POLIV_SW (POLIV)
3 Group POLIV_ST (POLIV)
4 Switch Valve1 "Полив 1" <text> (POLIV, POLIV_ST)
5 { http="<[loheat:1000:JSONPATH($..Devices[?(@.Name==Valve1)].Attributes[0].ValueText]}"
6 Switch Valve1_sw "Включить полив 1" <text> (POLIV, POLIV_SW)
7 { http="<[loheat:1000:JSONPATH($..Devices[?(@.Name==Valve1)].Attributes[3].ValueText)]
>[ON:GET:http://orangepi:8091/rest/devices/6/attributes/4/value?set=1]
>[OFF:GET:http://orangepi:8091/rest/devices/6/attributes/4/value?set=0]"
8

```

Рисунок 3 - Items

Подобным образом описываются все айтемы в системе.

Далее, что касается файла rules, наша система довольно простая, поэтому мы можем не создавать правила в openHAB, а добавить их, к примеру, на нашем «умном» реле.

После описания айтемов в sitemaps у нас получится следующий пользовательский интерфейс.

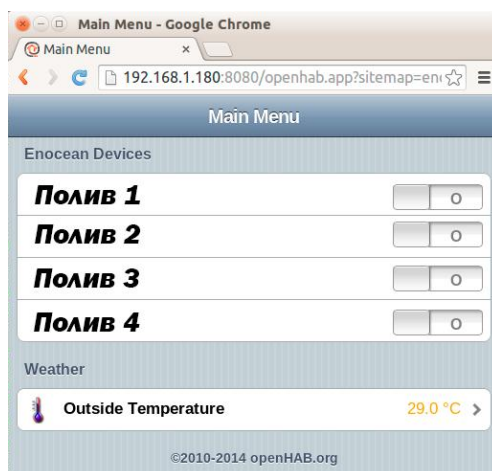


Рисунок 4 - Пользовательский интерфейс

### Элементы системы

Raspberry PI - одноплатный компьютер маленького размера, который разрабатывался как недорогая система для обучения информационным технологиям, но со временем области применения расширились и популярность компьютера возросла в разы.

Зачем нам нужен этот компьютер? К сожалению, протокол нашего умного реле Siemens LOGO! не может напрямую общаться с openHAB, поскольку такого биндинга элементарно не существует.

Для нашего проекта нам понадобится версия с Ethernet выходом.

Плата будет использоваться для преобразования протокола S7, на котором общается Siemens LOGO!, в http, на котором, в свою очередь, разговаривает биндинг openHAB. Raspberry PI был выбран, потому что нам нужно устройство, которое будет работать 24/7 и не потреблять много энергии.

Скрипт, который мы будем использовать, написан возмущенными пользователями Siemens, протокол которого является закрытым и сложнодоступным. Скрипт находится в общем доступе на одном из немецких сайтов.

Приведем и разберем для примера небольшую его часть:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <settings>
    <logo ip="192.168.4.224" />
    <httpWebservice port="8091" />
  </settings>
  <infrastructure>
    <group name="Vnukovo_224">
      <device id="6" name="Valve1">
        <attribute id="1" name="Valve1_ON" address="942.4" datatype="bit" >
          <valuertext value="0" text="OFF" />
          <valuertext value="1" text="ON" />
        </attribute>
        <attribute id="4" name="Valve1_sw" address="32.0" datatype="bit" >
          <valuertext value="0" text="OFF" />
          <valuertext value="1" text="ON" />
        </attribute>
      </device>
    </group>
  </infrastructure>
</configuration>
```

Рисунок 5 - Код для Raspberry PI

Часть settings определяет настройки IP и порта, к которому подключен наше «умное» реле. Далее описываем структуру (infrastructure), которую будем реализовывать. Напомню, что в нашем случае это 4 клапана. Для удобства мы сгруппировали наши 4 клапана и назвали эту конфигурацию «Vnukovo\_224». Далее идет описание каждого клапана (device). В атрибутах мы прописываем id, присваиваем название этому клапану, добавляем адрес, закрепленный именно за этим выходом Siemens LOGO для клапаном в технической документации (на нашем это 32.0 для управления клапанами и 942.4 для просмотра их состояния), затем присваиваем тип этому атрибуту. В следующих 2х строчках мы описываем, как будет выглядеть интерфейс при

подключении к Raspberry по локальной сети напрямую. То есть, если компьютер получает от LOGO значение «1», то в интерфейсе, напротив названия нашего клапана стоит значение «ON», если получаем «0»-»OFF».

Таким образом, Raspberry помогает общаться с LOGO и видеть состояния всех клапанов.

Siemens LOGO! восьмого поколения являются многофункциональными универсальными изделиями, используемые для построения простейших систем автоматизации с логической обработкой данных. Алгоритм функционирования элементов задается специальной программой, составленной из набора определенных функций.

В нашем случае, для программирования мы используем ПО, поставляемое в комплекте с устройством. Данная программа позволяет рисовать логическую систему на ходу, используя различные логические блоки операций. Для наших нужд не будут использоваться сложные схемы. Разберем минимально необходимый вариант:

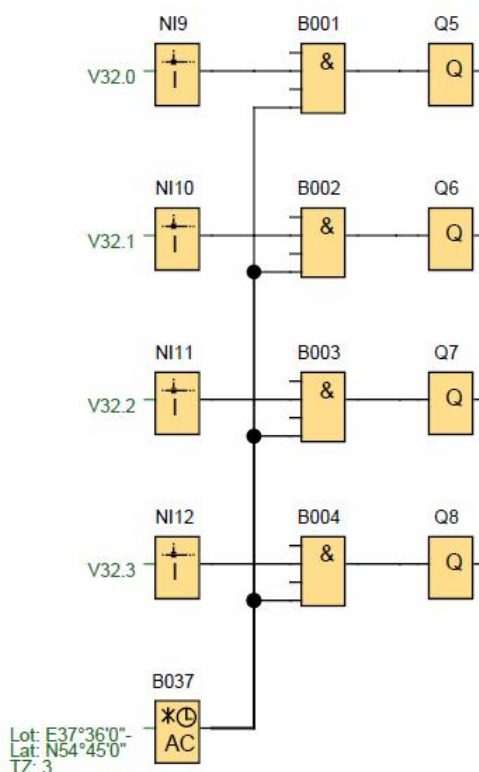


Рисунок 6 - Логическая схема на Siemens LOGO!

Блоки со входными сигналами из вне принимают команды от Raspberry, в скрипте которого мы прописали адреса управления (32.0; 32.1; 32.3; 32.4)

На эти адреса приходят команды логической единицы или нуля. После получения, сигнал поступает в конъюнктор, где выполняется операция

логического умножения сигнала с Raspberry на сигнал с мировых часов, которые получают время с отдельного сервера. На часах мы устанавливаем определенные периоды, в которые разрешен полив. У нас такими периодами является утро с 8:00 до 10:00.

Итак, если приходит логическая единица с компьютера в подходящий промежуток времени, то на выход реле подается логическая единица и контакты смыкаются, тем самым открывая электромагнитные клапаны.

Электромагнитные клапаны монтируются на шланги для дистанционного управления открытием или закрытием потока воды для полива территории.

К клапанам приходят провода от Siemens LOGO!, по которым идет сигнал, открыть или закрыть клапан. На фото мы видим шланг, с постоянным потоком воды, 4 клапана с подходящими к ним проводами и шланг, подключенный к выходу одного из клапанов.



Рисунок 7 - Реализация

Таким образом, собранная система работает логично и предсказуемо, при отключении одного из элементов все система не рушится, а продолжает работать по определенным правилам.

### **Заключение**

В последнее время умные вещи все плотнее и плотнее входят в нашу повседневную жизнь, облегчая ее и заменяя монотонные и рутинные действия человека. Многие фирмы выходят на этот новый сегмент рынка и добавляют что-то абсолютно новое и оригинальное в своем роде. На данный момент, умный дом - это больше развлечение, но в скором времени, по



моему мнению, такие устройства будут встречаться повсеместно, в каждом доме и каждой квартире. Человек может отдать машине некоторые свои мелкие повседневные рутинные дела, а сам потратить освободившееся время на то, что ему больше нравится. По моему мнению, в ближайшем будущем интернет вещей сделает огромный скачок вперед. Для этого созданы благоприятные условия. Например, любой человек с хорошей идеей, но без финансов на ее создание, может запустить проект на краудфаиндинговой площадке, тем самым одновременно проверить как люди отнесутся к его идее и собрать средства на производство продукта. Множество решений для умного дома приводят к тому, что возникают споры в протоколах связи между различными устройствами от фирм. Решением данной проблемы является, рассмотренный в курсовой работе сервис openHAB, который собирает и согласует все устройства воедино.

Во время создания сегмента умного дома, я столкнулся с некоторыми проблемами. Во-первых, это высокая цена на комплектующие. Не каждый решится потратить хотя бы какую-нибудь сумму денег на автоматизацию полива. Во-вторых, монтаж компонентов в уже готовый дом совсем не прост. Необходимо иметь выходы Ethernet по всему дому, либо использовать беспроводные связи, что увеличивает стоимость проекта в разы и уменьшает стабильность системы. Но с другой стороны, настройка связей понятна любому, кто хоть немного знаком с азами программирования. Все соединяется по определенным логическим правилам, интуитивно понятным для пользователя. Кроме того, автоматизация максимально упрощает жизнь, появляется желание сделать свой дом комфортным, а веб-интерфейсы позволяют следить за ним из любой точки мира и быть абсолютно спокойным за свое жилище.

Есть много мнений по поводу умных вещей, но я склоняюсь к тому, что будущее за интернетом вещей и автоматизацией во всех сферах человеческой жизни.

### **Библиографический список**

1. OpenHAB — стань программистом собственного жилища. - URL: <https://habrahabr.ru/post/232969/>
2. Официальный сайт openHAB. - URL: <http://www.openhab.org>