Table of contents

Preface 5

The rewards are high! 5

Welcome to the connected house 2.0 7

The (non)sense of home automation 8

The era of the-Internet-of-Things 10

What is a connected Product? 11

Everything connected! 12

Challenges 13

Certified for humans! 16

Voice assistants are a part of the family 17

Kids and Voice Assistants 19

Everything exists 20

The Connected House 2.0 23

The Ingredients 24

Introduction 25

Amazon Echo 25

Google Nest 28

Apple HomeKit 30

Samsung - SmartThings 32

OpenHAB 41

Home Assistant 45

Domoticz 46
Node-Red 48
If This Then That (IFTTT) 53
The AllThingsTalk Cloud by ALSO 56
The Things Network 73
HomeBridge 76
Hoobs 77
Homekit Bridged 79

The Technology 81

The glue between a system-of-systems architecture 82
MQTT 88
Z-Wave 96
The Zigbee Protocol 99
LPWAN 101
BlueTooth 103

The Hardware 104

The Smart HUB 105
Off-the-shelf sensors 108
Building your own devices 122

Recipes 135

Basic Recipes 137
Connected devices 167
Recipes for building User Interfaces 247

Recipes to add automation to your house 296

Security & Privacy 325

What can we do about it? 328

The Future connected house 336

More sensors results in more insights 337
Self driving cars 340
Living longer at home 342
Going from monitoring to autonomous behavior 343
Brain Power Index 345
The active home 349

Preface

The rewards are high!

Over the last few decades, technology has evolved so much that it allows us to provide affordable home automation that is more applied to humans. The IKEA TRADFRI product line and the Philips Hue lights reflect affordable home automation that can be seen as a commodity product today. The smart assistants, on the other hand, are an entirely new range of innovative products with a human-like interface that brings the connected home closer to its residents.

Manufacturers of household appliances are releasing more and more connected versions of their products. Startups are developing new innovative 'connected' products that do not yet exist. Even suppliers you wouldn't expect, such as Rituals, active in Home & Body cosmetics, release a connected version of their perfume diffuser with their Genie 2.0.

Most of these 'connected' products have a good out-of-the-box user experience, are easy to install and bring some added value to our lives. What is unfortunate is that they are isolated solutions, which do not have the openness and integration possibilities to talk to each other. At best, vendors offer or refer to an IFTTT integration (if that's the case) to link things together, but to be honest, you will quickly reach the limits of this approach.

On the other hand, there is the open-source community that offers solutions such as Openhab, Hassio, Zigbee, 2mqtt and Homebridge to name a few and offer a lot of functionality to build your own connected home.

This is the playground for the technical skilleld people. The internet is crowded with DIY smart home implementations brewed by people making use, or even contribute to the open-source community projects. Many people invest their time in it for free, they do it for honour and are proud when someone uses one of their contributions to the community. In most cases these products are superior to commercial solutions on many levels, such as the available features, the speed to support new products, the openness and (of course) the price. You can't beat a free meal!

The downside is that it is complicated for non-technically skilled people to work with these platforms. It is a rather long learning curve and requires a lot of patience. The reality of this DIY approach is that you will spend a lot of time looking for answers and hoping that someone has solved the problem for you. It is slow, often requires a lot of research, but the reward is high!

This was my motivation to write this ‘cookbook’. I try to avoid spending time reinventing the wheel. Giving an overview of what is available for free use and providing ‘ready-to-use’ building plans using function-rich open platforms where less technical people would be able to successfully build a solution within a reasonable amount of time.

I hope you will enjoy it.

Peter Leemans

October 2020

Welcome to the connected house 2.0

The (non)sense of home automation

Home automation systems have been around for decades and many people, including myself, are fascinated by them. The first commercial home automation systems originated in the 1980s. At the time, the idea that all your devices could be controlled via one central hub sounded as science fiction. Some of those systems were based on the X10 protocol, which still exists today. The Pico Electronics X-10 Powerhouse is an example of that. It communicated with lights via a 120kHz signal burst sent through the home's power lines. Although technically it worked, it had its limitations. Because the signals did not stop at the doorstep, they could penetrate your neighbour's home and interfere with a similar system. Therefore, hacking would have been child's play, as the data on the power line was not encrypted. The focus was on managing lights and other appliances using a wired remote control or an MS-DOS program on your IBM computer.

X10 controllers: A simple controller (bottom left), a radio controller (top center), and an original controller (bottom right)

Next, there was HAL 2000 (Home Automated Living), adding voice control using a Windows95 computer. You could manage your home with a microphone like today's smart speakers avant-la-lettre. The technology was not at the same level of current smart speakers and generated many unwanted commands if you had a conversation in front of HAL 2000 with someone else.

Hal 2000

In the same period, you had specialized companies such as the Belgian Teletask who were pioneers in building home automation systems. They focused more on the high-end residential and professional building market. Contrary to the X10 powerline and HAL2000 DIY systems, the Teletask systems used a worldwide network of certified distributors and system integrators. These systems were expensive and not flexible. Although technology shortcomings were to blame, lack of focus on customer needs by system designers was the primary shortcoming at that time.

Most of the installed home automation systems did not live up their potential. In many cases, only one person – a professional or 'dad' – had knowledge of the system. Other family members didn't know how to utilize and take advantage of the system.

Gradually, the industry began to understand its limitations and the technical evolution created new possibilities for home automation, such as wireless communication, miniaturization, the exponential growth in computing power with upcoming cloud services.

While the technical evolution was essential to the progression of home automation, the change towards a more consumer-centric thinking made the technology accessible to everyone. The idea was not so much to automate everything, but more about to help people connect to their houses and their world. This was achieved by offering people exactly what they want – whether that is valuable insights, money-saving opportunities or just convenience – and not getting carried away with futuristic utopias.

Because the connected home industry reinvented itself, it was able to create an entirely new market –

one with endless potential. Comprised of many different players, it aims to offer a great user experience while making people's lives easier, safer, and more comfortable. Vendors of these new 'connected' products for the home focus on ease-of-installation, avoiding the aid of a professional to install them.

The connected house 2.0 is about this new wave of connected products that assist in creating your connected home. With the help of IoT & Cloud Services and by taking advantage of open (source) platforms, it brings value and comfort to the residents.

The '2.0' refers to the wave of 'connected' products that emerge on the market every day. According to Statica Research, the total addressable market for smart home products and technology will reach $53 billion by 2022. From intelligent kitchen appliances to voice assistants, technology companies and startups are leveraging the Internet of Things (IoT) to connect everyday devices to the cloud and create new experiences for customers at a rapid pace.

This book aims to show how easy it is for a tech novice to embrace a number of these connected products and integrate them by using a best-of-breath approach.

The era of the-Internet-of-Things

During the last decade, the Internet of Things caused a rapid growth in connected products, One of the first real examples of the IoT originated from the early 1980s, when IT-students of the Carnegie Mellon University hooked up a Coca-Cola machine to the Internet to check if drinks were available and cold. However, IoT as a concept was only officially named in 1999 by the British technology pioneer Kevin Ashton in the United States.

Today, IoT includes billions of devices that collect and share data with one another. While some devices connect via wired and wireless networks, others do so over intranets and the Internet. Small cameras, sensors, monitors, and meters observe, measure, and report their physical surroundings. They detect motion, temperature, light and sound levels, energy usage, and air quality. They can track human activities, including health, mobility, eye movement, mood, location, paths of movement through space, and stress levels.

But despite all those connected devices already out there, IoT is still in his infancy.

The reason for this is the diversity of technologies that constitute IoT. Not only is it difficult to build great

connected products, it is even more demanding for existing businesses to transform into a digital world and find the right connected products for their business, also referred to as products & services.

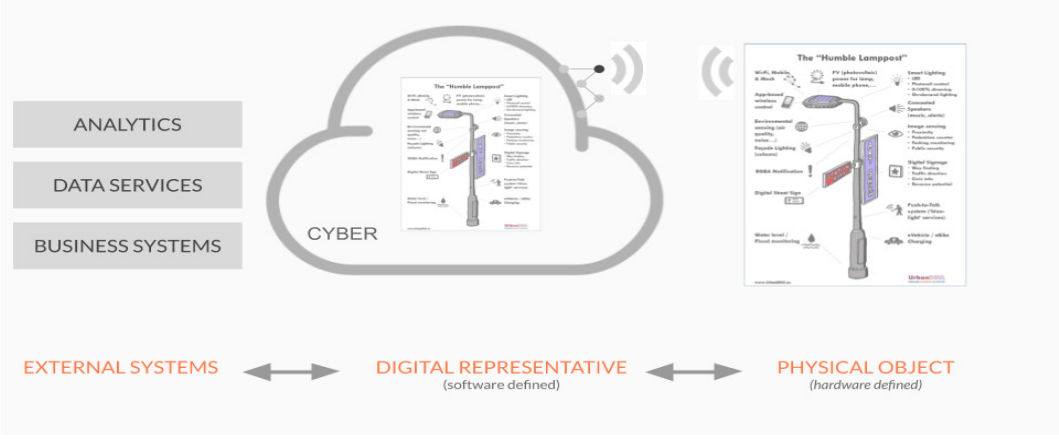
While most organizations today make decisions based on static, point-in-time data, all of that is about to change. Thanks to rapidly evolving hardware and software, we are seeing integrated solutions that promise real-time insights. Gradually, IoT is entering our homes, with smart speakers acting as home assistants as an example. Still, many objects are not connected yet, leaving a lot of opportunities unexplored at this moment.

The Internet of Things is enters an era where physical assets are connected to each other to feed sensor data and allow us to make actionable insights. Suddenly the IT world comes very close to our real world, where it is applied to 'simulate' human-like decisions.

It might take some time, but eventually, most 'things' and objects will become connected. With the data they produce, we will be able to receive some actionable insights. The world of the Internet-of-Things is growing. IoT is not the privilege of the tech-savvy , but for anyone who is curious to discover this domain and its endless possibilities

What is a connected Product?

To illustrate what a connected product is, let us look at the "humble lamppost", an example that was introduced by Jason Warwick. [1].

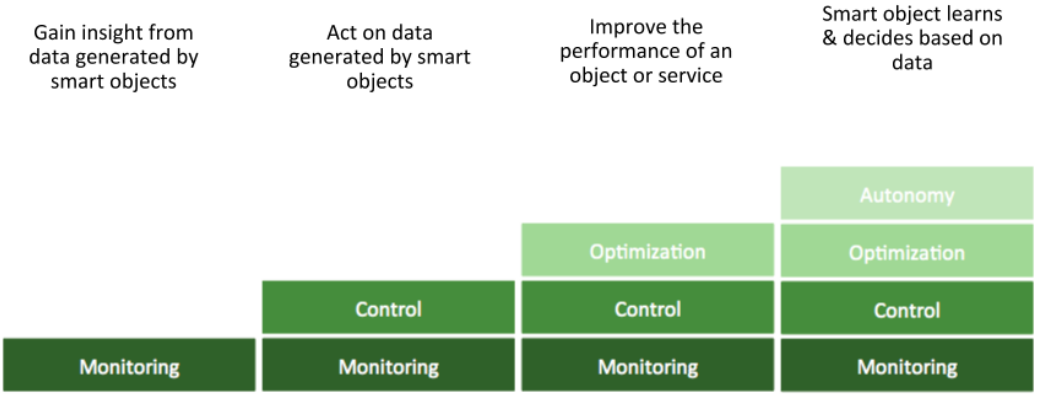


1 https://www.slideshare.net/theilp/pls-2015-the-humble-lamppost

On the right the physical lamppost (hardware-defined version), which consists of a dozen sensors and actuators. In the middle the digital representation of that object (software-defined version). Between them the network and communication to exchange the asset data. This is important because this is where an asset's physical state is transformed into a digital state.

For example, a temperature with a value in °C is converted into a number value with unit '°C.' On the left, we have the IT systems that interact with the digital representative. Because all asset values are digitized, it becomes easy to bundle the sensor data or send a command to an external system. The digitized values can used for various purposes. From simple monitoring to more advanced applications such as building autonomous systems, which operate without any human intervention.

The following image outlines the 4 Internet-of-Things application categories.



Everything connected!

In the early days, home automation systems tried to connect everything. Every light bulb and wall switch got wired to a central 'monolithic' brain, which needed to be programmed by a specialist. It was everything but flexible and became outdated quite quickly because of the continuous evolution of the hardware and software industry (Moore's law, if you remember!). Traditional home automation systems needed careful planning during the construction of a new home and required professionals to install that particular 'home automation' system.

The traditional home automation systems were, for that reason, expensive and suffered from a high level of gadgetry. The latest connected home products can be installed in existing homes without any special skills. These new smart home technology systems do not require you to rebuild your house. They operate wirelessly, for example by connecting to a WiFi network, which by now has become a commodity network in most of our houses.

Most importantly, these new wireless products are not built on one monolithic system. Each connected product (or product line) is managed by its own system, working independently. The fact that it does not rely on one monolithic system allows you to buy just the 'connected' products that make sense and bring value to you without having to connect everything in your home.

Challenges

So what are the challenges in the Connected House 2.0?

Connecting devices

Building a connected house means linking devices via one or another network. That can be a frustrating task and may require multiple steps to perform. Connecting a network printer to the home network can already be a demanding task. So it is not difficult to imagine the level of challenge for someone with limited technical skills to link all of their connected devices in a home. Setup problems are the number one support calls on customer service helpdesks, and even lead to customers returning products.

New initiatives based on autodetect and auto-discover protocols simplify this setup process for connected devices. Although manufacturers take initiatives to make it hassle-free and reduce the number of steps required to connect your device, it remains a challenge. For example, when you purchases a Wi-Fi connected device and plugs it in, your home router could securely send the customer's saved Wi-Fi password to the device, enabling it to join the Wi-Fi network automatically. You can use the Alexa app to scan the device barcode and then follow the simple instructions to complete the setup with fewer steps than before. TP-Link, Kasa, Philips Hue, and Eero have already implemented this in their products.

Connectivity

A short-range connectivity solution such as Wi-Fi can provide solutions within the periphery of the home. Still, once someone wants to extend this into the garden or even further, it gets challenging.

The Ingredients

Introduction

Now that we have an idea of what a connected house stands for, let's see what we can use to build our own. This chapter highlights various platforms, starting with the solutions from some dominant players such as Amazon, Google, Apple and Samsung. Next I discuss a number of open (source) platforms. This is merely a list of useful platforms you can use to build a cost-effective, state-of-the-art and open connected home system.

The term "connected home" might be too narrow, as our lives do not only take place at home but also at work, in our favourite pub to meet with our friends, or travelling between those locations, which we call our places-of-interest. So maybe we should call it our life assistant system, just like the role of the butler as explained in the previous chapter.

Let us start with the ecosystems of some dominant technology players and how they position themselves.

Amazon Echo

Amazon is one of the first technology players to provide a Smart Assistant based on the echo (DOT) hardware combined with the virtual assistant ALEXA which runs in the cloud. It is therefore not surprising that they place their home strategy around this Smart Assistant.

The company has sold more than 100 million devices, making it the leading digital assistant for the smart home. Today they have different variants of the Echo product, featuring the Echo, Echo DOT, Echo show, Echo plus and Echo Spot.

Recently they announced their 4th generation of the Amazon Echo. This latest version has a built-in smart home hub that supports Zigbee devices. This latest version comes for less than 100€ which is a very competitive price including a smart assistant and a Zigbee Home HUB.

In addition to Amazon's own product line, other 'smart speaker' hardware suppliers such as Sonos, Bose, Marshall have embedded Alexa in their products. With the Alexa App you can add WiFi and Bluetooth Smart Devices.

Supported Devices

The Alexa App supports a wide range of connected devices such as the Philips Hue product line (if you use the Hue Bridge), Fitbit Versa 3, Ecobee Smart Thermostat and Smart Camera, the Ring Doorbell,....

With the 4th generation of the Echo DOT a complete ecosystem of Zigbee devices is added to that portfolio of connected devices which can be used with the Amazon Alex App and its Smart Speakers.
It is unclear if Amazon would manufacture a complete portfolio of connected devices besides their Smart Speakers. At the moment there is an Amazon Smart Plug on the market.

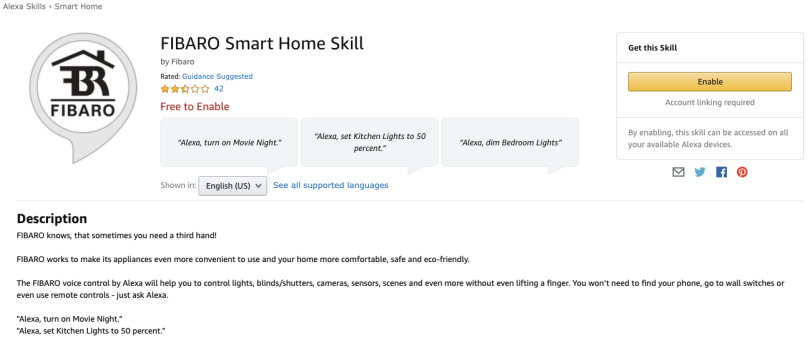
At the Amazon device event in September 2019, the company unveiled several proprietary consumer hardware solutions ranging from wearables including Echo Frames smart glasses and the Amazon Echo Loop, which puts Alexa on your finger over an Alexa microwave and Echo Auto.

Alexa Skills

Alexa skills are essentially apps for your Echo speaker that enable Amazon's voice assistant to do everything from get specific information,, play games and connect smart home devices. Whether it's turning on your Nest smart thermostat or controlling your Philips Hue lamps or your Sonos speakers, the skills enable your Amazon Echo to do just about anything. Amazon has a development programme that supports developers in creating skills to run on Alexa.

Anlexa skills work across the Spart speaker product line. Whether you have the Amazon Echo, Echo Show, or even the Echo Dot, the same skills are available.

Alexa smart home skills are available for all types of connected devices such as Philips Hue, Sonos, Ring, Litter thermostat/camera, Lifx, August thermostat, iRobot home,....



Google Nest

Google Smart Devices

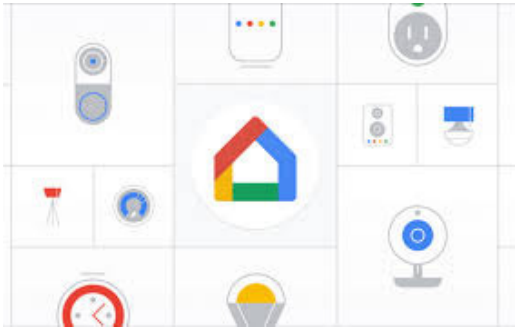
Google builds its home strategy around their Nest products. The first Nest product was the legendary self-learning Wi-Fi thermostat produced in 2011, developed by Nest Labs and designed by Tony Fadell, Ben Filson and Fred Bould.

The Nest thermostat was followed by the Nest Protect smoke and carbon monoxide detectors in October 2013. After the acquisition of Dropcam in 2014, the company introduced its Nest Cam branding of security cameras from June 2015. Google acquired Nest Labs for $3.2 billion in January 2014.

After Google reorganized itself under the holding company Alphabet Inc., Nest operated independently of Google from 2015 to 2018. In 2018, Nest was merged into Google's home-devices unit led by Rishi Chandra, In May 2019, it was announced that all Google Home electronics products will henceforth be marketed under the Google Nest brand.

Under the Nest brand, Google, like other dominant technology players, has its own line of Smart speakers. Starting with the Google Home mini, the Google Home Hub & Home Max and the Google Nest Audio which is the latest addition.

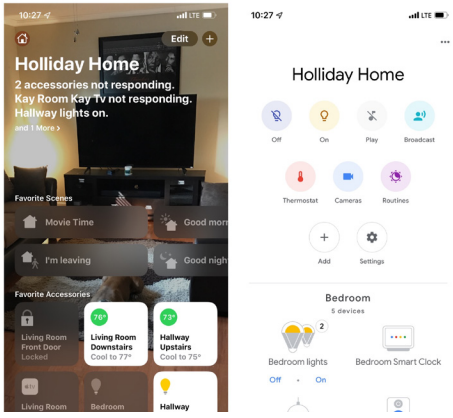
Google Home App



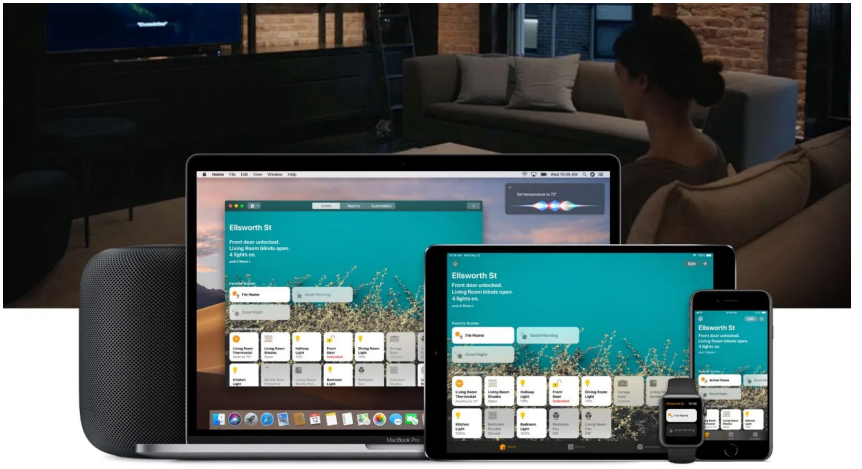
Google has its 'Works With Google' programme, which allows partners to create an ecosystem around the Google home solution. It lets you to control the smart home sensors and devices via the Google Home app and Google's smart speakers using Google Assistant voice commands.

You can turn off the lights, adjust your thermostat, lock your door or make all those things happen at once with a single command, using Google's Routines.

However, there are still many limitations, and many of those integrations are very limited. But the Google Nest Hub is a good, simple interface for managing your home and your digital life and will probably only get better.



Apple HomeKit



What is Apple HomeKit?

HomeKit is a smart-home iOS software framework that enables discovery and control of third party connected devices by an iPhone or other iOS device. While others use a Smart Speaker or a dedicated hardware HUB as the central controller for the home, Apple relies on any IoS device to play that role. Homekit is an integral part of IoS. Currently, Apple does not provide it's own collection of Smart Devices (other than Apple TV & Apple Homepod) for the home, instead it cooperates with Smart Device manufacturers such as Belkin, WeMo, Philips Hue, Nest, Fibaro to name a few. Apple has therefore developed a 'HomeKit' certification programme which allows device manufacturers to make their devices 'HomeKit' compliant.

Works with
Apple HomeKit

Many of today's smart devices are controlled by a companion App. This is in most cases a mobile application which is provided by the device manufacturer such as Belkin WeMo, Nest, Philips Hue. Apple

The Technology

The glue between a system-of-systems architecture

Now that we have dealt with different platforms, let us dive into the chapter on technology and see how these systems can interact with each other. We will look at three technologies: APIs, Webhooks and MQTT.

API's

A first technology to build a systems-of-systems architecture.

What is an API?

API stands for Application Programming Interface. In an Internet-connected world, humans use web and mobile applications, systems and applications use APIs. Websites and APIs both do the same things, like return data, content, images, video, and other information.
Web APIs are a set of rules for interacting with a web server, with the most common use case being data retrieval. APIs provide mechanisms for users to access and manipulate data stored by the API provider. The user makes a "request" to a web server, that web server accesses a database (which contains the data), and returns it to the requester in a "response".

APIs are not a specific service or tool, they are part of a system, and like the web you get something in return with every request. Instead of getting HTML back with every request, you get JSON, XML, and CSV - providing structured, machine-readable information that can be used by other systems and within other applications with very little human help.

What Are APIs Used For?

While APIs are primarily used by desktop, web, mobile, and other application developers, they are also used by non-developers to work with services like IFTTT, Zapier, and the growing number of low-code or no-code solutions out there – such as Postman[1]
Every company that has a modern application out there has APIs – these APIs might not be easily found. All the big tech company names you know like Facebook, Twitter, Google, Microsoft, and others – all have APIs. Any company who uses common services like WordPress, Quickbooks, SalesForce, and other common applications, technically also have APIs.
If your website runs on WordPress, your website has an API – and if you aren't aware of it, then you

1 https://www.postman.com/

aren't getting the opportunity to put it to use. APIs are ubiquitous, and underneath the surface of everything we do online today – you just may not have been fully aware of it until now.

As an example, let's look at OpenWeather, the API service that provides weather forecasts around the globe for any use case.
It provides current weather data, forecasts and historical data to more than 2 million customers, including Fortune 500 companies and thousands of other businesses globally.

If you want to know the weather in the city of London you can execute the following API call:

```
http://samples.openweathermap.org/data/2.5/history/city?q=London,UK&appid=b1b15e-88fa797225412429c1c50c122a1
```

The API response is as follows:


```
{
  "message": "",
  "cod": "200",
  "city_id": 2643743,
  "calctime": 0.0875,
  "cnt": 3,
  "list": [
    {
      "main": {
        "temp": 279.946,
        "temp_min": 279.946,
        "temp_max": 279.946,
        "pressure": 1016.76,
        "sea_level": 1024.45,
        "grnd_level": 1016.76,
        "humidity": 100
      },
      "wind": {
        "speed": 4.59,
        "deg": 163.001
      },
      "clouds": {
        "all": 92
```

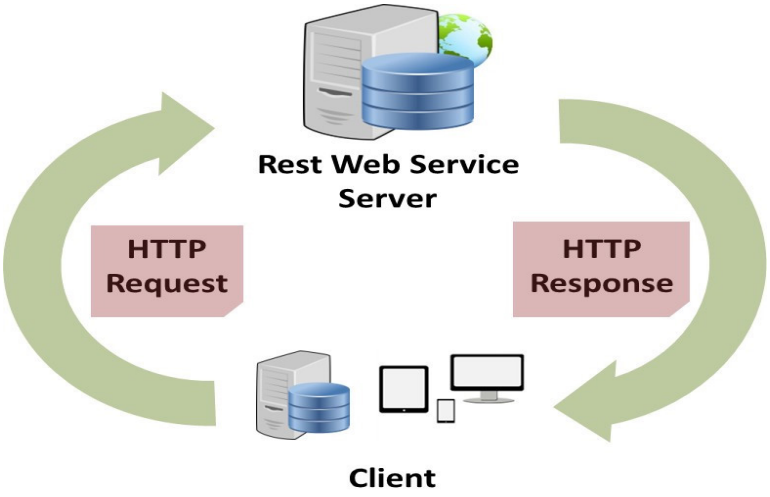
```
  },
  “weather”: [
    {
      “id”: 500,
      “main”: “Rain”,
      “description”: “light rain”,
      “icon”: “10n”
    }
  ],
  “rain”: {
    “3h”: 2.69
  },
  “dt”: 1485717216
}
```


Are APIs for you?

APIs describes how data is exchanged, content is published, media are consumed and algorithms are used on the web today. APIs are how you access your social data, your photos, your accounting information and much more.
APIs are often seen as highly technical, which can be true. However, many APIs are accessible to anyone curious enough to look behind the curtain of the web. If you've ever clicked "view source" on a website, APIs are for you. If you want to know how to get Tweets into a spreadsheet so you can play with social data, APIs are for you. If you're interested in how your Nest thermostat works with your mobile phone applications, then APIs are for you.

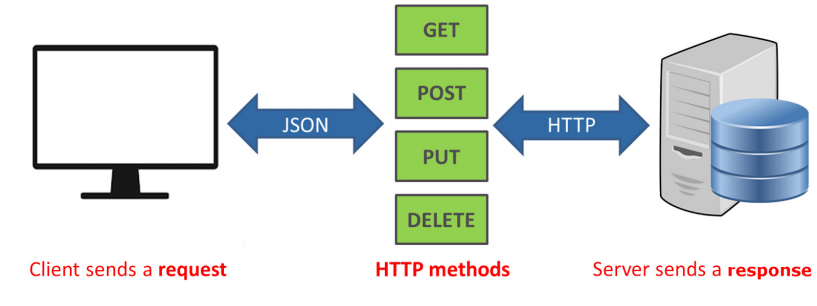
Postman is working hard to make APIs more visible and accessible to both developers and non-developers. If you want to know more about APIs, I recommend that you start by downloading the Postman application and find one or two interesting API collections in the Postman API network to start playing. You never know what you might learn along the way!

REST-API



Webservices are purpose-built web servers that support the needs of a site or other application. Client programs use Application programming interfaces (APIs) to communicate with Web services. Generally, an API exposes a range of data and functions to facilitate interaction between computer programs and enable them to exchange information. The API is the (inter)face of a web service that listens and responds to client requests.

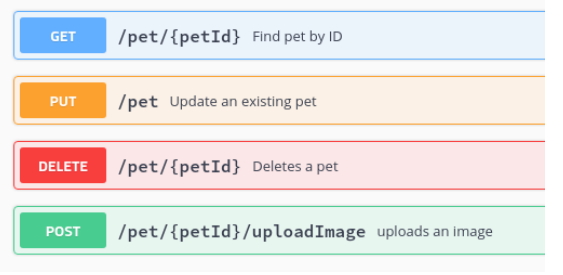
The REST architecture is commonly applied to the design of APIs for modern web services. A web API conforming to the REST architectural style is a RESTful API.



"Web resources" were first defined on the World Wide Web as documents or files identified by their URLs. However, today they have a much more generic and abstract definition that encompasses everthing or entity that can be identified, named, addressed, or handled, in any way whatsoever, on the Web. In a RESTful Web service, requests made to a resource's URI will elicit a response with a payload formatted in HTML, XML, JSON, or some other format. The response can confirm that some alteration has been made to the stored resource, and the response can provide hypertext links to other related resources or collections of resources. When HTTP is used, as is most common, the operations (HTTP methods) available are GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS and TRACE

Take as an example a PET registration database web service.

With GET /pet/{petId}, you can retrieve the information about a pet by its ID.
PUT /pet allows you to update an existing pet. DELETE /pet/{petId} will remove the pet with a given ID from the registration database. POST /pet/{petId}/uploadImage allows you to upload a picture for a pet with a given ID.



This simple example shows how you can exchange information between a client and a database with an API Web service and a listening and responding to client requests.

Webhooks

We have just learned that we can use REST APIs to exchange information. There is one caveat; the client needs polling to get the information.

Within IoT it may be more convenient that this happens automatically when an event occurs (e.g. change of a sensor value). Webhooks offer a way to send notifications to an external web server when specific actions take place on a repository or organisation.

A webhook is an event-driven system that calls the client when an event occurs that the client is interest-

ed in. This is contrasted with polling, which is when the client must continuously and inefficiently check if a certain event has occurred.

An appropriate analogy is if you organise a party and are waiting for a pizza. A “Polling” solution would be to leave the party and open the front door every 5 minutes to see if your pizza has arrived. A “Webhook” solution would be to stay at the party and tell the delivery boy to ring the bell when he arrives.

Webhook relay

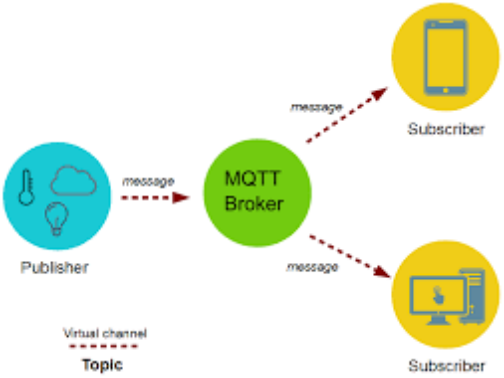
In a connected home context, it can be a challenge to expose web services running locally in the home to the internet. Most residential Internet broadband solutions do not provide a fixed public IP. You can work around this by using one of the free to use dynamic DNS services that exist, such as duckDNS. Yet in that case you have the security challenge of having to open some ports to allow incoming traffic.
A better solution might be the use of a webhook relay.

MQTT

Another way of interconnecting in a system-of-systems architecture is by using MQTT.

MQTT (MQ Telemetry Transport) is an open lightweight, publish-subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP; however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. It is designed for connecting power-constrained devices where a “small code footprint” is required that run over low-bandwidth networks.

MQTT is not new. It exists for over a decade, the advent of M2M (machine to machine communications) and Internet of Things (IoT) made it a popular protocol. Enterprise cloud platforms such as Amazon Web Services, Microsoft Azure, and IBM Watson expose their IoT PaaS through MQTT.



Origin of MQTT:

MQTT was created way back in 1999 by two engineers — Andy Stanford-Clark (IBM) and Arlen Nipper (Eurotech). They had to invent a new protocol for connecting oil pipelines over unreliable, satellite networks. The motivation for designing MQTT was to create a lightweight and bandwidth-efficient protocol that was data-agnostic with support for multiple levels of Quality of Service (QoS). Interestingly, even today, those are the same reasons for which MQTT is chosen for implementing IoT solutions.
In 2011, IBM and Eurotech donated MQTT to the proposed Eclipse project called Paho. In 2013, it was submitted to OASIS for standardization. The latest version of the protocol specification, 3.11 has become an OASIS standard.

The Hardware

Now we know which platforms you can use to build your connected home based on a systems-of-systems architecture and which technology is available to integrate them, its time to look at the hardware.

Which hardware is out there we can use to build our connected home?

The Smart HUB

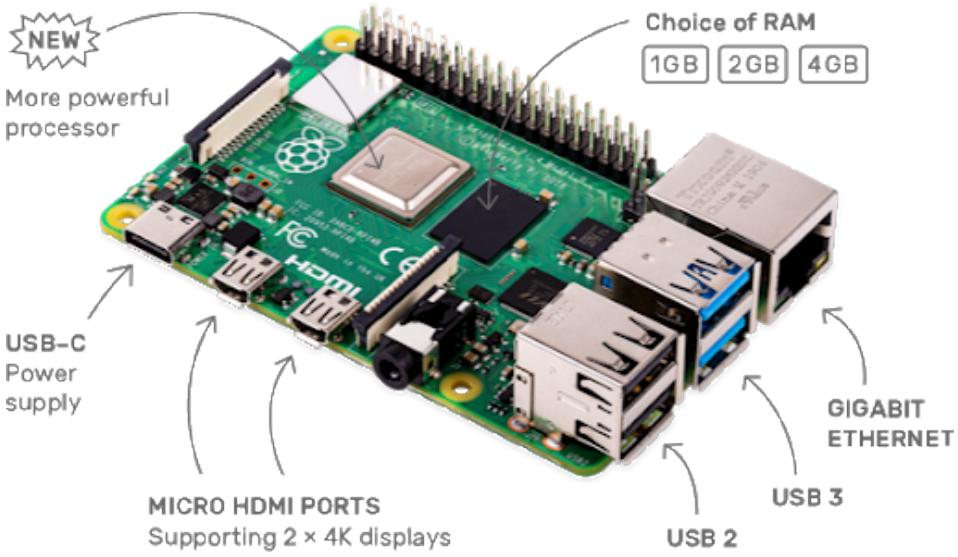
The first hardware component we are going to cover is the HUB (or gateway) which can manage the different networks and protocols. Companies such as SmartThings (acquired by Samsung), Fibaro and Philips Hue all deliver a HUB, but each of them supports only one or a limited number of networks and protocols, suited for their ecosystem. SmartThings supports Zigbee and Z-Wave [1], Fibaro focuses on Z-Wave and the Apple Homekit protocol While Philips Hue uses Zigbee to communicate with their devices.

Depending on what you want to achieve, these HUBs will all perform fine for a particular use-case, but we don't want to be constrained to one protocol or vendor and only can use their ecosystem. It would be a pity that you can't use Sonos Speakers or Ikea's Tradfri smart lights or the nice looking Fibaro Wall Plugs for that matter. You want to use them all!

The good news is, you can! As outlined in chapter 2, openHAB is the ideal software platform to build our Home HUB. Although different hardware can be used, the most popular and low-cost solution is using a single board computer such as the Raspberry Pi for our Smart HUB.

The Raspberry Pi is a single-board computer with lots of resources. It has the size of a credit card designed and manufactured in the UK with the initial intention of providing a cheap computing device for education. Since its release, however, it has grown far beyond the educational scene. The Raspberry Pi's commercial release was in February 2012 with the Raspberry Pi 1. Since then, the board has gone through several revisions. Currently, we are at revision 4. You can buy the Raspberry Pi 4 starting from 39.95€, depending on its memory size. The 1 GB version is good enough for your connected home project, but you can opt to go for more memory if you intend to run other applications aside from it.

1 See the technology chapter for more information about Z-Wave and Zigbee.



The Raspberry Pi 4 specs:

- CPU – Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM – 1GB, 2GB or 4GB LPDDR4-2400 SDRAM (depending on model)
- WiFI – 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Ethernet – Gigabit
- USB – 2 USB 3.0 ports; 2 USB 2.0 ports
- GPIO header – Raspberry Pi standard 40 pin
- HDMI – 2 × micro-HDMI ports (up to 4kp60 supported)
- Display port – 2-lane MIPI DSI
- Camera port – 2-lane MIPI CSI
- Audio – 4-pole stereo audio and composite video port
- Storage – Micro-SD card slot for loading operating system and data storage
- Misc – H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode), OpenGL ES 3.0 graphics
- OS – Debian Linux 10 based

Next to the Raspberry Pi, you need some additional components.

You need first of all an SD card. The SD card is required to install your operating system and software on it. In our case, we will install OpenHAB installed on it. A complete ready-to-use image is available for that purpose.

Second, you will need a Power Supply. Go for an official power supply. Most problems occur when not enough power is delivered to the board, caused by an inefficient power supply. Especially when the radio's start transmitting, the system might behave strange when using undersized or cheap power supplies.

You might also want a case to protect everything from the outside. A Keyboard and monitor is only handy during installation. Afterwards, the Raspberry Pi runs headless.

If you want to make use of Z-Wave devices, you will need a Z-Wave controller. A suitable Z-Wave controller which connects to one of the USB ports of the Raspberry Pi is the Aeon Labs Aeotec Gen 5 Z-Wave USB Interface. You can find one for less than 50€.

This Z-Wave controller has a battery on board which allows you to unplug it from the Raspberry PI and bring it close to a new Z-Wave sensor in your home to run the discovery service. Pretty handy!

It is relatively easy to build your own (Home) HUB thanks to the excellent community software and hardware out there. It is not only an open system able to support a lot of devices, it is also fun to build your own hardware solution.

Mandatory is that the HUB can be part of a system-of-systems architecture. It will give you more flexibility as technology evolves rapidly and allow you to use new features when they get released on any of

those platforms.

The first recipe further down in chapter 5 covers the installation of your Smart Home HUB. If you want, you can jump to there and try out that recipe and come back here to see what is next.

Off-the-shelf sensors

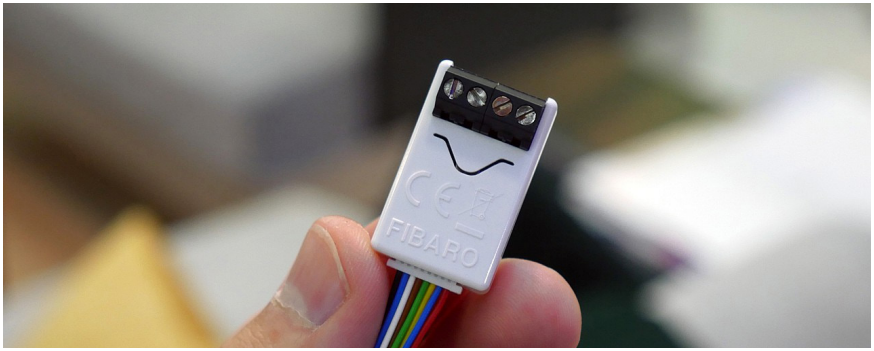
A decade ago, sensors and actuators needed to be wired to a central control unit, making it only possible to install in new homes or when substantial home renovations took place. The last year's sensors are wireless using a meshed network protocol such as Z-Wave or Zigbee. Most of these wireless sensors are designed for low power consumption so they can run on a single battery load for years. The capabilities to communicate wireless and able to run on a single load battery makes them ideal candidates to retrofit them in existing homes and accessible for everyone.

Zwave devices

There are a lot of Z-Wave device manufacturers for the home. A company that delivers Z-Wave devices is Fibaro. Fibaro is one of my favourites because they are not only working as expected; they also have an excellent design. This Polish company has a great portfolio of the sensors you can use to build a connected home.

Fibaro Smart Implant

One of my favourite Z-Wave devices is the Fibaro smart implant. This device is ideally suited to retrofit existing items such as a garage door, alarm system, HVAC installations, audio equipment, etc...



The small form factor (29x18x13mm) makes it fit in most existing product cases. It serves 2 analog/digital inputs (0..10 VDC) and 2 potential free outputs. Additionally it can serve upto 6 DS18B20 or 1 x DHT22 temperature sensors.

The smart implant can cope with a power supply between 9and 30 V DC.

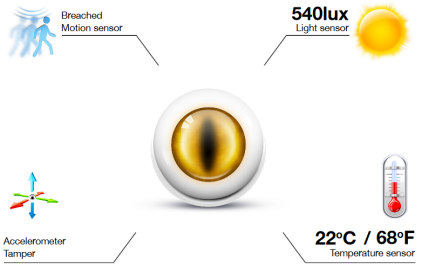
Fibaro Wall Plug

The Fibaro Wall Plug is a smart switch with power metering for electrical devices.

The FIBARO Wall Plug comes equipped with a power measurement feature. It helps you effortlessly identify the most energy-hungry pieces of hardware or monitor energy usage in particular rooms or during particular periods.

The crystal LED ring changes its color according to the amount of power used by the plugged device

Fibaro Motion Sensor



The Fibaro motion sensor is a battery powered multi-sensor device. Apart from sensing motion, the device also measures temperature and light levels, offering you a more complete motion detection solution. The accelerometer detects change in location or any attempt at opening its casing.

Sensative Stirps

The Sensative strips are only 3mm thick and can be used indoor or outdoor. The Strips Guard is a revolutionary magnetic sensor designed to monitor and protect windows, doors, and valuables. They're so thin that they can be used in numerous unique use-cases such as mounting invisibly to artwork and cabinets to monitor activity. Using an adhesive strip, mounting is quick and easy. Just set and forget with the custom designed battery that lasts up to 10 years.

besides the strip guard, thers is also the strip drip to detect water leakage and the strip confort which measure ambient light and temperature.

Note: The strips are also available in a LoRaWAN version.

RECIPES for the connected house

Let's get into the practical side of things. This is the **DO** part. In this chapter, you can find a collection of carefully selected recipes for the connected house based on the ingredients and technology outlined in this cookbook.

The recipes are in chronological order, but feel free to crawl through it and start executing them in the order you want. If there is a dependency on a previous recipe, this will be mentioned.

The recipes are divided into 4 categories.

The first category contains a collection of basic recipes. These are the recipes that form the basic building blocks and are typical prerequisites for other recipes.

A second category includes recipes for connecting appliances. These recipes include both commercial 'off-the-shelf' devices and recipes to build your own connected devices.

A third category includes recipes to build great user interfaces for web and mobile applications. Human interfaces, such as voice control and text-to-speech, also fall under this category.

The fourth and final category contains recipes that add automation to your home. Sample recipes include a calendar planner for your HVAC and home and away scenarios.

All recipes are accesibble on the website *theconnectedhouse.org* allow you to easily copy past commands and code. Some recipies in this cookbook include QR-codes to download Code blocks.

Let's start with the basic recipes.

BASIC RECIPES

VOOR 6-8 PUNTEN

CRÊPETAART MET SINAASAPPEL, CHOCOLADE EN COINTREAU

EEN BIJZONDER LUXE *crêpetaart met een heerlijke ganachevulling. De crêpes kun je een dag eerder bakken en afgedekt met plasticfolie in de koelkast bewaren, zodat je de taart alleen in elkaar hoeft te zetten als het tijd is voor het feest!*

CRÊPES
4 eieren
250 ml melk
250 ml water
100 g patentbloem
1 tl zout
15 ml kristalsuiker
geraspte schil van 1 sinaasappel
75 g boter + extra voor het bakken

CHOCOLADEGANACHE
200 g pure chocolade (70%)
100 g melkchocolade
300 ml slagroom
evt. 25 ml Cointreau (sinaasappellikeur, kan worden vervangen door 25 ml slagroom gemengd met de geraspte schil van ½ sinaasappel)

200 ml slagroom
geraspte chocolade
evt. 50 g verse frambozen

CRÊPES
1. Klop de eieren, melk, water, bloem, zout, suiker en sinaasappelrasp in een kom. Klop tot alle klontjes verdwenen zijn. Laat het beslag bij kamertemperatuur circa 30 minuten rijzen.
2. Smelt de boter en laat afkoelen.
3. Meng de boter door het beslag. Verhit een koekenpan met een diameter van 15-20 cm en doe er een klontje boter in.
4. Bak dunne crêpes en leg ze op een bord, het liefst met bakpapier ertussen, zodat ze later gemakkelijk gescheiden kunnen worden. Laat helemaal afkoelen.

CHOCOLADEGANACHE
1. Hak de chocolade in kleine stukjes en doe ze in een hittebestendige kom.
2. Verwarm de slagroom in een pan. Haal de pan van het vuur voordat het begint te koken en giet de warme room over de chocolade. Schenk de Cointreau erbij en laat een paar seconden staan. Roer daarna tot de chocolade volledig gesmolten is en er een glanzende ganache is ontstaan.
3. Schenk 100 ml van de ganache in een aparte kom (die wordt aan het eind gebruikt, ter garnering) en laat de rest in de pan. Laat de ganache bij kamertemperatuur staan tot hij een smeerbare consistentie heeft.

DE TAART SAMENSTELLEN
1. Leg de eerste crêpe op een schaal en strijk er een dunne laag ganache over. Herhaal tot je alle crêpes en ganache hebt gebruikt. Zet de taart enkele uren in de koelkast als hij instabiel lijkt.
2. Klop de slagroom stijf en schep of spuit deze op de taart.
3. Verwarm de bewaarde ganache voorzichtig in de magnetron (of au bain-marie in een kom) tot deze weer enigszins vloeibaar is – bij mug echter niet helemaal warm worden. Schenk de sans over de taart en decoreer met geraspte chocolade en desgewenst frambozen.

BASIC RECIPES

Building a smart HUB 139

Installing Node-RED 147

Installing MQTT 151

Connect to an external MQTT Broker 155

Alexa Integration 161

RECIPE

Building a smart HUB

Time to cook 120min
Difficulty +++++

Ingredients

- Raspberry Pi 3B+ (other versions might also work)
- Power supply for your Raspberry PI
- SD card (16 GB Recommended)
- Raspberry Pi case
- OpenHABian (free to download)

Prerequisites

Tools

- Etcher (free to download)
- Putty (free to download)

Intro

If you have gone through the previous chapters, then you already know that OpenHAB is a great open source home automation platform that fits perfectly into our system-of-systems architecture for building a connected home. OpenHab can run on a low-cost hardware Single board computer (SBC) such as the popular Raspberry Pi. The software version of openHAB that runs on a Pi is called openHABian.

The smart HUB functions as an edge gateway, Besides the local processing for our home automation, it connects all types of devices over various protocols and networks, even those that do not travel the Internet such as Bluetooth and exposes them to other platforms in our home or in the cloud.

This recipe covers the installation of openHABian V1.5 on a Raspberry-Pi Model 3B+.

OpenHAB did a wonderful job by providing a ready to use image which includes the Operating System (Debian linux), the openHAB platform and some interesting add-ons. This makes the installation more or

less straightforward. As technology progresses fast, probably the versions and procedures have evolved already. Therefore i suggest to follow the procedure on this link [1]

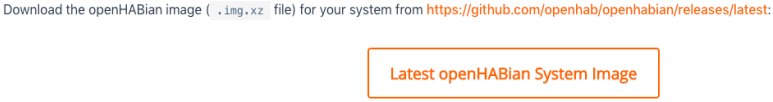
The Building Plan

Preparing your image

let's burn the image on our SD card:

- Go to the download section of the OpenHAB website [2]
- Select the Raspberry Pi as system on the download page.
- Select the version you want to install. The choice is between Stable or Snapshot. choose stable for your production environment.

- Download the Latest openHABian system image by clicking on the link as outlined in the image below



- Select the 32-bit version
- Once you have downloaded your openHABian image, you should flash it on an SD card for your Raspberry Pi. OpenHab proposes to use Etcher, which is a popular tool for burning your SD card. Just follow the instructions on the openHAB download page.
- Next insert the SD card in your Raspberry Pi and boot it. The installation will take approx. 45 minutes.

1 https://www.openhab.org/download/
2 https://www.openhab.org/download/

Best is to connect your Raspberry Pi to your home network using a LAN cable. You can use a WiFI connection if your Raspberry Pi supports it, but as the Raspberry Pi will be used as a main hub to connect all kinds of IoT devices and services, it will be more stable when it has a LAN connection and not depend on a less-stable WiFi connection.

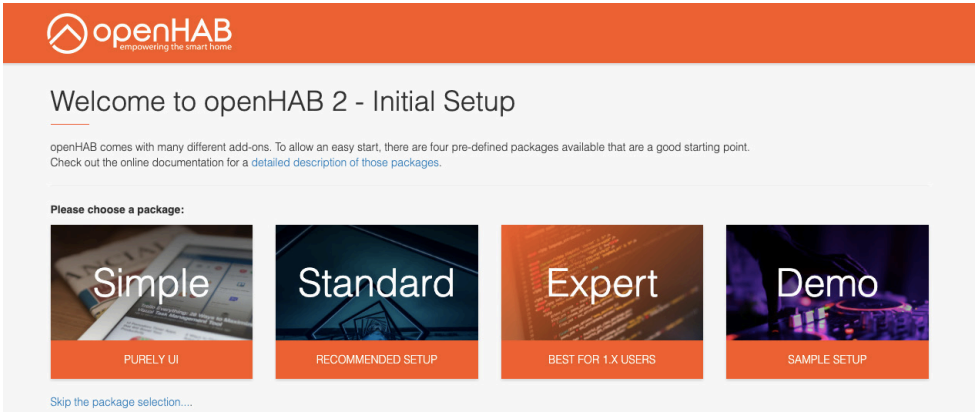
When the installation is finished, you will be able to browse to your Raspberry Pi. OpenHAB has a frontend which you can reach on port 8080 and includes one or more applications, depending how you set up your device. verify your Internet modem to see which IP address your Raspberry Pi received and use that in your browser.

Example: http://your-ip-address:8080

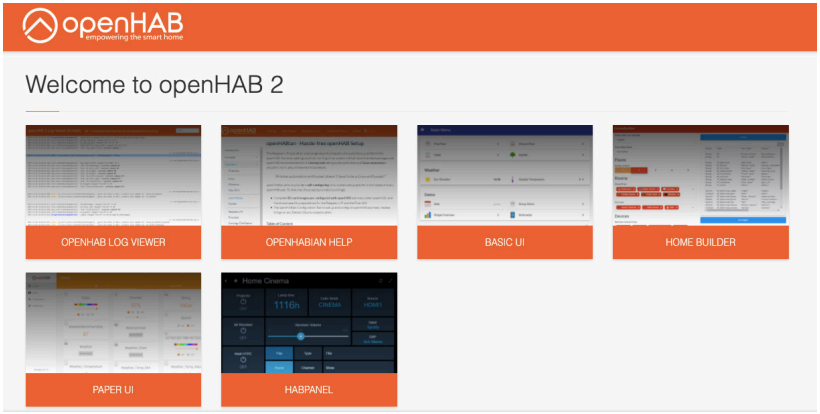
If your raspberry pi got the IP address 192.168.0.18. you should enter: http://192.168.0.18:8080

You see the Initial Setup screen from openHAB 2. There are 4 pre-configured packages. The Standard package is the most recommended and good for our purpose.

- Select the Standard package



After selecting the standard package you should see the following applications in the OpenHAB welcome screen. You might need to refresh your browser a few times and wait a bit until all applications appear.



Change the default password

After the installation of openHAB, the minimum you should do to secure your openHAB installation is change the default password of the openhabian user.

The default username and password: **openhabian**

You can do this as follows:

- Login with the user openhabian using an SSH connection onto your Raspberry Pi. (You can use a tool like Putty to establish an SSH session to your raspberry Pi for Windows or use the Terminal program on a Mac OS)
- Type: **passwd**
- Enter your **current password**
- Enter a new password for your account

This is it. You have executed your first recipe! The outcome is that you have a system which is ready to connect different kinds of sensors using different protocols.

The next section is optional, but highly recommended if you want to control your Smart HUB remotely or

want to use a voice assistant.

Installing OpenHAB Cloud Connector

The openHAB Cloud Connector allows connecting the local openHAB runtime to a remote openHAB Cloud instance [1], such as myopenHAB.org [2], which is an instance of the openHAB Cloud service hosted by the openHAB Foundation [3].

The openHAB Cloud service (and thus the connector to it) is useful for different use cases:

- It allows remote access to local openHAB instances without having to expose ports to the Internet or to require a complex VPN setup.
- It serves as a connector to Google Cloud Messaging (GCM) and Apple Push Notifications (APN) for pushing notifications to mobile phone apps.
- It brings integration possibilities with services that require an OAuth2 authentication against a web server, such as IFTTT or Amazon Alexa Skills

So, if you want to connect to your Smart HUB via the Internet, this is something you want to install. It is a secure way to connect to your Smart HUB from the Internet.

UUID and Secret

To authenticate with the openHAB Cloud service, you need to install the add-on in openHAB.

- Open PaperUI from the openHAB url.
- In the left menu, selct **Add-ons**
- Select **MISC** in the top menu
- Goto **openHAB Cloud Connector**
- Click **install**

The add-on generates two values. These values need to be entered in your account settings of the openHAB Cloud service.

File	Regular Installation	APT Installation

1 ithub.com/openhab/openhab-cloud/blob/master/README.md
2 https://www.myopenhab.org/
3 https://www.openhabfoundation.org/

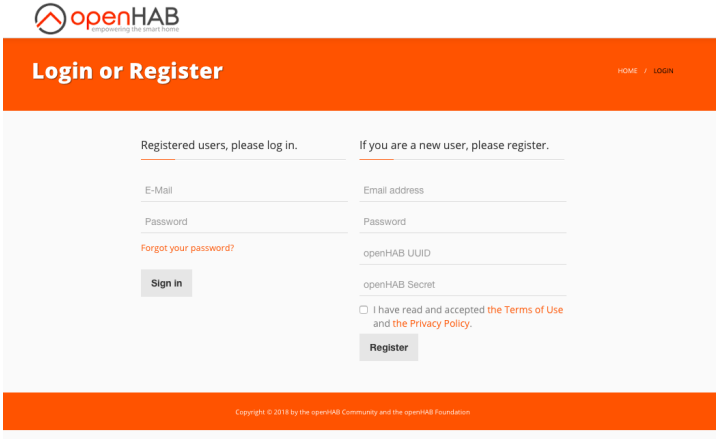
UUID	userdata/uuid	/var/lib/openhab2/uuid
Secret	userdata/openhabcloud/secret	/var/lib/openhab2/openhabcloud/secret

```
[10:40:09] openhabian@openhab:~$ cat /var/lib/openhab2/uuid
540da818-ac6c-4f02-a9a3-8274fa4fb0b8
```

```
[10:43:02] openhabian@openhab:~$ cat /var/lib/openhab2/openhabcloud/secret
NsluLVmZauDSgXocPJ88
```

The first one is a unique identifier, which allows you to identify your runtime.
You can think of it as something similar like a username for the cloud authentication. The second one is a random secret key which serves as a password. Both values are written to the local file system. If you lose these files for some reason, openHAB will automatically generate new ones. You will then have to reconfigure UUID and secret in the openHAB Cloud service under the My account section.

Next, Goto myopenhab.org and register/login an account.



Configuration

When talking to people that want to set up their home automation system at home, they are very suspicious to open their systems to others, which is quite understandable. On the other hand, to build a

system-of-systems approach and get the most out of it, you need to peer with other systems.

You can (and should) limit the access to third party systems. This can be done by only expose the items you want to share with the other platforms. You can do this in OpenHAB in the Paper UI under Configuration -> Services -> IO -> openHAB Cloud:

Alternatively, you can configure the settings in the file conf/services/openhabcloud.cfg.

Configuring a static IP address for your Smart HUB

Most home network routers provide a DHCP address. This means that the router will distribute an IP address to IP capable devices within a certain range. The advantages is that you don't have to set it up manually, but the disadvantage is that this IP address might change in future (when the lease expires). This behavior is something you might want to avoid on your Smart HUB especially if you would refer to it's IP address in a system-of-systems architecture.

To assign a static IP address to your Raspberry Pi, proceed as follows:

- Login with the user **openhabian** using an ssh connection onto your raspberry pi
- Edit the dhcpcd.conf file
    - **sudo nano /etc/dhcpcd.conf**
- Scroll all the way to the bottom of the file and add the following lines of code

```
interface eth0

static ip_address=192.168.0.10/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```

Note: You'll need to edit the numbers in the snippet so they match your network configuration

interface: This defines which network interface you are setting the configuration for.
static ip_address: This is the IP address that you want to set your device to. (Make sure you leave the /24 at the end)
static routers: This is the IP address of your gateway (probably the IP address or your router)
static domain_name_servers: This is the IP address of your DNS (probably the IP address of your router).

You can add multiple IP addresses here separated with a single space

- To exit the editor, press **ctrl+x**
- To save your changes press the letter **Y** then hit enter

Now all you need to do is reboot, and everything should be set!

- **sudo reboot -h 0**

You can double check by typing

- **ifconfig**

And checking the interfaces IP address

What's next?

A good next step is to add some devices to your Smart HUB. Have a look at recipe 'Using Zwave devices' further in this book.


Recipes to connect
your objects for the
connected house

Connected devices

Using Z-Wave devices 169
Retrofit the garage door 176
WiFi connected switch & energy monitoring 181
Connecting Sonos smart speakers 193
Connecting Flic Buttons 196
Building a connected thermostat 203
Building an (outdoor) plant sensor 209
Monitor a solar boiler 219
Connect an IKEA TRADFRI smart light bulb 225
Connect an IKEA TRADFRI remote control 241

RECIPE

Using Z-Wave devices

Time to cook 120min
Difficulty ++++++

Ingredients

- An Aeon Z-Wave Stick
- A Z-Wave Fibaro Wall Plug

Prerequisites

- Cook the recipe: Building a smart HUB (Basic recipe)

Tools

- Putty (free to download)

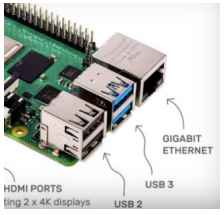
Intro

Z-Wave is a wireless communication protocol mainly used for home automation purposes. It is a network that uses energy-efficient radio waves to communicate from device to device, making it possible to wirelessly monitor and control household appliances and other devices such as lighting, security systems, thermostats, windows, locks, etc.
The Smart Home HUB we built in one of the other recipes is capable of supporting Z-Wave appliances. Actually, one of the most popular bindings used on the Smart Home HUB is undoubtedly the Z-Wave binding.
The mesh network is managed by a Z-Wave controller. In this recipe we are going to use an Aeon Z-Wave stick to connect to our Smart Home HUB via USB, but you can use another type of Z-Wave controller for your project.

The Building Plan

First insert the Z-Wave controller in one of the USB-ports on the Raspberry-Pi.

The number and type of USB ports on Raspberry Pi depends on the model. The Raspberry Pi Model B is equipped with two USB 2.0 ports; the B+, 2B, 3B and 3B+ have four USB 2.0 ports. The Pi 4 has two USB 2.0 ports and two USB 3.0 ports. On a Raspberry Pi model 4, insert the Z-wave controller in one of the USB 3.0 Ports.



Adding Z-Wave to openHAB

Add the Z-Wave binding on the Smart Home HUB. The PaperUI tool which is available on the welcome page of your openHAB installation on the Smart Home HUB allows you to configure your openHAB environment, including adding all kinds of bindings.

- Open **Paper UI**
- Select **Add-ons** from the left menu
- Select **Bindings** from the top menu
- Search or scroll down for the **Zwave Binding**
- Click **Install**

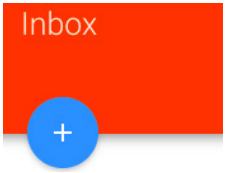
More information about the Zwave binding can be found here [1]

1 https://www.openhab.org/addons/bindings/zwave

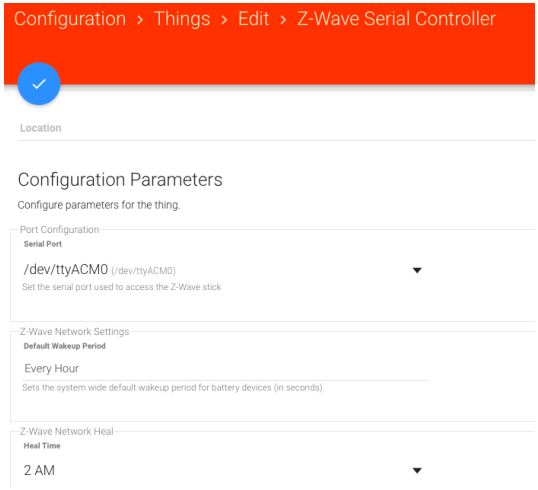
```
https://www.openhab.org/addons/bindings/zwave/
```

After installing the Zwave Binding, we need to configure the Zwave Binding:

- Click on **Configuration → Things** in the left menu of Paper UI
- Click the **+** sign in the Inbox

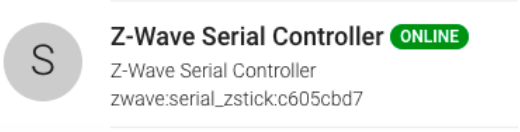


- Select **Z-Wave**
- The AEON Zwave serial controller should be auto-detected
- Add it as a 'Thing' and edit the configuration.
- Select the serial port on which the Aeon Zwave Stick is mounted



Now, the Zwave Serial Controller should come ONLINE as shown in the following image:

- Click on **Configuration** → **Things** -> Zwave node



⚠ It is important to select the correct USB port. This is the port on which the Zwave controller is connected. In my setup this is /dev/ttyACM0. Remember we are running linux. If you are not sure which port to look for, login into your RPI using an SSH connection and go to the /dev folder. List all ports using the command line instruction:

- cd /dev
- sudo ls -all

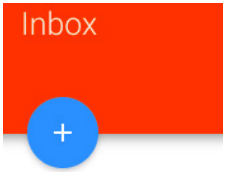
Then plug the Zwave controller in one of the USB ports on your RPI and re-run the command line instruction. Compare both lists and you should be able to find the port on which the Z-Wave controller is attached.

Including your first Zwave Device

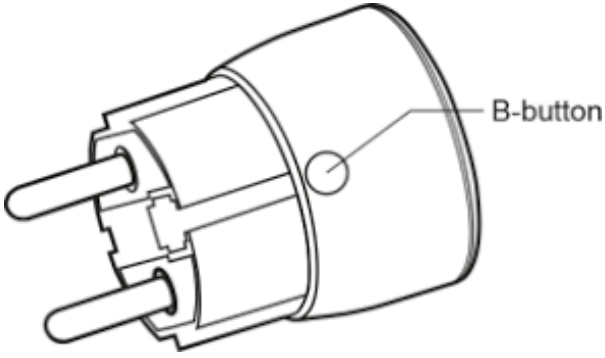
We are now ready to add our first Zwave Device. As an example we will add a Fibaro Wall Plug. This is a nice looking connected wall Plug which we can switch on and off is also capable of measuring the energy consumption. An led ring can be used to visualize the device state like on/off or change the color depending on the power consumption.

Z-Wave includes an autodiscover protocol. When you set the Z-Wave controller we have previously configured in inclusion mode, it can discover new Z-Wave devices. The same process can be used to exclude a Z-Wave device.

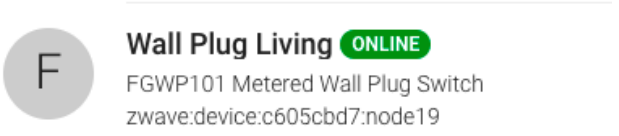
- Click on **Configuration → Things** in the left menu of Paper UI
- Click the **+** sign in the Inbox



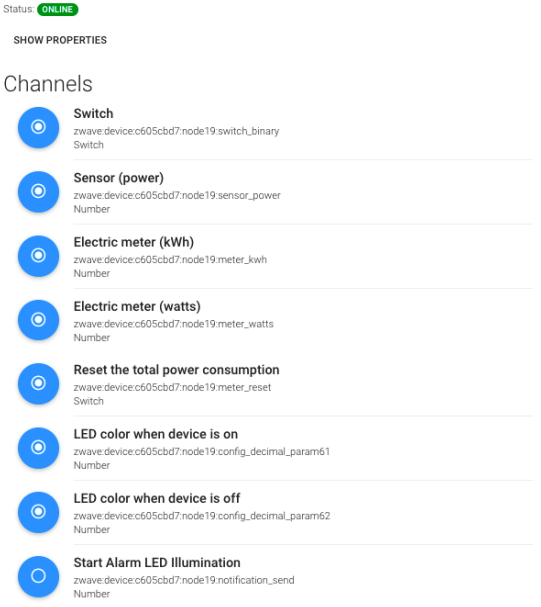
- Select **Z-Wave** → (This will put the Zwave controller in inclusion mode)
- Plug the device into a socket nearby the main Z-Wave controller.
- Quickly, triple click the B-button located on the casing.
- Wait for the adding process to end.



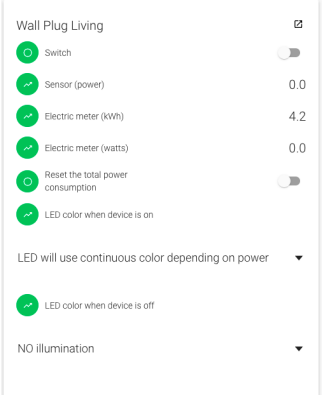
- A new Z-Wave node will appear in the openHAB inbox. openHAB includes a large set of pre-configured Z-Wave devices in it's database. The Fibaro Zwave wall plug is one of them. It will recognize the Z-Wave device and the new Z-Wave node will present itself as a Fibaro wall plug including all pre-configured channels (channels can be seen as device assets, a collection of sensors and actuators).
- You can now add the newly discovered Z-Wave node as a 'Thing' to your configuration. The result could look like this:



The list below shows all the channels that have been discovered under the Z-Wave wall plug. The first channel is the actual actuator which turns the Z-Wave wall switch on or off. The second, 3rd and 4th report the power consumption.



You can control your Things from within openHAB paperUI. Go to Control in the left menu and you will find an overview of all your 'Things' for which you have channels. For the Z-Wave wall plug described above the UI looks as follows.



When you toggle the switch button, the wall plug will be turned on/off. The sensor (power) and Electric meter will report the power consumption.

This interface is nice for testing out your devices and a first impression. In this Cookbook you will find other ways to build user interfaces or use for example Alexa to control your wall plug!

RECIPE

Retrofit the garage door

Time to cook 120min
Difficulty +++++

Ingredients

- A Fibaro Smart Implant
- An Aeontech zwave controller

Prerequisites

- Cook the recipe: Building a smart HUB (Basic recipe)
- NodeRed installed with the following nodes:
  - Node-red-contrib-ui-led
  - Node-red-dashboard

Tools

- Screwdriver
- Cabling

Intro

This recipe is an example of how you can retrofit a garage door which is equipped with an electric motor. This way your garage door becomes a connected garage door!.

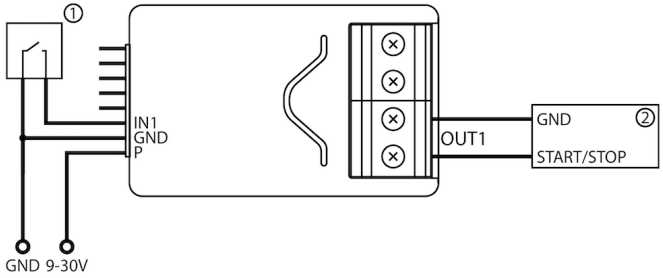
In the hardware chapter we covered the Fibaro Smart Implant. This device is positioned as a retrofit for (not yet) connected products. We are going to use the Smart Implant to actuate the garage door and to monitor its state.

The Smart Implant can communicate with our Smart HUB using Zwave. Therefore the setup and configuration is similar to the other zwave devices we already covered in previous recipe.

The Building Plan

Retrofitting the garage door

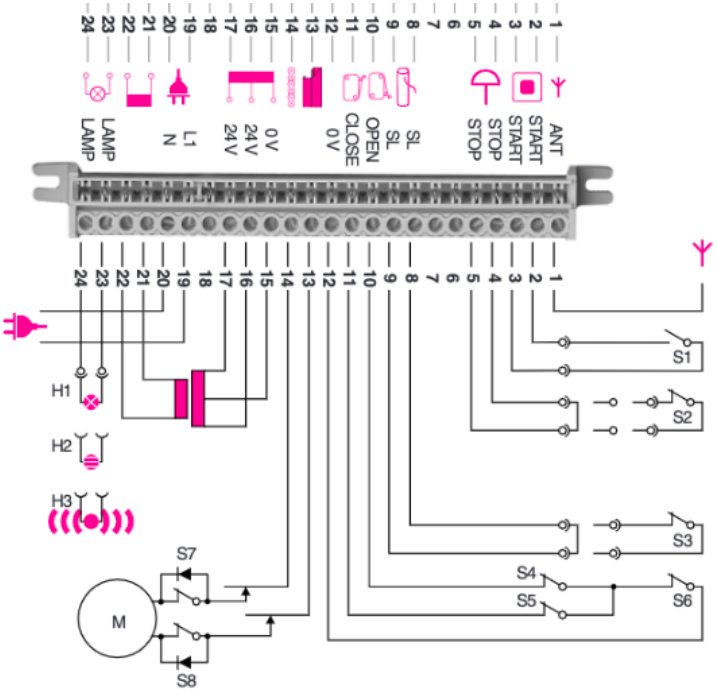
To control the garage door, a Fibaro Smart Implant z-wave device is used as a retrofit device on the garage door unit. This device is ideal for retrofitting existing devices in the home and connecting it to our systems-of-systems architecture.
The Fibaro Smart Implant hosts 2 outputs which we can use as a ‘dry contact’. We use one of them to interface with the hardware of the garage door. a wiring diagram is given below.



The picture below comes from the manual of my garage door. It shows the schematics of the electric motor and how everything is connected.

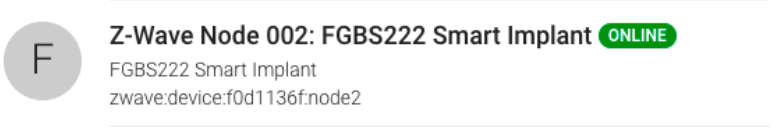
The picture tells us that the indoor button to open and close the garage door is wired to terminal 2 and 3. The description says it’s a potential free NO contact. This is what we need to connect the output of the Smart Implant. Further we need to provide power to the Smart Implant. Something between 9 and 30V DC. On the schematics we can see that we can shunt this from terminals 15 and 16.

To monitor the state of the garage door, we can retrofit a magnetic switch on a moving part such a s the door or the motor on the slider to detect whether the door is open or closed. The magnetic switch can be wired on the input (between IN1 and GND) of the Smart Implant, so we can also detect when the door opens or closes.

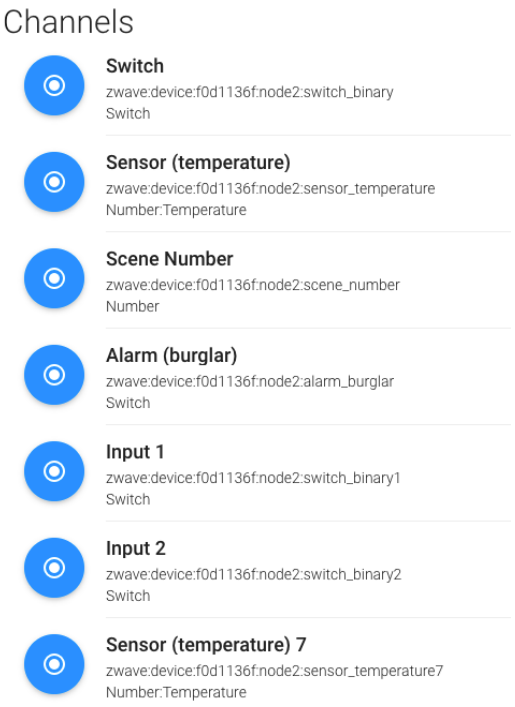


Including the Fibaro Smart Implant

Next we need to Include the Fibaro Smart Implant with the Z-wave controller. (see recipe connecting Z-wave devices). When the device is discovered and added, the Smart Implant shows up as a 'Thing' in the smart HUB.

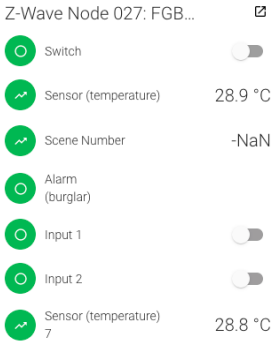


The Smart Implant 'Thing' includes the following channels.



The Switch channel is used to actuate the garage door. The Input 1 provides us with the garage door state.

You can control the garage door from within openHAB paperUI. Go to Control in the left menu and you will find an overview of all your 'Things' for which you have channels. For the Z-wave Smart Implant described above the UI looks as follows.



This interface is nice for testing out your devices and a first impression. In this Cookbook you will find other ways to build user interfaces or use for example a virtual key lock to control your garage door!

RECIPE

WiFi connected switch & energy monitoring

Time to cook 120min
Difficulty ++++-+

Ingredients

- Sonoff POW R2

Prerequisites

- Cook the recipe: Building a smart HUB (Basic recipe)
- Cook the recipe 'Installing MQTT' (Basic recipe)

Tools

- Arduino IDE
- NodeMCU PYflasher
- FDDI to USB module

Intro

The Sonoff Pow R2 is a low cost (about 10€) connected WiFi switch able to remotely manage and control your appliances and allow you to monitor your home energy usage. The connected WiFi switch reports on power consumption, voltage and real-time current.
The Sonoff controller is based on the popular ESP8266 WiFI module and made by the chinese manufacturer itead* that provides users with smart home control. The POW version is a wireless connected power switch with electricity usage monitor feature.

The Sonoff comes with its own firmware that can be controlled by Iteads App EWeLink.

Theo Arends (from the Netherlands) has created an alternative firmware on top of the popular Sonoff. The Alternative firmware for ESP8266 based devices provides a web UI, rules and timers, OTA updates, custom device templates and sensor support. The firmware allows control over MQTT, HTTP, Serial and

KNX for integrations with various platforms. The firmware is written for the Arduino IDE which makes it easy to use.

This alternative Sonoff-Tasmota firmware is ideal for our system-of-systems approach where we can use data communication protocols such as MQTT.

The Building Plan

Updating the firmware

As mentioned above, when you order the Sonoff Pow, it has the itead firmware installed. The first thing we will do is to replace it with the Sonoff-Tasmota firmware from Theo Arends [1], which is a better fit into our system-of-systems approach.

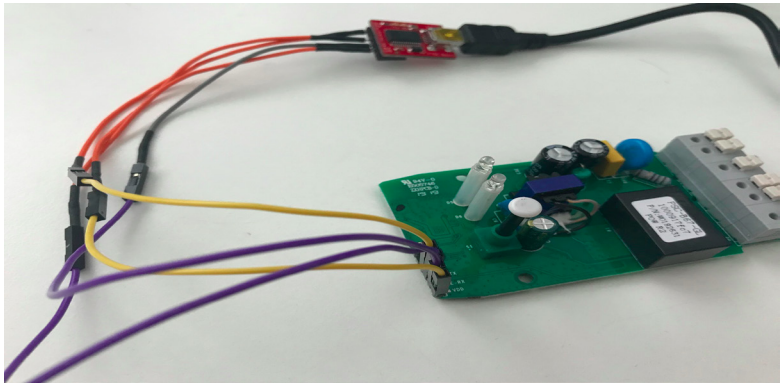
To be able to upload new firmware on to the Sonoff POW, we need to Solder a header pin on the PCB and connect a serial-to-USB converter TX and RX pins to the ESP8266 RX and TX pins and powering the chip with the 3.3V and GND pins

⚠ Make sure the Sonoff-POW is NOT connected to the MAINS power!

More info about the hardware preparation can be found here [2]

1 https://github.com/arendst/Sonoff-Tasmota/wiki/Flashing

2 https://github.com/arendst/Sonoff-Tasmota/wiki/Hardware-Preparation



Serial adapter	Sonoff Pow
3V3	VCC
TX	RX
RX	TX
GND	GND

Now we can flash the new firmware on the device. Either you can compile the Tasmota software using an Arduino IDE and upload it onto the board. Another and easier way is to download a ready-to-use binary file and upload it using the NodeMCU PyFlasher.

- Download a ready to use bin file from: https://github.com/arendst/Tasmota/releases/
- (tasmota.bin is the most appropriate in many cases)
- Download NodeMCU PyFlasher (https://github.com/marcelstoer/nodemcu-pyflasher/releases)
- Press the S1 button on the Sonoff device while powering on the device, the RED led should be permanent ON.
- Select the serial port
- Select the bin file (example: sonoff.bin is the most appropriate in many cases)
- set flash mode to DOUT
- If it is a virgin device, select erase Flash : No
- Click on Flash NodeMCU


Security & Privacy

When discussing the connected house, one needs to consider security and privacy. Each connected object is a source of concern for safety. Every data stream is a concern for our privacy. It goes without saying that one product has a greater impact on safety and privacy than another.

A soil sensor in the garden is less of a security and privacy risk than a connected door lock. If someone can hack your connected door lock, he or she can gain access to your house as if they had the physical key. It's not just about the 'classic' physical security like accessing a building. It's also about gaining access to the data of various objects in a house that can reveal the whereabouts of the occupants of that house. Last but not least, connected devices can be misused for malicious purposes when they are hacked.
.
In 2016, on the morning of Friday 21 October, internet users from the East Coast of the US noticed that web pages such as Twitter, Etsy, Spotify, Netflix and GitHub were unresponsive. The source of the problem was that Dyn, one of the largest internet management companies in the United States, was flooded with junk traffic. Part of the attack was caused by a "botnet" of Internet of Things devices that caused a DDoS (distributed denial of service) attack on Dyn.
Hackers used a DDoS attack to bring together swarms of Internet-connected devices such as routers, security cameras, and even connected refrigerators in a botnet.

Many security problems arise because the connected products are 'new'. Manufacturers of household appliances have good domain knowledge about their traditional product, but therefore not necessarily about how to turn it into a connected product. Developing a connected product is a complex process and requires a lot of expertise in different domains. The required expertise is often underestimated, which means they have to solve the problems later on. Ultimately it will result in a reliable and safe product, but it will take some time to get there.

The Future connected house

Now that we are almost at the end of this cookbook, it should be clear that today's technology enables us to connect all kinds of objects in our homes. But we are still at the beginning of what is really possible. You could compare it to a child growing up. In this situation we are a 2 year old child who is starting to use its senses, while the brain still needs to develop fully.

This cookbook will help you integrate or build your own connected devices - transforming a passive home into an active living space. The result is still very basic and in most cases based on conditional logic (IF... THEN...). More intelligent devices already exist - such as the digital assistant - but they still need to be correctly integrated and programmed in order to be efficient.

Fusing sensor data with artificial intelligence and machine learning makes the connected home smarter. In this chapter we offer a glimpse into the future and see how the connected home could evolve in the coming years.

More sensors results in more insights

The digital energy meter

The traditional house will become more and more connected. Today, residents have to send their energy meter data once a year to a utility company. This process not only provides minimal information on the consumption patterns of electricity, gas and water, but is also inefficient.

Last year, the installation of smart meters started in Flanders. Fluvius, a utility company, started installing the smart meters in new and renovated houses.

Smart meters will make automatic meter registration possible. The utility company will have access to real-time data, giving it better insight into energy consumption and more accurate forecasts. Better forecasting is needed to cope with the shift towards a more decentralised energy network. This means that energy is not only produced by a handful of energy companies, but that each home acts as a potential energy producer.

The digital water meter

Just like the digital energy meters, utility companies want to gain a better insight into water consumption and detect leaks. Studies have shown that more than 25% of water intended for consumption is lost due to leaks. With the coming climate change, this is no longer justified. In more and more regions there is a shortage of water due to prolonged droughts. Over the past year, measures had to be taken by the authorities to regulate water consumption. Farmers suffered heavy losses because the crops could not grow due to the prolonged drought and it was forbidden to irrigate due to the low water level in the canals and rivers.

By continuously monitoring water consumption in homes, utility companies can detect leaks much more easily and intervene, so that valuable water is no longer wasted. In Flanders, the utility companies have started rolling out the first batch of 300,000 smart water meters. The smart water meters use the Sigfox LPWAN technology to communicate with the outside world. Every day, one payload of data is uploaded from the smart meter to the utility company to report the water consumption.

The choice of Sigfox was based on network availability, range and low power consumption. Most meters are located in hard-to-reach places for wireless communication, such as basements. SigFox uses the free sub GHz band (868Mhz in Europe) which allows deep penetration. It is also efficient in power consumption, as every water meter must be able to communicate every day on a single battery charge for 16 years [1]. Communicating daily for 16 years is not a given , It requires a unique design and batteries with almost no self-discharge specifications.

Well Monitoring

Gradually we are surrounded by a plentitude of sensors. Flanders is thinking of making the wells 'connected'. Every new home in Flanders will have to install a well to collect the rainwater used to flush toilets, washing machines, water the garden, etc.

By monitoring the water level in the well and emptying it when heavy rainfall is expected, the wells can serve as a buffer that prevents rainwater from flowing directly into rivers, thus avoiding flooding in low-lying areas. Flanders has 1.5 million residential wells, suitable for approx. 15 billion litres of water storage. The above examples show that the connected house is a fact. Similar plans are being implemented not only in Flanders, but in every country.

The Number of connected devices per person

The number of Internet-connected devices that people have is going up. On average, there will be four networked devices and connections per person globally by 2021, according to the latest annual visual networking index forecast by Cisco [1].

However, in North America, there will be 13 networked devices and connections per person, up from eight last year. This means that beyond smartphones and connected TVs, North American consumers will be adopting many more connected devices.
North America is well above the average by region when it comes to getting connected. Below are the projected number of networked devices and connection per person by region by 2021:

1 https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html