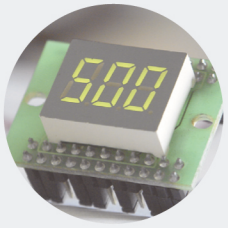
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES






Integração de Sensores em Controlador de Domótica Residencial

VLADIMIR MATIEVSKIY
Novembro de 2019

POLITÉCNICO
DO PORTO

Integration of Sensors into smart home controller

Vladimir Matievskiy

Department of Electrical Engineering

Master`s Degree in Electrical and Computer Engineering

Area of Specialization in Telecommunications

Report prepared for partial satisfaction of the requirements of the Thesis of the Master`s Degree In Electrical and Computer Engineering

Candidate: Vladimir Matievskiy, Nº 1170092, 1170092@isep.ipp.pt

Scientific Orientation: Professor Nuno Filipe da Fonseca Bastos Gomes, nbg@isep.ipp.pt

Company: Central Casa

Supervision: Carlos Silva, carlos.silva@centralcasa.pt

Department of Electrical Engineering

Master`s Degree in Electrical and Computer Engineering

Area of specialization in Telecommunications

2019

Summary

In this paper, you will find information on how to integrate low cost sensors into a smart home controller using the z-wave protocol. The implementation of this project will be done using raspberry and arduino boards, and comprehend the integration of CO2, humidity, temperature, movement and door sensors. Experiments will be carried out with various controllers, such as vera lite, fibaro home center lite, zipabox and z-wave.me. It also provides information on home automation, the various ways to do it, and the disadvantages, advantages of various protocols.

The main part of the work was done with the help of a Russian smart platform, named z-uno. The z-uno is pretty complet smart controller based on an arduino board which allows the communication with z-wave devices.

Index

List of Figures

Acronyms

IoT	–	Internet of Things
IFTTT	–	If This Then That
IPB	–	Universal Powerline Bus
NFC	–	Near Field Communication
SOF	–	Start Of Frame
EOF	–	End Of Frame
GPIO	–	General Purpose Input Output
RAM	–	Random Access Memory
CRC	–	Cyclic Redundancy Check
CRM	–	Customer Relationship Management
CSMA/CD	–	Carrier Sense Multiple Access/Collision Detection
ER	–	Elemento de Rede
eTOM	–	Enhanced Telecom Operations Model
FAB	–	Fulfillment, Assurance & Billing
FCAPS	–	Fault, Configuration, Accounting, Performance, Security
FCS	–	Frame Check Sequence
FIFO	–	First In First Out

1. Introduction

Smart home is a set of solutions for automating everyday activities that will save you from the routine. It includes household appliances, from vacuum cleaning robots to devices controlled by a smartphone and systems controlling everything that happens in the apartment.

In fact, home automation is improving the quality of life. The comfort consist of trifles, and home automation will take upon itself all the trifles. If you woke up at night and went to the kitchen for a glass of water, you wouldn`t have to crawl along a dark corridor in search of a switch: the light turns on automatically. Have you ever been worried that you didn`t turn off your iron, or TV? Down with disturbing thoughts: just send a command to smart socket from your smartphone, and it will turn off the device that is powered by it.

Why do you need it? Everything is obvious: to make your life easier and better. A smart home is peace and great savings. Let`s start with peace of mind. If worrying about everything is common for you, a smart home will help to dispose of at least those worries that are associated with your apartment. Are you afraid that the washing machine launched before leaving the house went out of order and flooded the neighbors from below? Not a big deal. If it really leaks, the leak sensor will instantly tell you about it. What is the result: you are less anxious about non-existent problems and free your brain from unnecessary thinking. You can check the situation at home at any time, using a smartphone. Now about saving. To many people, this advantage will seem doubtful. Say, about what savings are we talking about, when you need to buy some sensors, outlets and a video camera? But let`s take a closer look at smart socket for example – it can track how much energy is consumed by the device, connected to it. As a result, you can calculate the most money-loosing devices and decently save on paying bills.

In this work, I will consider various controllers for home automation, as well as ways to integrate low-cost sensors into these systems.

1.1 Contextualization

This project arose from the desire to learn more about home automation. After have studied introductory course about the knx protocol, I was very interested in the theme of the smart

home, since often projects required creativity for solving a problem, therefore it was interesting to work on them. This automation area is still developing, in my country. The demand for services on this area is just beginning to appear. I decided to do an internship at CentralCasa, which installs, configures and sells home automation equipment, to gain experience with various protocols and home automation devices. The choice of the topic of work is determined by the inability of the market for the main equipment manufactures to satisfy all the desires of costumers. The choice of devices is very limited, and the implementation of your idea may need a lot of money, or it will be impossible to implement it at all. The solution to this problem is to integrate low-cost sensors into your home automation system.

1.2 Objectives

The main objective of this work is the integration of low-cost sensors (ex. temperature, humidity, CO2, etc.) into commercial home automation controllers such as Vera Lite, Fibaro Home Center Lite, Zipato or Z-Way. The integration of sensors should be made using Raspberry, or Arduino boards, using z-wave. Also we should check the compatibility of the above controllers with sensors, about which there is no compatibility information in the technical characteristics of the controllers.

1.3 Calendarization

For the development of this project, I passed an internship in the Central Casa, who provided me with all the equipment for this task. You can see the plan of the work done in the table 1, the only thing that I didn't specify there is working with clients and fulfilling their orders, such as finding a solution for their question.

1.4 Organization of the report

This thesis consists of 4 main parts. The first part contains basic information about this document, the main goals are indicated, as well as in paragraph 4 of the first chapter there is a table with information about the calendar plan of the work performed. The second chapter contains theoretical information about home automation, a brief history of the development and capabilities of smart homes. The practical part of the thesis is based on two projects, each of which offers a way to solve the problem of integration of low-cost sensors to home automation systems. In project Angela, the integration of low-cost sensors was performed using a raspberry board, and in project Bill using an arduino-based system. The final chapter contains the results of the performed work.

Table 1 Calendarization of the project

#	Stapes name	Start	End	Duration	Feb 2019		Mar 2019		Apr 2019		May 2019		June 2019		July 2019	
					25-2		11-3	25-3	8-4	22-4	6-5	20-5	3-6	17-6	1-7	8-7
1	Familiarization with the knx and z-wave protocols	25.2.19	8.3.19	2w												
2	Testing of Fibaro, Zipato and Vera controllers	11.3.19	22.3.19	2w												
3	Familiarization with energy measurement, plotting and storage of information	25.3.19	5.4.19	2w												
4	Study documentation, work with raspberry board	8.4.19	19.4.19	2w												

5	Testing of different software for the raspberry board, integration of radio communication, learning arduino documentation	22.4.19	3.5.19	2w											
6	Development of the thesis	6.5.19	31.5.19	4w											
7	Working with z-uno	3.6.19	28.6.19	4w											
8	Thesis design	1.7.19	12.7.19	2w											

2. Smart Home

The term «smart home» is used to describe a house that contains a communication network that connects different appliances and allows them to be remotely controlled, monitored and accessed, according to the Department of Trade and Industry.

Smart devices connect to the internet and many have smartphone apps allowing you to access and control them remotely over wi-fi.

It`s becoming easier to connect an entire home too. Broadband is faster, more reliable and more affordable than ever before. The improved signal range of wi-fi routers means that a single router can offer wireless coverage across more rooms in our homes, allowing more devices to be connected.

What`s more, low-priced networking equipment has made it cheaper to extend home networks into rooms that were difficult to cover using just a single wi-fi router. Even previously difficult properties, such as older homes with trick walls, can now benefit from a home network that covers the entire property.

2.1 History

The history of the smart home began in the last century. Americans were the pioneers in this area. Actually, the concept of smart house appeared in the Washington state at the Institute of Intellectual Buildings. Revolutionary projects of that time were being developed there, suggesting the possibility of transmitting various types of information over a single wire, which would allow control various devices.

The official date of birth of the smart home system is 1978. Then the Leviton and X10-USA companies developed and launched into mass production the original cable technology X10, which allowed them to manage household appliances through the wires of the electrical

network. It was a real breakthrough in the territory of a single state, since the technology was designed to work exclusively in American networks.

X10 at that time was a simple standard, through which you could execute six commands. The technology was used mostly for lighting control. Over time, this was not enough. The next significant stage in the history of the development of the Smart Home is in 1992.

Shortly before, the EIA (Electronic Industry Alliance) was created. Together, the developers of the Alliance have created a new data line for consumer electronics, which was called CEBus. The peculiarity of the new technology lies in the fact that it is an open standard.

By the mid-nineties European protocol for the communication of elements of the Smart Home has appeared - EIB, created by the EIBA association. It includes more than 15 well-known brands. Admittedly, both technologies, American and European, turned out to be long-lived and are still used today, but with some modifications.

The products of the association are sold under different brands. Tebis, Instabus and ABB i-Bus are the most popular. After several years, more than a hundred global and European manufacturers have manufactured certified EIBA products. This association has become a leader in the development and manufacture of devices for smart homes in Europe and controlled about 80% of the market.

Not surprisingly, by the 2000, more than ten million EIBA instruments had been installed worldwide. In the spring of 1999 another association of smart home appliances manufacturers appeared in Europe. Later it was called the KNX Association. It includes three major unions from Europe, including EIBA. It essentially became the leading organization in the newly created association.

After the merger, the expected merger of their technologies takes place. EIB, Batibus and EHS standards are united. The technology is developing rapidly and 2003 was marked by the approval of the EN50090 as European protocol. Three years later, it was approved under the name ISO / IEC 14543, already as an international standard.

The beginning of the XXI century was a time of rapid development of home automation. Its capabilities have expanded significantly and continue to expand. Technologies designed to perform a very limited set of functions are being transformed into multifunctional and large-scale

ones. Today, not only an apartment or a house, but also a large hotel, high-rise building, airport or stadium, can possess intelligence tasks. Such projects exist and work successfully.

[1]

2.2 Home Automation vs Smart Home

These terms often replace each other, but there is a difference between them. Before we proceed to the further study of this topic, let us see what the difference between these two terms is.

Many real estate owners consider any automatic or semi-automatic device that performs the function of turning on / off any device as an element of the “smart home”. However it is far from truth. Even the ability to remotely control individual functions using the Internet does not make the house smart.

A truly “smart” home is a complex intelligent automation of management of the whole complex of life support systems based on artificial intelligence of a computerized control system and operating in a completely autonomous mode. Human intervention is required only in emergencies or in the programming process.

Therefore, numerous home automation installer firms do not always objectively and reliably convey to the potential customer user a sense of innovation.

It is not always explained that the overwhelming majority of household appliances included in the “smart home” do not need automation, since they already have built-in functions:

- Air conditioners do not require outside intervention to maintain a given temperature
- Washing machines have a delayed start timer
- Lighting on / off systems are easily controlled by relays with photodiodes responsive to light levels

The list can go on for a very long time, but the point is that home automation is most likely already done in your home.

The next important aspect that distinguishes home automation from smart home systems is the connection of all devices to the Internet. That is, the automation and the creation of certain conditions, such as turning on the light, if there is movement in the room, possibly even without the Internet. In a smart home should be the ability to remotely monitor and control all parameters of your home, as well as the ability to change scenes at a distance.[2]

2.3 Appliances and Services of Smart Home

Components can be very different, since it is possible to select certain subsystems, taking into account their relevance to a specific user. Let`s take a look at the main ones that are the most popular and popular.

- Lighting control

  It saves energy and provides a high level of comfort. The user can select the desired lighting mode, create new scenarios that meet the needs and mood. Lighting control takes place using a remote control or a single multi-function panel. There are also options for managing voice commands.

- Climate control

  Subsystem sensors constantly monitor the indoor climate according to a number of parameters (humidity, air temperature, etc.). If necessary, turn on the heating, air conditioning or humidifier.

- Heating system control

  It is possible to connect heating and underfloor heating to the control system, turn it on and off only when necessary, set the on / off time and even warm up the rug on the threshold of the house or in the hallway. Also, the climate control system allows you to turn on the heating by a certain time, thus saving energy resources and not heating the house when there are no residents there.

- Security Management

The system for a given scenario includes video surveillance, alarms, can simulate the presence of residents (including occasional light in the evening). It can also be used to control the leakage of water, gas, etc.

- Management of multimedia devices

One of the most pleasant components of a smart home, allowing you to play content from any device in the house, control sound and image, providing high quality sound and video.

- Purity control

The house can be equipped with an intelligent dust removal system. This allows you to significantly reduce cleaning time, improve its quality, eliminate the need to carry a vacuum cleaner around the house and save everyone from unnecessary noise.

- Shopping Management

This subsystem can save you the need to go shopping. It analyzes the presence of certain products in the refrigerator and automatically sends an order for them to the delivery service. You can also create an algorithm for the purchase of certain things after a certain period of time.

You can monitor the operation of all subsystems of the “Smart Home” remotely using any mobile device or computer. As a rule, the functionality on the computer is wider and allows you to change any parameters of the subsystems. With the help of mobile devices you can control the work and set main tasks.[3]

2.4 Home Automation Protocols

If you decide to design your own automation for a smart home, then first of all you need to decide which protocol will be used by the system components to communicate with each other. There are surprisingly many possible options, so let`s take a look at the most popular European standards.

- 1-Wire

1-Wire - data transfer protocol in both directions on one wire. Perhaps the most commonplace way to tie all the devices together is to adopt the 1-Wire standard. The fact is that the main way for data transmission here is a bi-directional bus, which in the simplest case looks like a two-wire cable. In other words, when creating a network, you can get by with even a cheap telephone cable. One wire in this case is used for power and data transmission, the other - for grounding. The network topology is a common bus, i.e. slaves are literally “strung” on a single cable. Of course, the 1-Wire network does not have to look clumsy: you can take as a basis a high-quality “twisted pair” or FireWire cable, and connect the components via RJ sockets. Actually, the better the network is organized, the longer it can be: in ideal conditions, this value reaches 300 meters. Nevertheless, the main advantages of this standard is cheapness and unpretentiousness. It also has a negative characteristic - low fault tolerance.

- X10

X10 is a standard developed back in 1975. The peak of its popularity is already behind, but it is still actively used in the design of smart homes. The secret of such vitality is extraordinary versatility at a relatively low cost. Unlike 1-Wire, X10 does not require laying a special cable: the building's electrical wiring is used to transmit the signal. It is also possible to use transceivers that catch the radio signal from wireless devices, convert it to the desired format and transmit to the electrical network. This function is used to interact with sensors and remote controls.

A serious disadvantage of X10 is the low data transfer rate, due to which the response to an action occurs with a second delay. But it has a great number of various executive modules (actuators): with their proper selection, automation is able to control electrical appliances, lighting, heating, ventilation and security systems.

- KNX

KNX is an expensive option that is popular in Europe. The standard is characterized by the abundance of its functions, as well as the complexity of design and installation. As a medium for data transmission, the KNX protocol can use a bus (twisted pair), an electrical network or a radio channel. Most often, the first option is used, often the necessary wires are laid together with power cables at the construction stage. The standard provides various options for network topology. The resulting system must have its own power source, but it may not have a central controller, i.e. KNX allows you to create decentralized solutions in which sensors and actuating

modules interact directly. The protocol is suitable for automation of large buildings. Up to 58,000 devices can be combined into one network. In this case, the actuators are diverse and allow giving the house non-trivial functionality.

- Wi-Fi

Wi-Fi is used in almost every smart home. Typically, this communication standard is used to associate a smartphone or tablet with a ready-made automated system. The mobile device is always at hand, and therefore it is more convenient to control the house with its help than to use a computer, a wall touch panel, a remote control or a speech interpreter for this purpose. This is especially true in cases where a special application is provided for interacting with the system, and not just a web interface. However, no one will use wi-fi as the main protocol for a smart home, and the main reason is the high-energy consumption.

- ZigBee

ZigBee is a radio communication protocol that fits perfectly into the concept of a smart home. First, the standard allows you to create sensors with low power consumption and excellent responsiveness: most of the time their wireless modules are in sleep mode, but it takes only 15 milliseconds to wake up the latter. Secondly, ZigBee supports network mesh topology, in which individual components can act as an intermediary, transmitting a signal from one device to another. Such a structure is capable of self-organization and self-restoration, the failure of one or two elements, as a rule, does not lead to serious consequences. The cellular topology also allows you to significantly increase the coverage area of the wireless network, so with the right approach, ZigBee can be used to automate not only residential buildings, but also large working premises.

In general, ZigBee is suitable for solving all typical tasks related to designing a smart home. At the same time, the cost of equipment can be called affordable, and the installation process is relatively simple. However, there is big disadvantage in this standard : ZigBee-devices from different manufacturers are often incompatible.

- Insteon

Insteon is very popular in the United States, but it has only recently come to Europe. The incompatibility of the original version of the protocol with European power grids has become a

factor, that slowed growth of the protocol. Like the X10, Insteon uses building wiring to transmit signals. However, unlike the outdated competitor, the new standard also supports communication over the air, and the wired and wireless networks operate simultaneously, complementing each other and significantly increasing the reliability of the automated system. In addition, Insteon has no problems with the responsiveness, unlike x10.

- Z-Wave

Z-Wave is a wireless communication protocol that is in many ways similar to ZigBee – both have low power consumption and network mesh topology support. These standards are also comparable in equipment cost. Of course, a detailed study of the protocols can reveal some differences in the technical part, but the fact that their developers have a different attitude to the issue of standardization is much more important. Z-Wave has this strictly: all devices, regardless of the manufacturer, are based on Sigma Designs wireless modules (this company developed the protocol). As a result, they are all compatible with each other. Also, less software was released for this standard than for ZigBee, but it is guaranteed to work with any Z-Wave equipment.

Thus, Z-Wave is like “LEGO” from the world of smart homes. Yes, this is not the cheapest and not the most reliable option, but for those who want to automate their home on their own, it will be the best option.[4]

2.5 How to choice the correct protocol

The choice of protocol for creating a smart home, as you can already understand from the description of the advantages and disadvantages of protocols, depends on several factors:

- Functionality
- Price
- Area size
- Country of residence
- availability of existing networks

In any case, the best way out is to consult with a specialist who can evaluate all your wishes and based on his experience to offer the best option.

2.6 Z-Wave structure

Z-Wave is a wireless protocol designed specifically for remote control. Unlike IEEE 802.11 data transmission standards, operating mainly at 2.4 GHz (ZigBee, Wi-Fi and Bluetooth), Z-Wave operates in the frequency range up to 1 GHz (908.42 MHz in the US and 868.42 MHz in Europe and Russia) - the choice of this range is due to the small number of interference sources, in contrast to the loaded range of 2.4 GHz. All main features of z-wave protocol can be found on the table 2.

Table 2 Z-wave features

Specification	Z-wave support
Standard	ITU-T G.9959 (PHY and MAC)
RF Frequency Range	868.42 MHz in Europe, 908.42 MHz in US
Data rate	9.6, 40, 100 Kbps
Maximum Nodes	232
Architecture	Master and slave in mesh mode
MAC layer	CSMA/CA
RF PHY modulation	FSK (for 9.6kbps and 40 kbps), GFSK with BT=0.6 (for 100kbps)
Coding	Manchester (for 9.6kbps), NRZ (for 40 and 100 kbps)
Distance	30 meter in indoors, 100 meters in outdoors

- Z-wave frequency bands

Following table 3 mentions frequency bands, data rate and channel bandwidth supported by z-wave technology throughout the world.

Table 3 Z-wave frequency bands

Region	RF Center Frequency	Data Rate	Channel Width
Australia	f_{ANZ1}/919.80, f_{ANZ2}/921.40	100/40/9.6Kbps	400300/300KHz
Brazil	Same as Australia		
Canada	Same as USA		
Chile	Same as USA		
China	f_{CN1}/868.40	100/40/9.6Kbps	400300/300KHz
European Union	f_{EU1}/869.85, f_{EU2}/868.40	100/40/9.6Kbps	400300/300KHz
Hong Kong	f_{HK1}/919.80	100/40/9.6Kbps	400300/300KHz
India	f_{IN1}/865.20	100/40/9.6Kbps	400300/300KHz
Israel	f_{IL1}/916.00	100/40/9.6Kbps	400300/300KHz
Japan	f_{JP1}/922.50, f_{JP2}/923.90, f_{JP3}/926.30	100/100/100 kbps for all bands	400/400/400 KHz for all bands
Korea	f_{KR1}/920.90, f_{KR2}/921.70, f_{KR3}/923.10	100/100/100 kbps for all bands	400/400/400 KHz for all bands
Malaysia	f_{MY1}/868.10	100/40/9.6Kbps	400/300/300 KHz
Mexico	Same as USA		
New Zealand	Same as Australia		
Russia	f_{RU1}/869.00	100/40/9.6Kbps	400/300/300 KHz
Singapore	Same as EU		
South Africa	Same as EU		

UAE	Same as EU		
USA	f_{US1}/916.00, f_{US2}/908.40,	100/40/9.6Kbps	400/300/300 KHz

- Z-wave network

The Z-Wave protocol is based on a mesh network in which each node or device can receive and transmit control signals to other network devices using adjacent nodes. You can find block diagram of the network on figure 1.

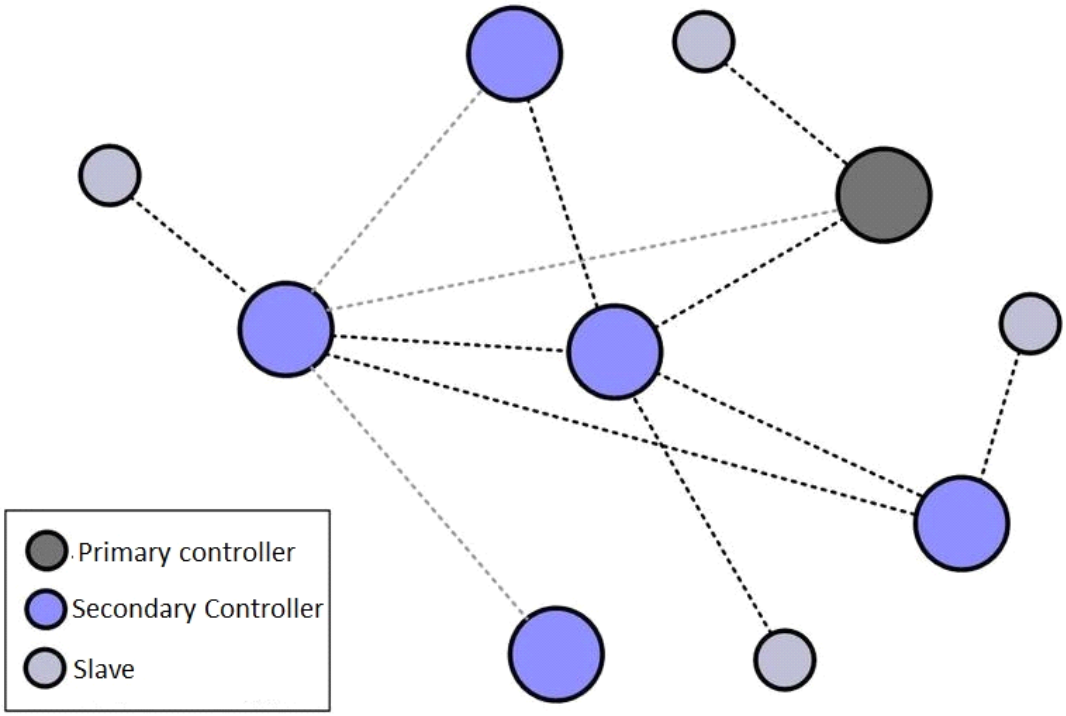


Figure 1 Z-wave Network

The mesh topology in Z-Wave is self-organizing — for example, if a barrier occurs between two closest network nodes, the signal will go through other network nodes that are in range. In total, the network can be up to 232 nodes. Each logical Z-Wave subnet is assigned its own Network ID, which is 4 bytes long. Devices on subnets with different Network IDs cannot communicate with each other.

Unlike ZigBee, the addressing in the Z-Wave network is made by the device address within the network (Node ID, has a length of 1 byte), and not by its unique identifier. The primary (central) controller is responsible for the interaction between devices on the network, and it is also responsible for communication with the outside world. There can be only one central controller,

but the protocol supports the possibility of implementing several secondary controllers in a network.

The central controller also stores a network of routes that transfer data between devices on the network (the maximum size of one data packet is 46 bytes), and when initializing new gadgets it remembers their location. However, since the addressing is not based on the device identifier, moving the device within the network may lead to the appearance of non-working routes, so it may be necessary to update the routes manually.

Also, the Z-Wave protocol has the function of confirming the delivery of data - when the end device receives the data packet, it sends a confirmation command to the central controller. If the command does not arrive within some time, then the data packet is sent again.

Controllers are divided into two groups: static and portable. The central controller belongs to the first group - it is usually a device connected to the power supply network (it is possible to use a PC as a primary controller) and having a certain unchanging address on the network - that is, it can both transmit and receive commands. The portable controller is a remote control - it does not have a permanent address on the network and it is not in the route list, so it can only send a command, but cannot receive it.

Also in the network can be the so-called FLiRS-devices: they are forced to work with the incoming signal (for example, "smart" locks). In this case, to save battery power, such devices check once per second whether there is a data packet for it in the network, and if it exists, the device wakes up and performs its function.

- Z-wave protocol stack

A characteristic feature of Z-Wave is strict standardization from the physical layer to the application level. That is, the Z-Wave protocol stack covers all levels according to OSI classification, which allows ensuring compatibility of devices from different manufacturers when creating networks. The stack is presented on the following figure 2.

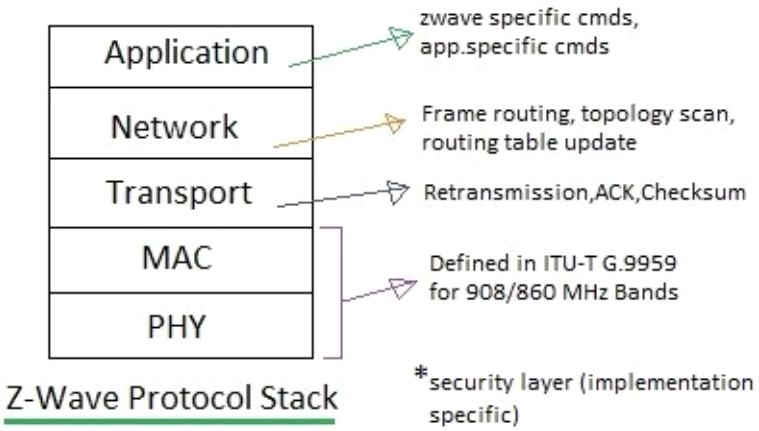


Figure 2 Z-wave protocol stack

At the physical level (Physical Layer - PHY), Z-Wave data is transmitted at frequencies from 865 to 956 MHz, in the ranges provided for the respective country (region). The frequency range used in Z-Wave allows you to achieve very good "long-range" performance compared to competing technologies like WiFi or Bluetooth, which use higher frequencies that have worse penetration. It means that it is Z-Wave that is more convenient in large houses and apartments with thick walls, where the Bluetooth signal can simply not pass through obstacles. The negative side of this is that it is Z-Wave that is most susceptible to attacks by intruders who are near, but not inside a house equipped with Z-Wave equipment. All components of z-wave structure can be found on the following figure 3.

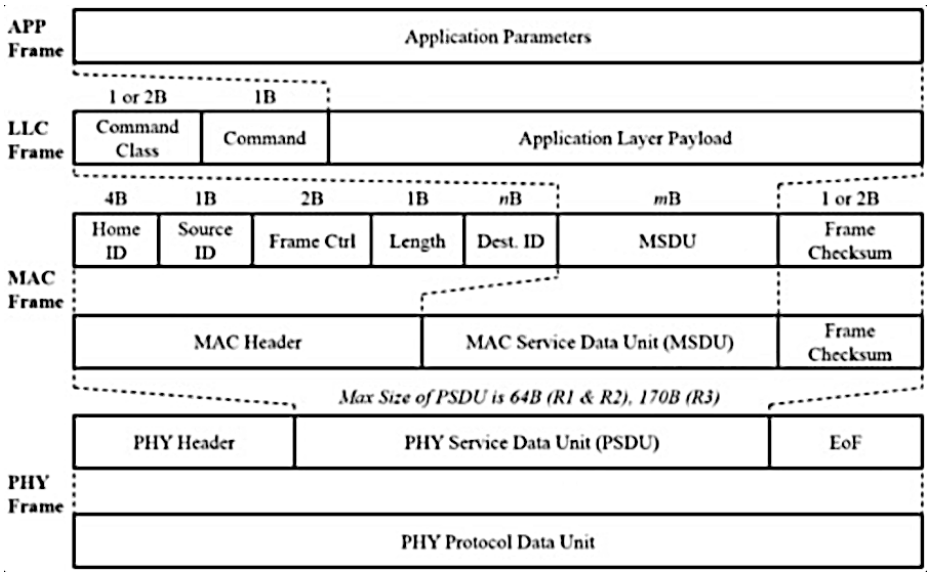


Figure 3 Z-wave frame structure

The functions of the MAC sublayer include channel selection, collision resolution, integrity monitoring, packet delivery confirmation and retransmission. To perform such functions, the transmitted data packets contain special service fields, and the packets themselves can be transmitted in requests of different types (the type of request is also specified by one of the service fields).

To ensure message routing, each node in the network has its own unique (within a specific network) 8-bit Node ID, which is assigned by the primary controller when the device is connected to the network. Also, when turned on, the included device remembers the Home ID of the primary controller for further communication. In total, each Z-Wave network can contain up to 232 devices.

The Logical Link Control (LLC) sub-layer is intended to provide communication between the channel and network layers of the protocol. It provides encapsulation for multi-channel transmission (Multi-Channel encapsulation). This is necessary because at the physical level, three channels are used simultaneously to ensure fast and reliable data transfer.

In a conventional wireless network with a central controller (such asa WiFi network), this controller is required to be able to establish direct communication with each node or device of the network. In Z-Wave networks, this restriction is not presented. Z-Wave is a mesh network, where each node knows its surrounding nodes and can send packets through them. The view of mesh network is presented on figure 4. In addition, each individual node can work as a repeater. That is, if any of the nodes is not accessible to the controller directly, then through the chain of repeater nodes it can lay a route to the desired node (up to 4 repeater nodes can be contained along the route).

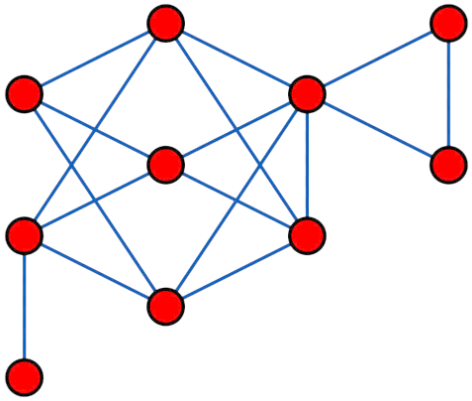


Figure 4 Mesh network view

Due to the fact that, by its nature, the Z-Wave protocol is asynchronous, there is a segmentation and assembly layer (Segmentation and Reassembly - SAR) at the transport layer. On this sublayer, the incoming data (datagrams) are divided into packets. which in size correspond to the allowable packet length on the MAC sublayer (MAC Protocol Data Units - MPDU). Appropriately on the receiving side of the received MPDU packets, the reverse assembly occurs in the data layer datagrams.

Z-Wave also defines the algorithm for interpreting commands received at the application level. This level is described by a set of command classes . For some Classes, there are several options for interpreting commands that depend on the Device Class, which determines the type of device. All commands in Z-Wave are extremely compact. This is necessary to reduce the size of the packet, which has a positive effect on the time taken on the air, as well as a reduction in transmission losses. Z-Wave is designed to transmit short commands without opening a session, that is, it is not at all adapted for streaming data.

- Command classes. All application-level data is transmitted in the form of short packets of the form: “Command Class ID • Command ID • <command-specific data>”. First comes the command class, then the command in this class, then the data specific to this command. Through strict standard describing classes of commands, devices from different manufacturers can understand each other without any problems.

- Device class. Each real device is characterized by its functional type or device class. Each such class defines the required classes of commands supported by the device, and how to interpret these commands.[6]

3. Practical Part

The main goal of this work is to expand the capabilities of your home automation system through the integration of low cost sensors. As mentioned above, the list of devices supported by the z-wave protocol is limited, and if you need extraordinary environmental data, such as the content of CO2 in the room, then you probably will not find official devices, therefore you need to look for ways to solve this problem.

In this paper, will be considered two options, which correspond to two different projects. The first project uses a raspberry microcomputer which has various inputs, and outputs, for integrating various sensors using wires. The second project is based on a z-uno board, which is an arduino-based board with a built-in z-wave module and control software.

On the next section we will present the most important relevant components used on the two projects.

3.1 Projects Components

- Z-uno

Z-Uno is a Z-Wave device prototyping board based on the ZM5101 chip. The Z-Uno board allows you to develop a Z-Wave device, and it will be 100% compatible with any other Z-Wave device. The trick is that the Arduino IDE is used for development, which greatly speeds up the programming process.

In fact, Z-Uno is an Arduino, based on a different chip and with a radio module. Any pin can be configured to output or input, there are four PWM pins and four ADC pins, UART and SPI are present for communication, power is from 3.3 to 18 V. The appearance of the device you can see on the following figure 5.

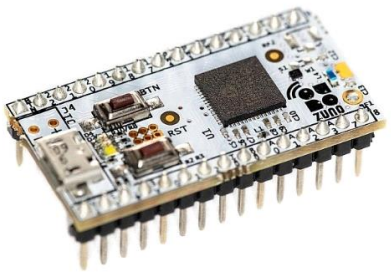


Figure 5 Z-Uno

Z-Uno is a fully DIY product. It is made for those who are limited by existing choice of Z-Wave products and wants to extend their smart homes with more sensors and actuators : connect LEDs, buttons, switches, motors or any low voltage sensor including most of Arduino compatible sensors.

- Raspberry Pi 3 b+

  The Raspberry Pi is a micro-computer initially designed for education. The device is presented on the figure 6. It has all of the components you would see on a normal family desktop PC — a processor, RAM, HDMI port, audio output and USB ports for adding peripherals like a keyboard and mouse.

Figure 6 Raspberry PI 3B

  Alongside these recognizable components is one of the key parts of the Pi — the GPIO (General Purpose Input Output) header.

This is a block of pins that let you connect your Raspberry Pi to the real world, connecting things like switches, LEDs, and sensors (and much more) which you can control with some simple code.

- Razberry

The RaZberry2 expansion card for the Raspberry Pi turns the most popular and cheap mini-computer into a Z-Wave home automation controller. The card is presented on the figure 7.

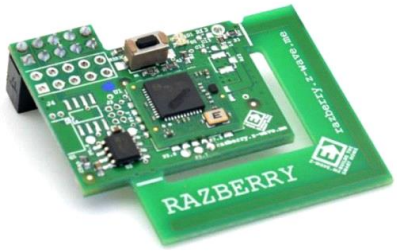


Figure 7 Razberry

On Linux, RaZberry is visible as com-port / dev / ttyAMA0. The board essentially consists of a ZM3102 transceiver, an EEPROM memory for storing Z-Wave network data, a PCBA antenna, and a UART foot connector for a GPIO Raspberry Pi. Only Vcc, Gnd, TX and RX feet are really used. The rest of the pins are not used to work the board and only help to fix it more tightly. In theory, these pins can be used for other needs.

Then, we will consider a method of connecting wired sensors to a raspberry board without the use of additional devices. It will be much more complicated, take more time and require some knowledge of working with the Linux system.

- CO2 sensor

Measuring carbon dioxide is a prime example of the useful expansion of the functionality of your smart home. This value is very important, which should be monitored as well as automatically monitored, but I have not found such devices on the market for devices that have a built-in z-wave connection. However, this device has one disadvantage - the price, since it is a non-trivial sensor with a more complex measuring process than the other sensors used in this work. However, if you want the state of health and efficiency of your employees in the office to always be at a high level, then you will have to spend 30 euros and install such a sensor. I have used MH-Z14A. View of the sensor can be found on the following figure 8.

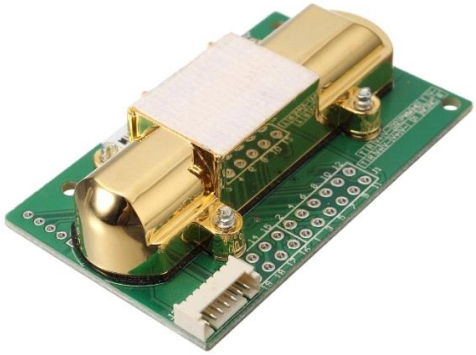


Figure 8 *MH-Z14A carbon dioxide sensor*

- Home controllers

To test the compatibility of equipment with smart home controllers, I chose the most common models from various manufacturers. The most popular controllers are presented on the figure 9. The Vera edge controller from the Vera Company, fibaro home center lite controller from Fibaro Company and zipabox controller from Zipato Company.

Figure 9 Home automation controllers

- Other components

To complete this task, we will need some sensors: 18bs20 temperature sensor, a door-window sensor, a DH-11 humidity sensor, and the most common fibaro motion sensor. To connect all the components you need ordinary materials, such as resistors, wires, LEDs and a micro-sd card for installing a Z-Way system on it.

3.2 Project Angela

Nowadays, there are quite a lot of controllers for your smart home, and there is a possibility to create such a controller by yourself, using raspberry PI board. Let's take a closer look at the process of setting up this controller and connecting low-cost devices to it.

Raspberry will be our low-cost home automation controller to which low-cost sensors can be wired. In the case when the customer already has home automation, and he wants to expand its capabilities with specific sensors, the raspberry board will act as a secondary controller, which will act as a connecting node between the sensor and the main controller.

In this project, we will get a home automation controller with an installed z-way system and a connected temperature sensor and relay, which will allow us to do automatic temperature control, as well as add a wireless motion sensor to check the compatibility of the system with conventional z-wave devices. On he following figure 10 you can find block-diagram of raspberry-based system.

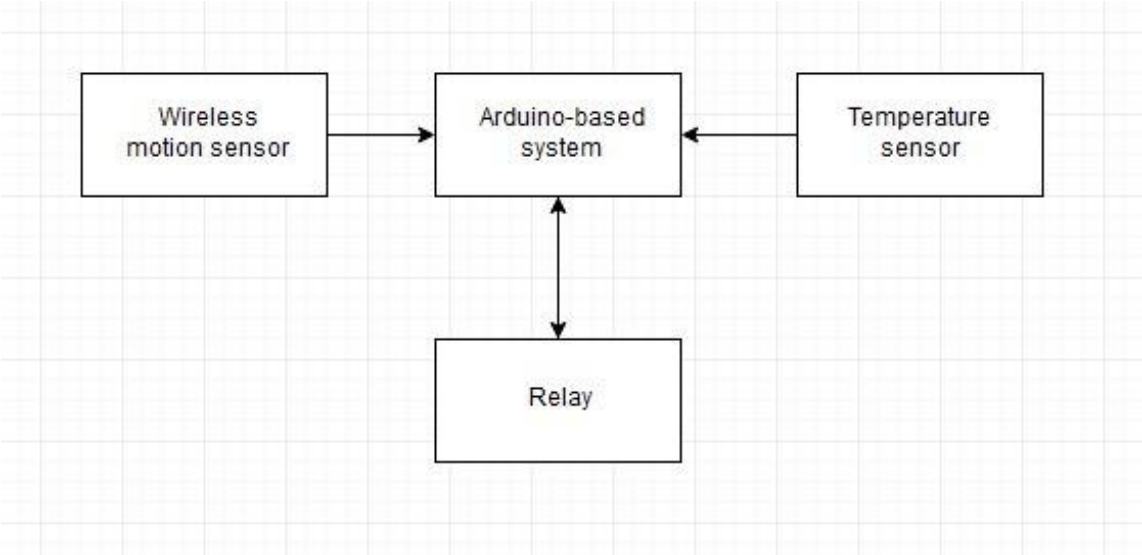


Figure 10 Block-diagram of raspberry-based system

In order to start working with Raspberry it is necessary to install software on it. There are many different softwares, which can be installed on Raspberry. As I'm using the RaZberry card to insure z-wave communication of Raspberry, I'm using the software, called z-way, which was made by the same producer and which it performs better. Each software has its own advantages and disadvantages. For example, radio communication cannot be integrated in to z-way, but it supports a great number of devices and it has high speed of communication.

Detailed installation instructions for installing z-way on the razberry board can be found in the Appendix A.

On the next sections we will show how to install the software on raspberry, enable reception of data from certain pins, connect a relay and a low-cost temperature sensor, and create a simple heating automation scene.

3.2.1 Including of wireless z-wave sensors

Now we will check the performance of our z-way system and try to add any device that works with the z-wave protocol. To do this, go to the menu for adding devices, as shown in the following figure 11.

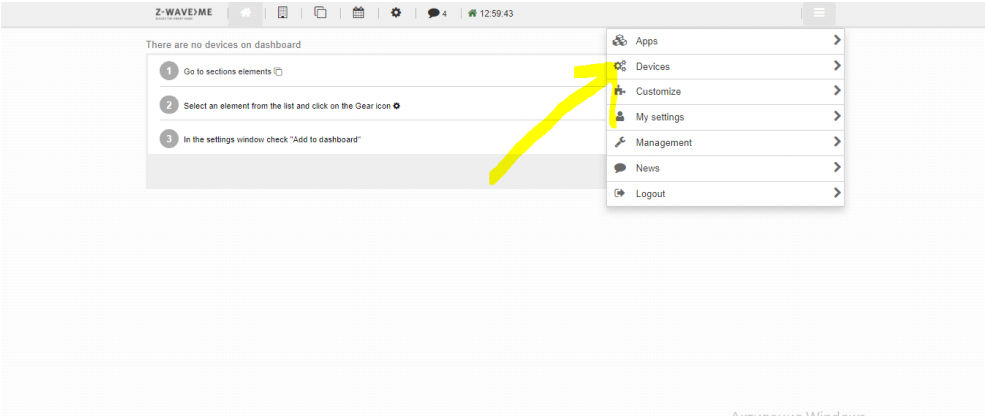


Figure 11 Z-wave.me dashboard interface

As we can see, we have several options, while choosing the device, we want to add. In our case we should choose z-wave device, as it shown in the figure 12.

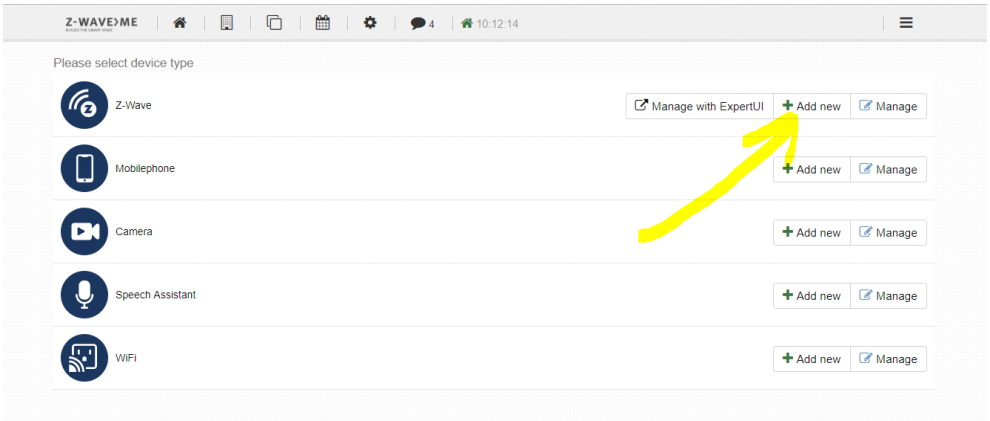


Figure 12 List of supported device types

I had at my disposal the motion sensor of the fibaro company. After adding a device to z-way, you can find it by clicking on the tab «elements». The following figure 13 shows the fibaro motion sensor in the z-way elements tab.

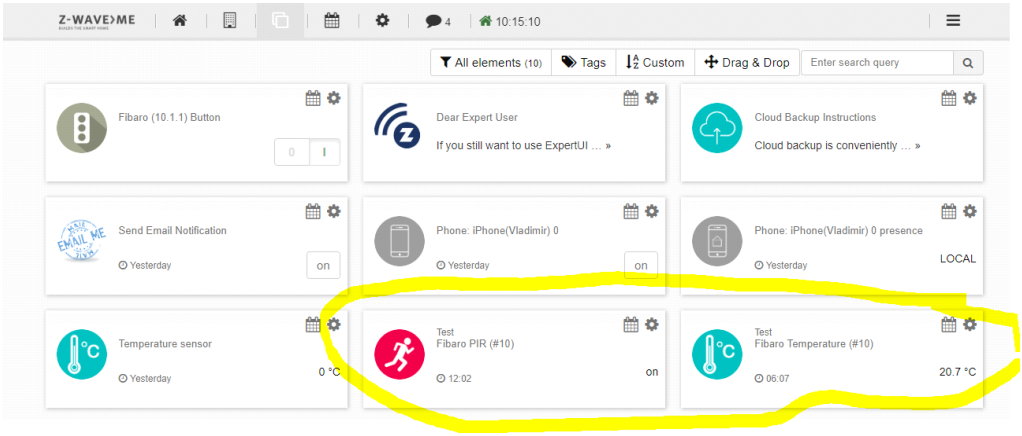


Figure 13 *Fibaro motion sensor on the device panel*

3.2.2 Including of wired DS18B20 temperature sensor

The main advantage of using a raspberry board is the ability to connect any low-cost sensors directly to the board. Quite a lot of these devices are not represented in the catalog of large manufacturers, so the functionality of the raspberry board in this case is not limited. Consider connecting a cheap temperature sensor 18b20, which is shown in the figure 14. First, consider the connection to our raspberry board.

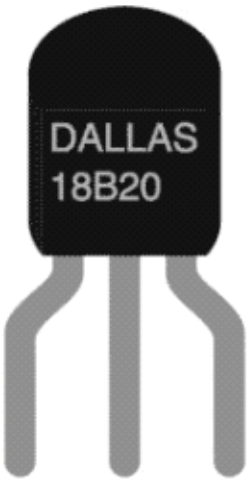


Figure 14 ds18b20 sensor view for pins description

The temperature sensor has three pins. The left one is connected to the ground, the middle one displays information from the sensor, the right one is connected to the power supply 3.3 v. The following figure 15 shows all the pins required to connect the sensor.

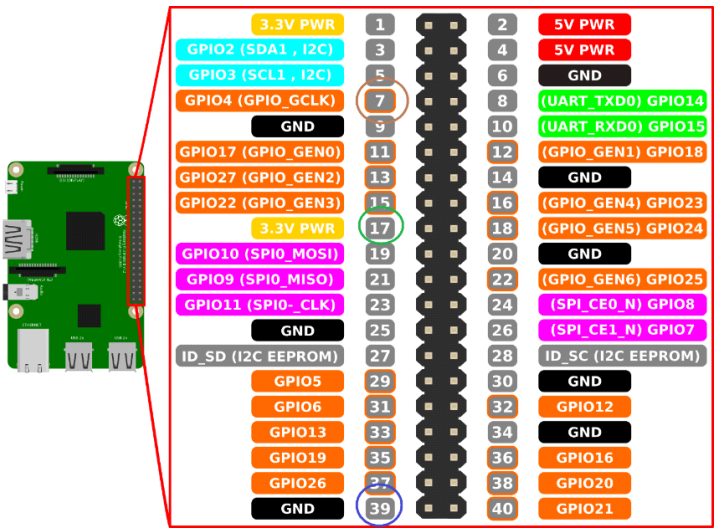


Figure 15 *Raspberry PI pinout*

The DS18B20 temperature sensor is cheap and easy to connect. It works on a 1-Wire bus at a distance of up to 300 meters. You can connect several dozens of sensors to one Raspberry Pi pin, which should be enough for most household needs. The default for 1-

Wire is pin 4, but the RaZberry board blocks it, but there are duplicate pins on it, to which we will connect our sensor.

The next step is to configure the controller system to read information from our sensor. Connect to the controller using the putty program.

Enable the one-wire interface

- Find the config.txt file
- Then add the last line, showed at figure 16, to the bottom of the file

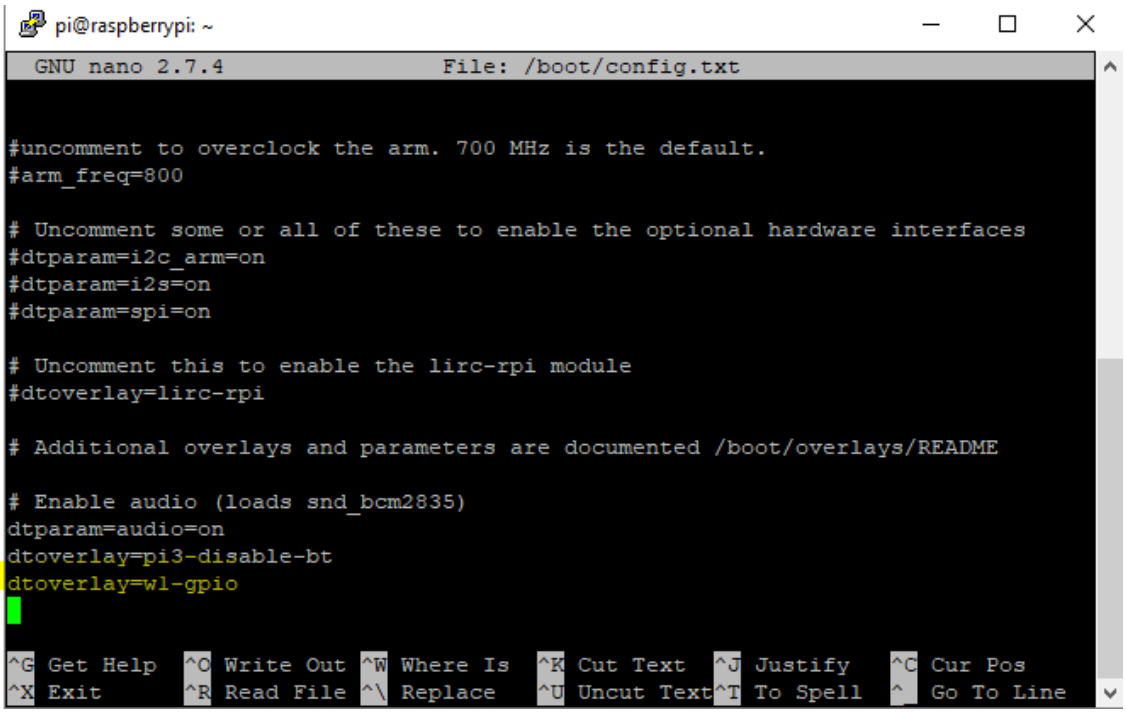


Figure 16 *Activation of 1-Wire in Linux*

- Exit the console and reboot the PI
- Log in to the PI again, and at the command prompt enter the lines, showed in the following figure. By this commands we will activate w-1 communication and enter the folder, where we can see all the devices connected with this type of communication, as it shown in the figure 17.

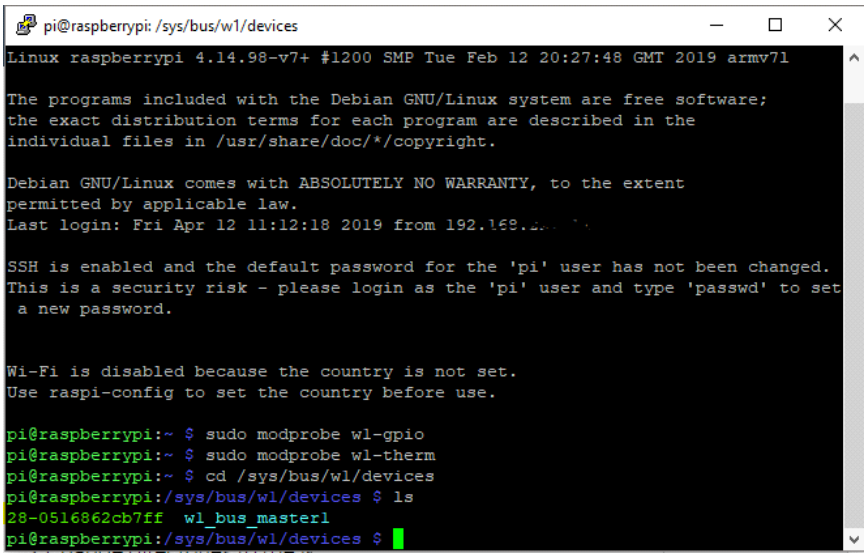


Figure 17 List of connected 1-Wire devices

28-0516862cb7ff and w1_bus_master1 are displayed in my case.

- Now enter we should enter the first folder to see the values from the temperature sensor
- In the following figure 18 we can obtain the data from the sensor

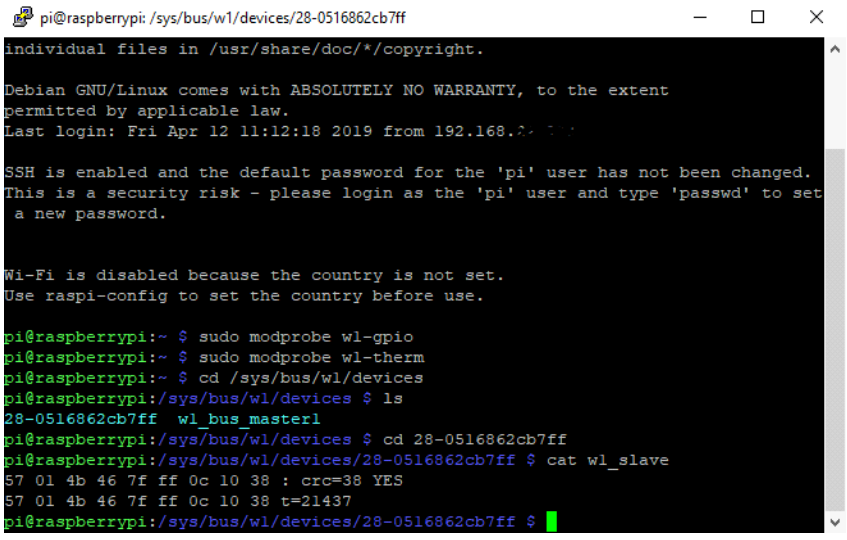


Figure 18 data, received from the sensor

Here is the temperature reading is t=21437, which means a temperature of 21.437 degrees Celsius.

That is all that is required to set up the one wire interface. Now you can run one of the programs below to output the temperature.

There is only one more step. We have to allow z-way server to retrieve the information from our temperature sensor.

For this, follow the directory

```
$ cd /opt/z-way-server/automation/
```

Then add to .syscommands file cat function as I did in the following figure 19.

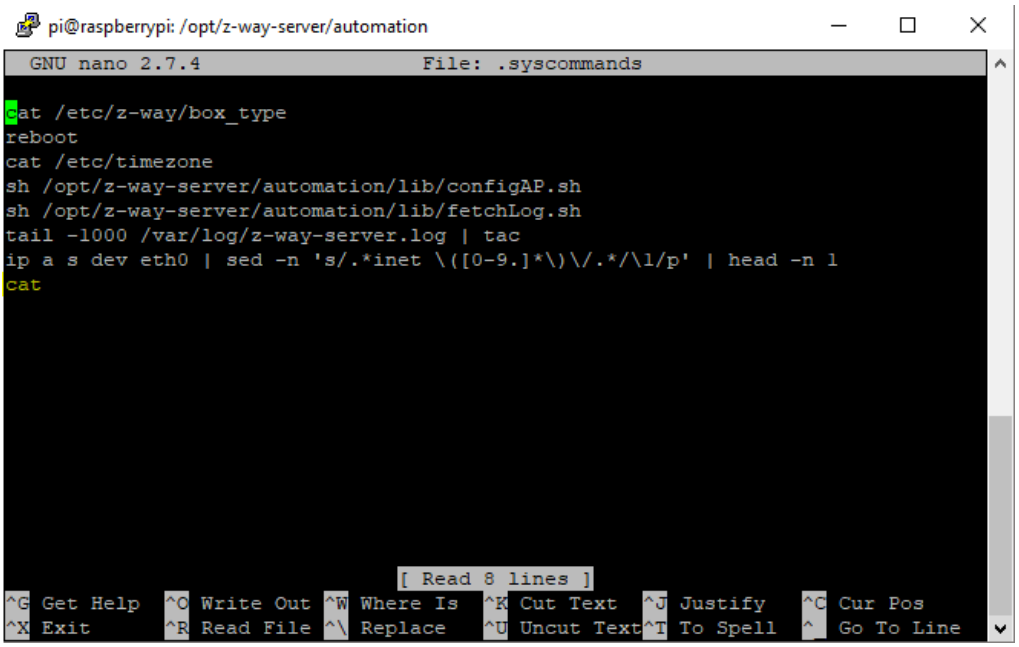


Figure 19 Permission to use data from sensor by programs

Save the file and reboot the controller .

Now we can add a sensor to our devices. Let's go back to z-way site and add a new device.

To access the site, you must use the login «admin» and password that we specified during registration.

We should follow to “Local Apps”, as I did in the following figure 20, where we will find several ways to add a custom device to our system.

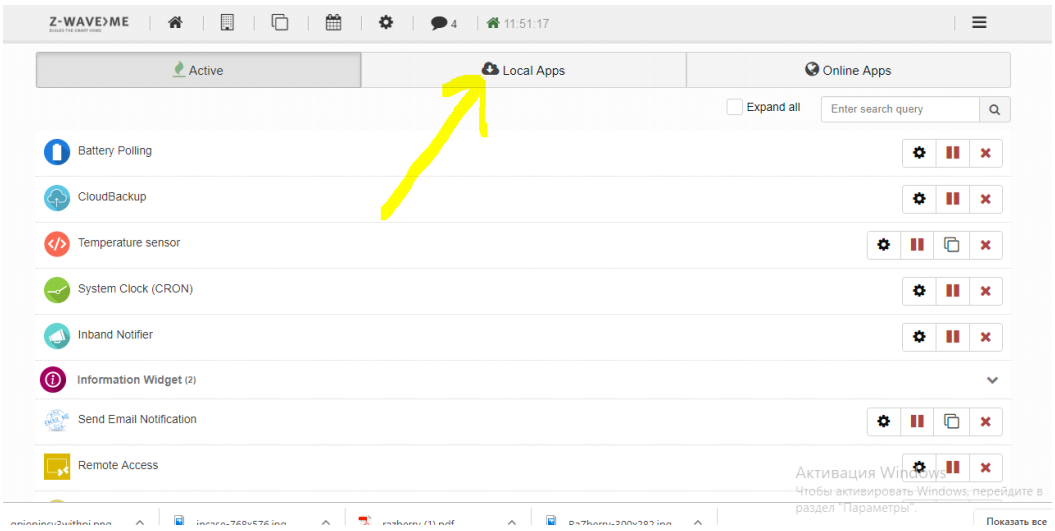
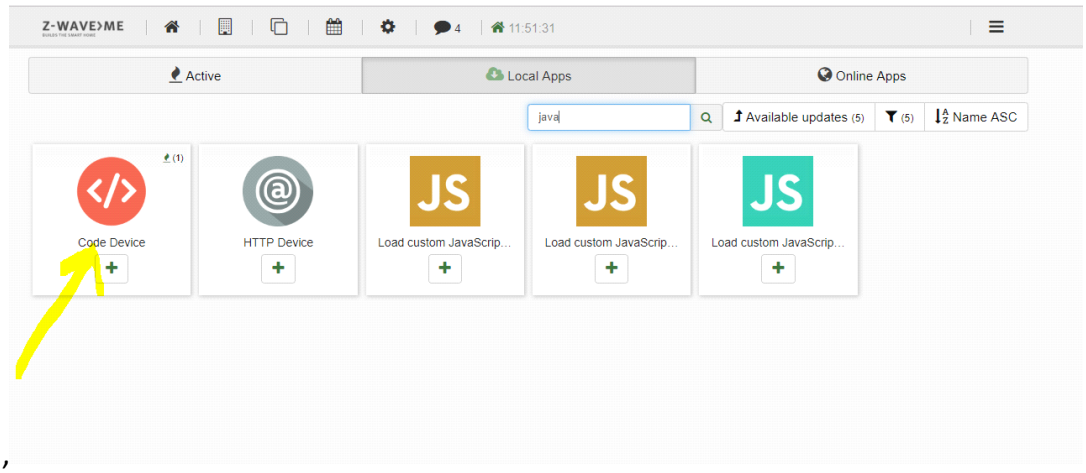


Figure 20 Z-wave.me application interface

In our case, we will use simple java code device, shown in the next figure 21, as the temperature sensor is already connected to the board, and everything we should do is to get the information from the file and transform to normal view our value.



,

Figure 21 Types of custom devices

Code to get the value:

```
system('cat/sys/bus/w1/devices/28-
0516862cb7ff/w1_slave')[1].match(/t=([\-0-9]+)/)[1]/1000
```

Then click to “save”. On the next figure 22 you can see all the parameters of our temperature sensor.

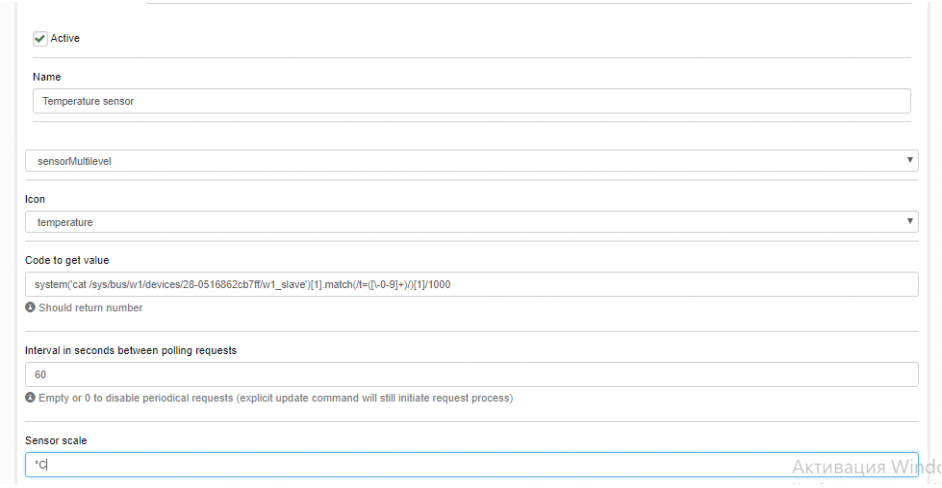


Figure 22 Temperature sensor settings

Now on the figure 23 we can see our device at elements tab.

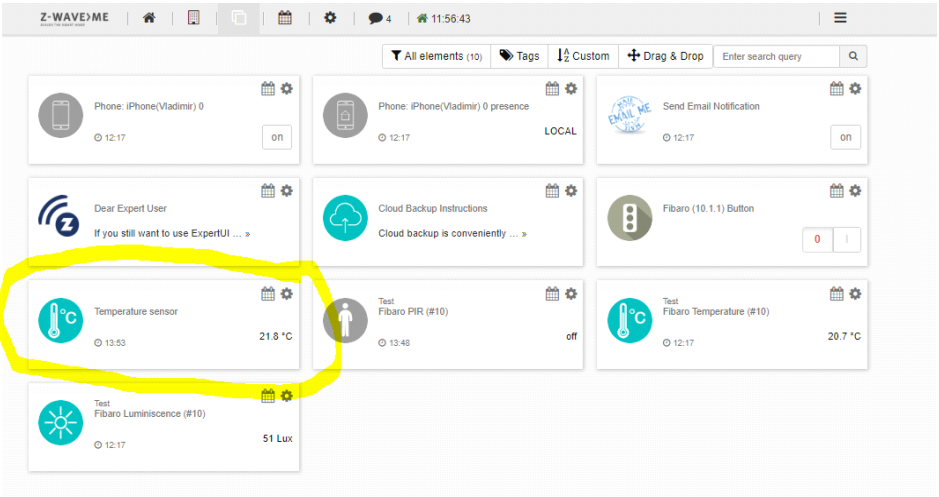


Figure 23 Custom temperature sensor on the elements tab

3.2.3 Including of wired Arduino relay

For Arduino and not only cheap modules with one, two, five or more relays on board are sold. The modules are controlled from 5 V and switch 220 V, they can be used to control lighting and household appliances such as pumps, ventilators, fans. By connecting a relay to the Raspberry Pi, you can configure a scenario where the wireless sensor turns on the light. The connection scheme is simple: the module is connected to the power supply of 5 V and controlled from any free pin. For example, I chose the 25th pin, as it shown on the following figure 24.

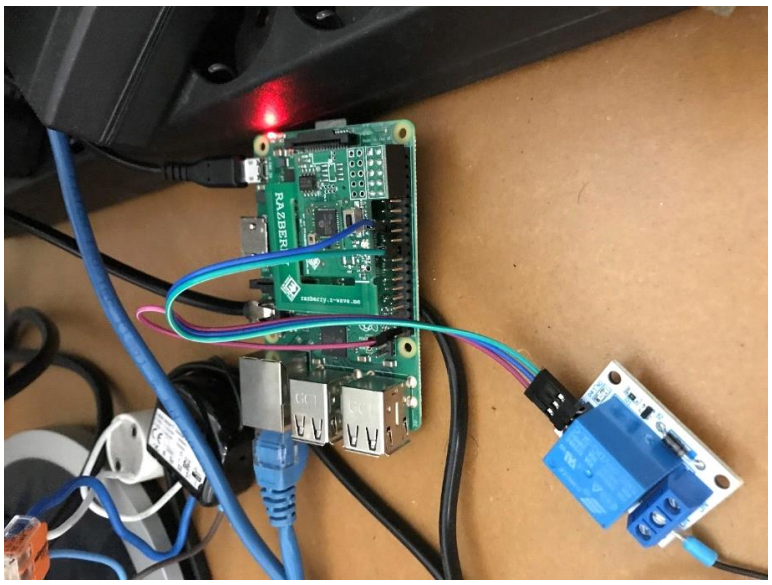


Figure 24 Connecting of wired relay to Raspberry PI

To add a relay to the Z-Way automation system, you need to create a virtual device: “Menu → Applications → Local Applications → JavaScript Device”. The procedure is identical to the addition of a wired temperature sensor, so I will not re-add screenshots for this procedure.

In java commands we should add 2 line for turning on and turning off the device

To turn on

```
$ system("echo '1' > /sys/class/gpio/gpio25/value")
```

To turn off

```
$ system("echo '0' > /sys/class/gpio/gpio25/value"
```

After all we should see our relay on the elements tab, as it shown on the following figure 25.

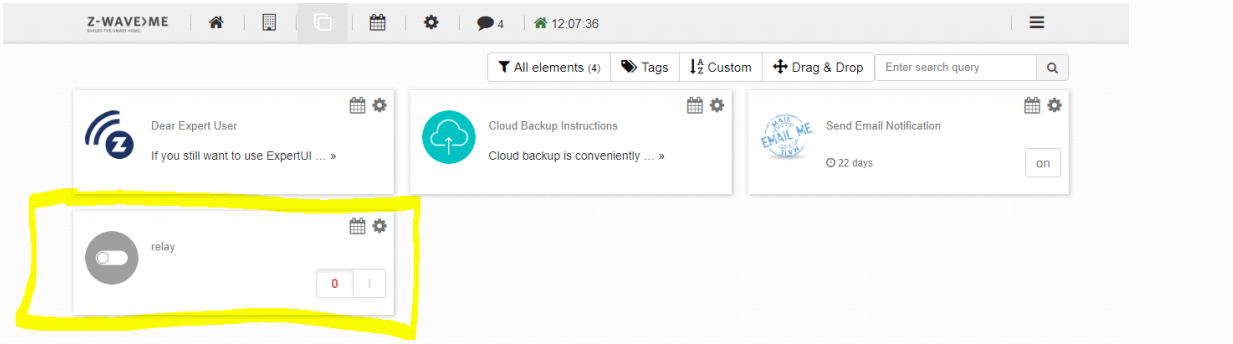


Figure 25 Wired relay on the elements tab

At each reboot, you will need to initialize the 25th pin to the output, so we will create a boot initialization script, you can find on the figure 26: “Menu → Applications → Local Applications → Custom JavaScript Code”.

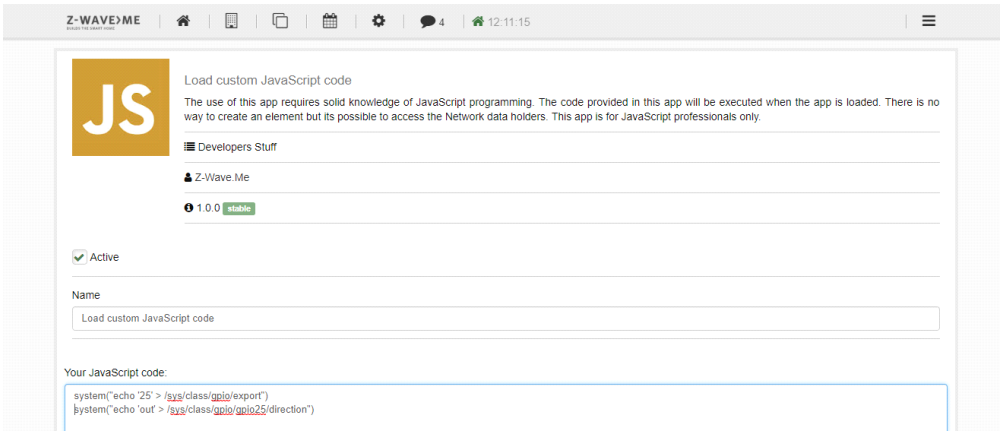


Figure 26 Java script for automatically relay configuration

Commands for initializing the 25th pin to output

```
$ system("echo '25' > /sys/class/gpio/export")
$ system("echo 'out' > /sys/class/gpio/gpio25/direction")
```

3.2.4 Automation Setup

More than fifty automation applications are built into Z-Way, and more than a hundred more can be downloaded from a free online store. Some of applications are presented on the following figure 27.

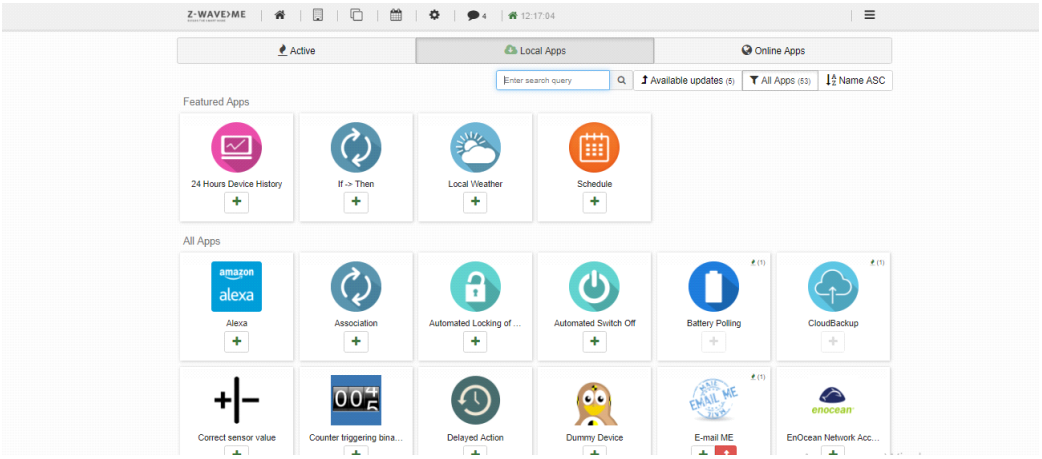


Figure 27 Z-wave.me inbuilt applications

There is an application "Smart Light", that you can see on the following figure 28, in the settings of which you only need to select a motion sensor and an LED lamp. Suppose the algorithm of operation is as follows: from 7:00 to 00:00 the lamp will turn on to the maximum, from 00:00 to 7:00 - only by 20%.

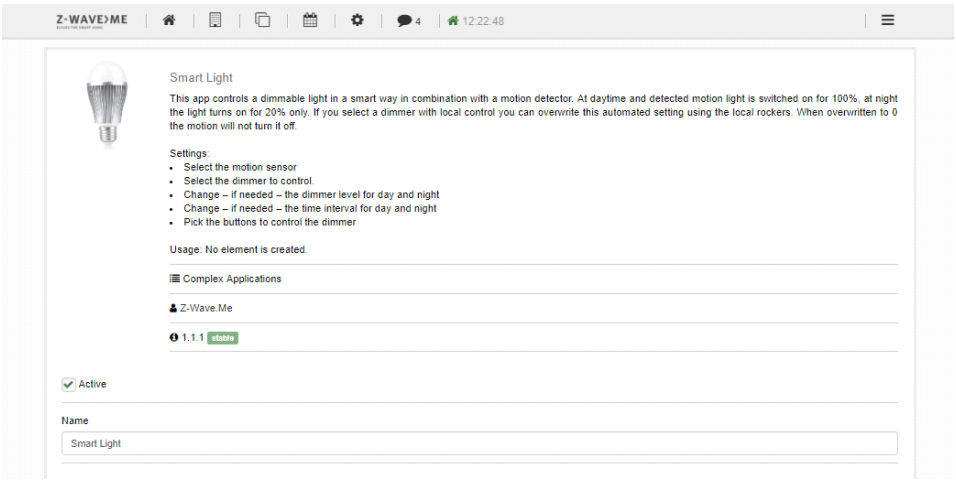


Figure 28 Smart light application

All the parameters of the application are presented on the figure 29.

Figure 29 *Application for configuration a light*

For comparison, recently the customer had an identical requirement for another controller, so I performed this task for all controllers tested in this work. To perform it on other controllers, knowledge of the Lua or Luup programming language is required, depending on the type of controller, but here you just need to insert the values you need.

One of the temperature sensors and relays can be activated to control the heater using the Virtual Thermostat application, presented on the following figure 30. In the application settings, you need to select a temperature sensor, a relay, set the hysteresis and the mode “Heating / Cooling”. In the “Heating” mode, the relay will turn off when the set temperature is reached.

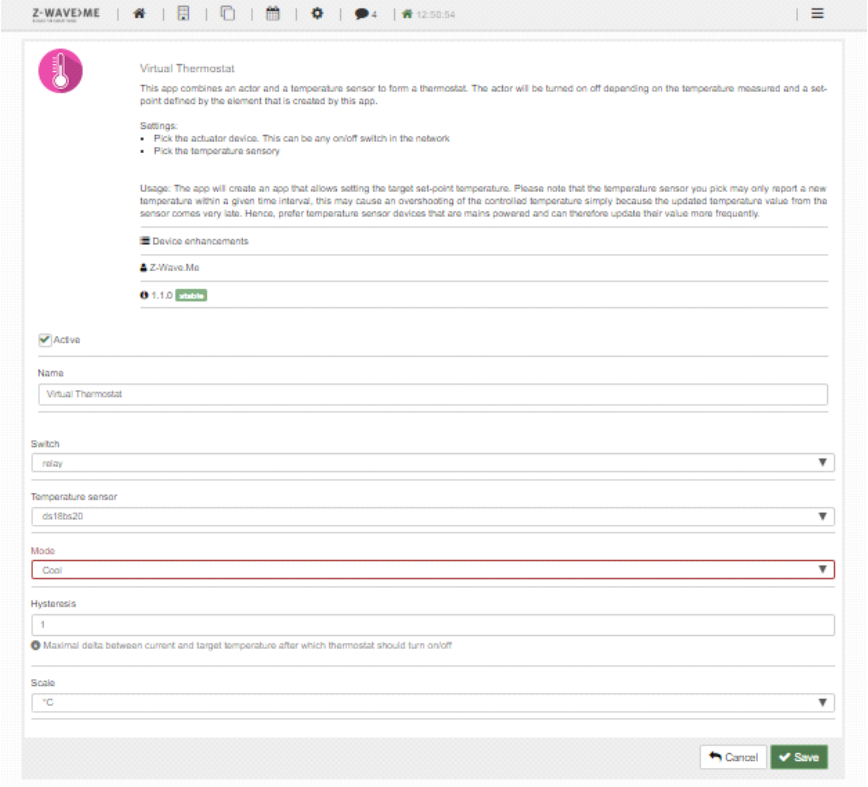


Figure 30 Application for activating relay based on temperature sensor

If none of the applications suits you, then you can always write your own in JavaScript. The automation system is completely open; you will find the source code on GitHub.

Also, if you want to expand the capabilities of your controller using a raspberry controller, then you just need to turn on the function of adding a device on your controller and turn on the learning mode on the raspberry board by pressing the button once and then hold it, so you can control or receive Information up to 10 devices from the raspberry board.

3.3 Project Bill

Z-uno is a Russian smart controller system that combines an arduino board, a z-wave module and software.

In this project, I have tried to use as many features of this device as possible, using pins of various types for communication with devices. Usually devices for z-wave protocol can measure one or two variables, for example, the most common motion sensor from fibaro also measures

temperature. In our case, we get a device that can combine up to 9 devices at the same time, not only sensors, but also executive devices. The price of this device will be below the average market price of any of the z-wave sensors. On the following figure 31 you can obtain the block-diagram of z-uno based system.

In this block diagram, you can see the ZUno with the sensors connected to it. This device will be displayed on the main panel of the home automation controller as one device with different channels. Zuno is a device that allows you to receive information from cheap sensors and transfers this information to the main controller, which is actually our main task. I connected wireless a temperature, humidity, carbon dioxide sensor, a door sensor, diodes and relay control to this board. The main task was to use as many communication protocols as possible to demonstrate the capabilities of this board. Its main drawback is the need for a wired connection of all sensors, that is, the board will be located near all devices, because in low-cost sensors, as a rule, there is no wireless communication.

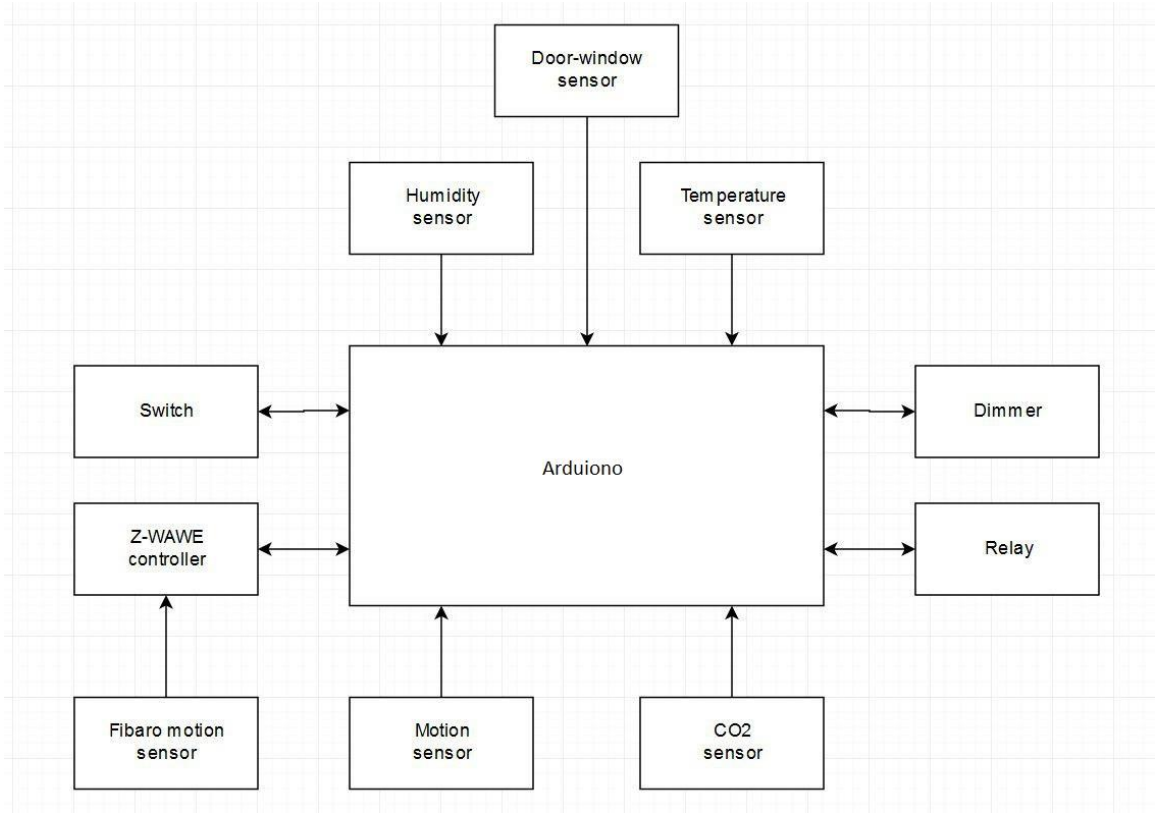


Figure 31 Block-diagram of the system

This integration has many advantages. First of all, it is an easy programming language Arduino IDE, which makes it easy to start working with equipment without relying on programming. Secondly, it is the ability to integrate a huge number of cheap sensors and devices designed for arduino devices. You will not have to solder anything, since arduino devices allow you to use breadboard. Also an important factor is a large community, where you can find many examples of working with this equipment and solving all problems that arise at the stage of familiarization with the equipment.

Usually, z-wave devices are designed to perform a specific function, adjusting the light or sensor of any parameter. Z-uno is a kind of Swiss knife in the z-wave world . You can make anything of it, at the same time you can connect up to 10 devices, which differ in communication protocols, purpose and power supply.

One of the main advantages of this device is the ability to expand the capabilities of your smart home by incorporating sensors that measure parameters such as radiation, carbon dioxide concentration, pressure, water level, earth humidity, and other parameters that cannot be measured using official devices compatible with z-wave protocol. In addition, you can fully customize the logic of any device that is, its sensitivity, sampling frequency, measurement scale etc.

3.3.1 Including of wireless z-wave sensors

You can find step by step instructions on how to set up the device in Appendix B.

3.3.2 Arduino code design

We should start with the fact that the Arduino programming language uses the basics of the C language.

- Conditional branching and loops: if-else, switch, for, while, do
- Variable types : byte, int, long, char
- Math operations : +, -, *, /, %, ++, --
- Comparisons: ==, !=, <, >, >=, <=

When launching the Arduino IDE program, we can observe two components that must necessarily be in any sketch.

- *setup* runs once device power on or wake up – the right place to set up peripherals and set pin modes

- *loop* runs eternally when device is idle – the right place to do sensors and buttons polling and control peripherals

  There is also and additional optional component *delay*, which hangs code execution for a defined period of time, but the device will still handle the z-wave communications even during execution *delay.*

  Z-uno has different types of inputs/outputs in order to be able to connect various types of sensors and devices.

- 26 pins for general purpose input/output

  digitalRed(pin) to read state and *digitalWtite(pin,value)* alter pin state

  3.3V is corresponds to HIGH, 0V corresponds to LOW

- ADC

  analogRead(pin) returns 0…1023 for the range of 0…3.3V

- PWM

  analogWrite(pin,value) accepts 0…225

- USB

  Serial monitor for debugging

- UART, SPI, I^2C, 1-Wire

  Popular buses for interaction with various sensors

- Interrupts and timers

  INT1, INT2, INT3, key scanner, GPT

The programming feature of the device, which is not found in either Arduino or C, is a way to define channels. Each channel represents one z-uno feature and the main rule is “one feature – one channel”.

There are five different types of channels

- Switch binary – relay
- Switch multilevel – dimmer
- Sensor binary – security sensor
- Sensor multilevel – environmental sensor
- Meter – meter

You can combine up to ten different channels in one device. All pins of the device are presented on the following figure 32.

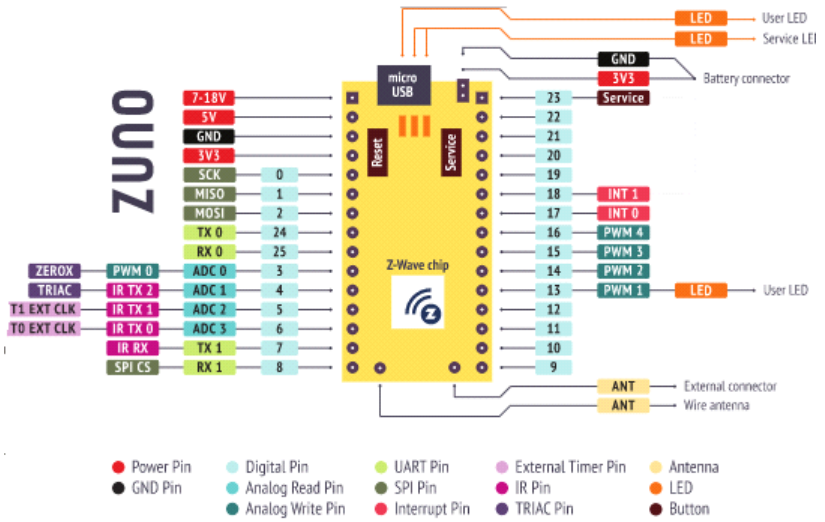


Figure 32 Z-Uno pinout

There are two different ways to define any channel. Let's take a look at temperature sensor .The code is presented on the following figure 33.

```
ZUNO_SENSOR_MULTILEVEL_TEMPERATURE(getterTemperature)

ZUNO_SENSOR_MULTILEVEL (

ZUNO_SENSOR_MULTILEVEL_TYPE_TEMPERATURE,
SENSOR_MULTILEVEL_SCALE_CELSIUS,
SENSOR_MULTILEVEL_SIZE_TWO_BYTES,
SENSOR_MULTILEVEL_PRECISION_TWO_DECIMALS, getterTemp)
```

Figure 33 code for defining a channel of z-uno

In the first case, we indicate which parameter we will measure together with the type of sensor, after which we will accept the standard settings for this type of sensors. A list of all variables can be found in the device documentation. In the second case, we specify all the parameters separately, that is, the measured value, in which dimension we want to see the result, the size of the transmitted data packet and the precision. In both cases, one parameter is required - is the name of the function, which we will further be used in our sketch for this sensor. Since in this case the temperature sensor has been presented, the only possible option is a *getter*, since we can only get data from the device.

While working with any device, we have two options: get a value from it or set a new value for it, for this we should use the *getter* and setter functions. Lets consider an example of a switch binary channel shown on the figure 34:

```
ZUNO_SETUP_CHANNELS(ZUNO_SWITCH_BINARY(getter,setter))
```

Figure 34 code for getting or setting values to z-uno

- getter – returns the channel value

  Called on a Get request from the z-wave network (to answer the asker)

  Called on a Set received from the z-wave network (to report to the LifeLine group)

  Called from the user code to send an unsolicited report to the LifeLine group

- setter – accepts new values on the channel

All channels are numbered in the order to which you write them, and after that, when writing code, we can use their serial number to send information to them:

zunoSendReport(channelNumber) – send an unsolicited report

On the following figure 35, you can see the block diagram, which shows us how the device communicates with other components of z-wave network.

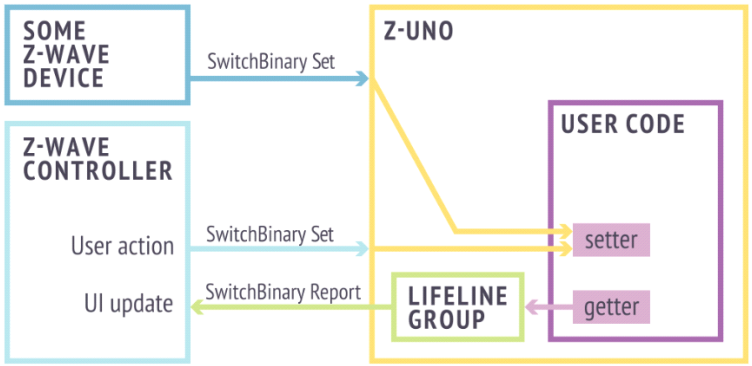


Figure 35 Z-Uno setting new values

This is all the necessary information to start working with the device, next we will look at examples of sketches for single sensors and an example of how z-uno works as a multichannel device.

3.3.3 Including of wired sensors

Let's start by considering a simple example of a door and window sensor, since it takes only 2 states, and we cannot set any value on the device, but can only read its state, this is an excellent example for the concept of code structure, as it always will be the same. The following demonstrates figure 36 an example.

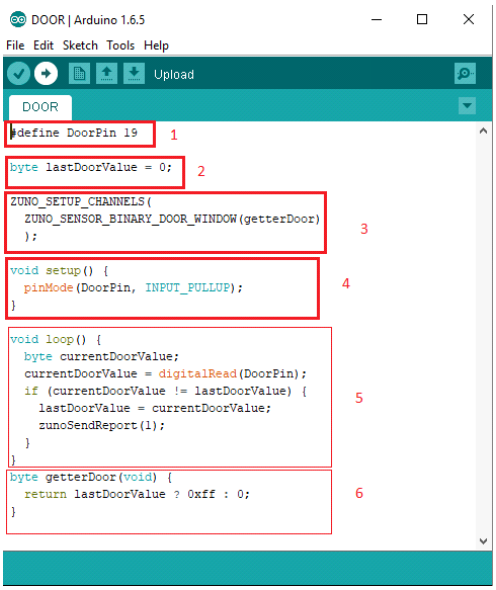


Figure 36 Z-Uno code structure

- At the beginning of the sketch, we always determine the pin number of the device to which the device is connected and assign a name to it.

- In the second part, we define global variables that will be used in the sketch.

- Then we define the channels, since we have one device connected, then we have ti create only one channel.

- In the fourth part, we determine the nature of pin operation, and this is only necessary for digital signals, the analog ones are automatically configured.

- Loop is the main part of the code, where we determine how the device will work and this part will be repeated until we turn off the device. In this case, the order of operation of the device is very simple: we send the status value of the device to the device every time it changes its value, or as written in the code, when the previous value is not equal to the current one.

- In the last part of the code, we call the function, with the name that was set when the channel was created, and display the current state of the instrument on the main panel in the user interface of our controller.

Now we can upload the sketch into the device by clicking on the Upload button. Let's test equipment with raspberry controller. Since zuno and z-way software, the installation of which we made in the last chapter, is manufactured by the same company, when adding a device, we can find z-uno in supported devices, we can see it on the following figure 37.

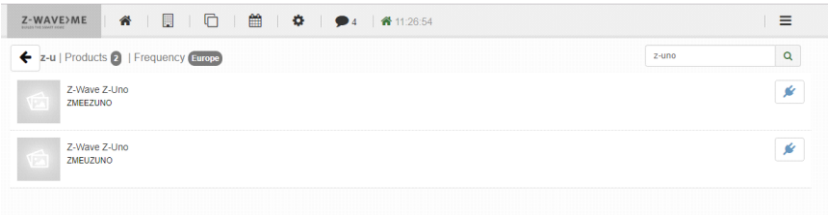


Figure 37 Z-Uno template in z-wave.me

After adding the device, we can see it on the element's tab. Z-uno creates two devices, one to indicate the current state of the device, the second for security settings, with which you can configure alarms and alerts. On the next figure 38, you can see door-window sensor at z-wave.me elements tab.

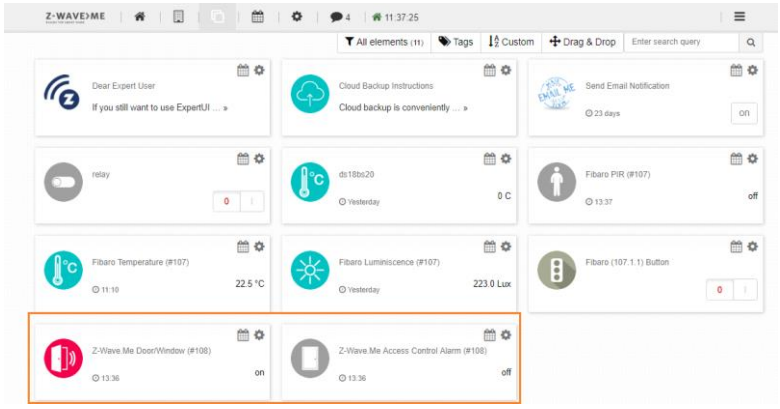


Figure 38 Door-window sensor on elements tab

Consider the next example where z-uno will act as a multichannel device. Let's connect an additional temperature sensor to the existing door and window sensor. On the next figure 39 you can see the device with connected door-window sensor and temperature sensor.

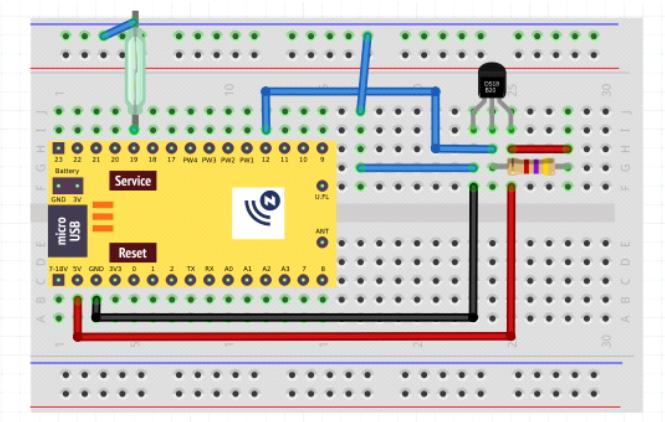


Figure 39 Connecting of temperature and door-window sensors to z-uno

On the following figure 40 we can see code for this example.

Door_temp §

```
// add library ds18b20
#include "ZUNO_DS18B20.h"

// pin connection ds18b20
#define PIN_DS18B20 12
#define DoorPin 19

OneWire ow(PIN_DS18B20);

// onewire connection temperature sensors
DS18B20Sensor ds1820(&ow);

byte addr1[8];
byte lastDoorValue = 0;
int temp; // here we will store the temperature

// set up channel
ZUNO_SETUP_CHANNELS(
   ZUNO_SENSOR_MULTILEVEL(ZUNO_SENSOR_MULTILEVEL_TYPE_TEMPERATURE,
                          SENSOR_MULTILEVEL_SCALE_CELSIUS,
                          SENSOR_MULTILEVEL_SIZE_TWO_BYTES,
                          SENSOR_MULTILEVEL_PRECISION_TWO_DECIMALS,
                          getterTemp),
  ZUNO_SENSOR_BINARY_DOOR_WINDOW(getterDoor)
);
```

Figure 40 Example of code for multichannel device

For a large number of sensors compatible with Arduino, libraries for z-uno were created, a list of which can be found on the github of the device. This temperature sensor is one of them, so initially we add a library for this sensor. Code is presented on the following figure 41.

The second new point is that we use 1-wire communication, respectively, after defining the pins, we should inform our device about this.

Door_temp §

```
void loop() {
   byte currentDoorValue;

    ds1820.scanAloneSensor(addr1);
    // obtaining readings from the sensor ds18b20
    float temerature = ds1820.getTemperature(addr1)
    // make scaled word value for report
    temp=int(temerature*100);
    Serial.print("Your sensor address is: ");
    for(int i = 0; i < 8; i++) {
            // print OneWire code
            Serial.print(addr1[i], HEX);
            Serial.print(" ");
    }
    Serial.println();

    Serial.print("Temperature: ");
    Serial.println(temerature);

    // send data to channel
    zunoSendReport(1);


    // Door/Window sensor
  currentDoorValue = digitalRead(DoorPin);
  if (currentDoorValue != lastDoorValue) {
    lastDoorValue = currentDoorValue;
```

Figure 41 Example of using libraries

Next, we can observe the basic principle of working with the z-uno code - is the using of already created libraries. On the github and the zwave-me forum, you can find a lot of examples and libraries, the only thing you need to be able to do is to use it correctly. If you follow all the instructions I described earlier and correctly reuse the codes already created for the devices, you will not have any problems with setting up any equipment. This is one of the advantages of the device: you do not have to be a programmer and create codes for each device, you only need to understand the structure and correctly use the library of already developed codes.

Upload a proven sketch. The important point is that when changing the number of channels or its type, each time you need to remove the device from the controller and add it again. On the next figure 42, we can see graphical presentation of our sensors.

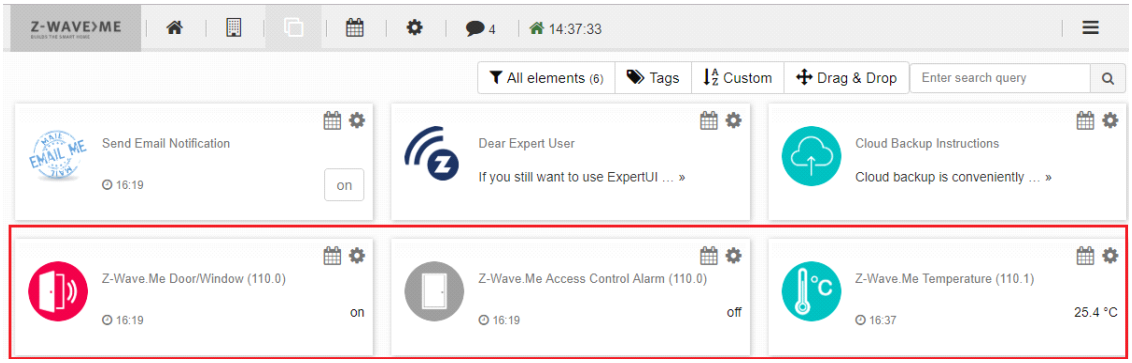


Figure 42 Multichannel z-uno on elements tab

The last example we consider is the complete execution of the task. In this case, it will be a multichannel device that includes 9 different types of devices, namely, a motion sensor, a humidity sensor, a temperature sensor, a carbon dioxide concentration sensor, a door and window sensor, a relay for switching on a conventional lamp, a switching on LED and dimmer. For this, I used different types of communication: 1-wire for the temperature sensor, UART for the carbon dioxide sensor, PWM for the dimmer, and for different cases, both analog and digital outputs/inputs were used. On the elements tab of z-wave.me we can see all the elements, as it is shown in the following figure 43.

An association group was also created, which I will discuss in the next chapter.

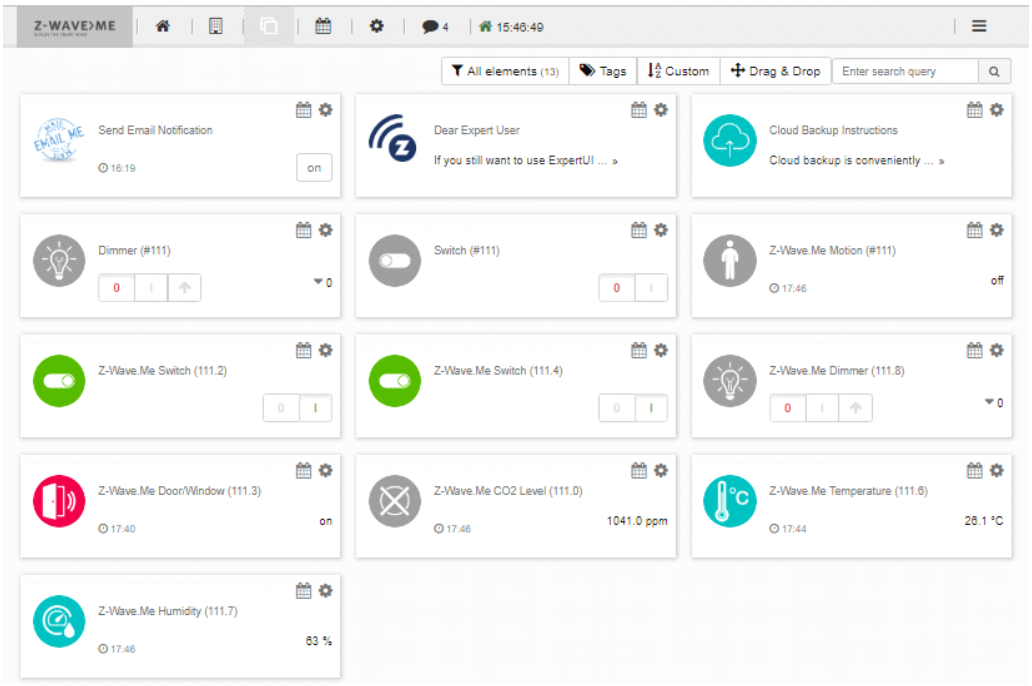


Figure 43 Z-Uno as eight-channel device

All functions work, the response time is minimal, which allows us to state with confidence that it is fully compatible with the raspberry controller.

3.3.4 Associations

Each home automation controller have a scene creation function. This allows you to connect different devices to each other: the most clear example is the turning on a light when motion is detected. After creating scenes, information from one device goes to the the main controller, from which information will be redirected to some active device such as relay or dimmer. This method is not very effective, as it slows the controller when a large number of scenes was created, as well as all the processes will not occur instantaneously, as time for the controller to process information is needed.

Groups of associations are presented in all z-wave devices and z-uno is not an exception. These groups allow you to send information directly from one device to another one or to several devices at once, without sending an information to the main controller.

If you are going to create associations for two or more devices connected to the z-uno, then it will not make any sense, since this can be done in the sketch itself. As an example, I have linked a motion sensor with a LED that turns on without delay whenever motion is detected.

If it is necessary to connect any device connected to the terminal with another device located on this network via z-wave, then they need to be assigned to an association group, after which it will send information to all the devices in this group.

First, we need to indicate that we are going to use groups of associations, as it is shown on the next figure 44.

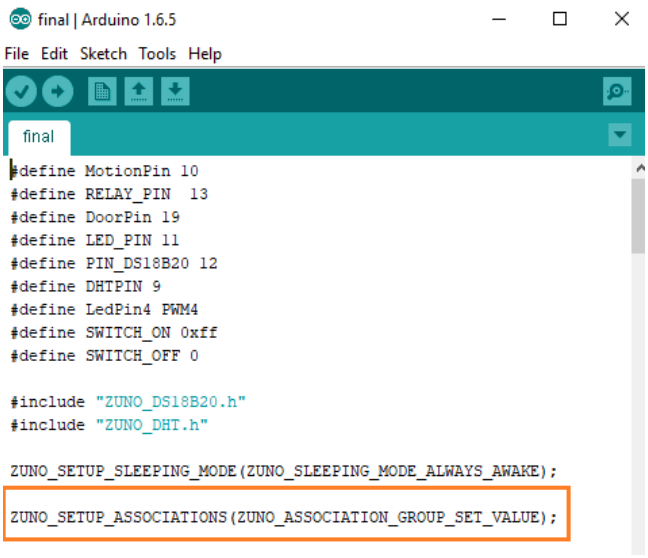


Figure 44 Association group activation

Then, while sending information to the controller, we also indicate that we want to send this information to the group of associations and indicate what information this signal means. On the following figure 45 we can see the code, with help of witch we can send command to our group of associations.

```
void loop() {
  byte currentMotionValue;
  byte currentDoorValue;
   int i = 0;

    // Trigger motion and wait for relax (about 5 sec) before repo
    currentMotionValue = digitalRead(MotionPin);
  if (currentMotionValue ) {
    if (relaxMotion == 0) {
      lastMotionValue = 1;
      zunoSendReport(1);
      zunoSendToGroupSetValueCommand(CTRL_GROUP_1, 1);
      digitalWrite(LED_PIN, HIGH);
  }
```

Figure 45 Adding of device to an association group

In this case, we send information about motion detection by the sensor to association group 1, and upon receipt of this command, the actuator will move to the ON position.

There are four types of control commands

- Set new value to relays and dimmers (Basic Set)
- Start/stop dimming (Switch multilevel start/stop)
- Active a scene (Scene activation set)
- Operate door locks (Door lock set)

All types of commands for each group are listed in the documentation for the device.

In total, there are 5 groups of associations available, each of which can include up to 5 devices. On the following figure, we can see block-diagram, which help us to understand how association groups work in the device.

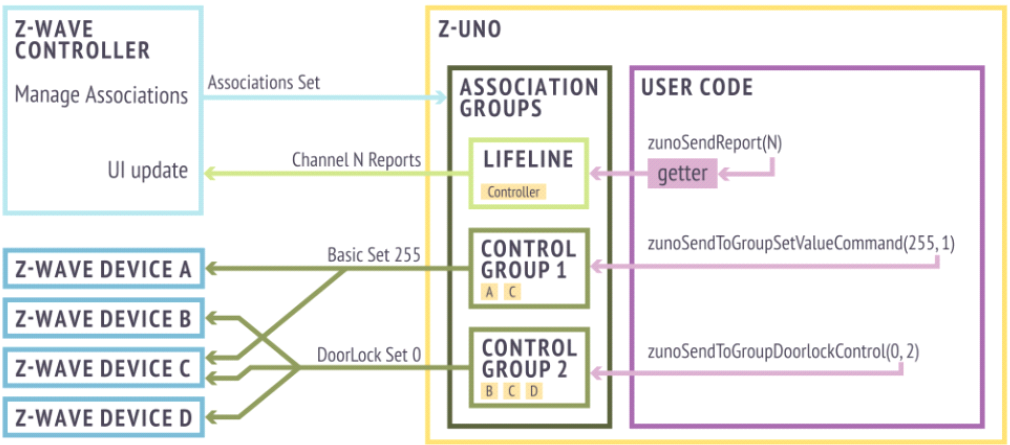


Figure 46 Z-Uno association`s communication

In different controllers of home automation, group of associations and the connection of devices to them is set differently, but the most friDoorendly and intuitive this function is performed in Zipato controllers.

After including of the device to zipato controller you should go to Device manager → Z-uno settings→advanced→Manage associations, as it is shown on the figure 47.

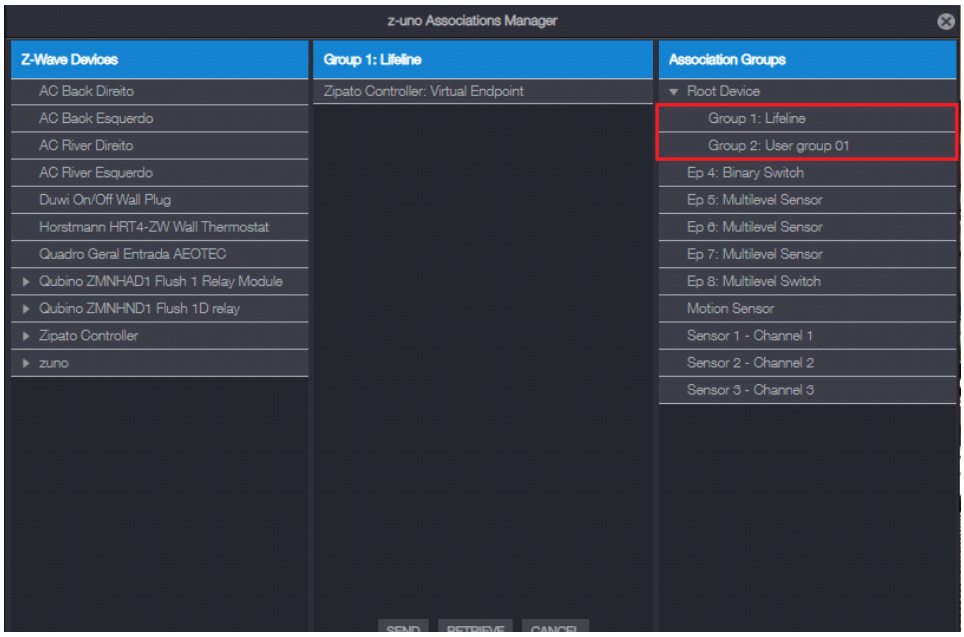


Figure 47 Created association group in Zipato controller

Lifeline is a group that each device has with which the device communicates with the main controller, the second group is the group we created. Now we can connect our motion sensor with any relay or dimmer directly.

3.3.5 Compatibility with different controllers

As we already understood, z-uno is fully compatible with the raspberry controller, since the z-way software and the device itself are made by one manufacturer and they are fully compatibles. Now let's test it with controllers from other companies. Let's start with the zipabox controller.

After adding a device to the controller's network, we need to find it in the device manager, go to its settings and enable the function “show as a device”, that shown on the next figure 48.

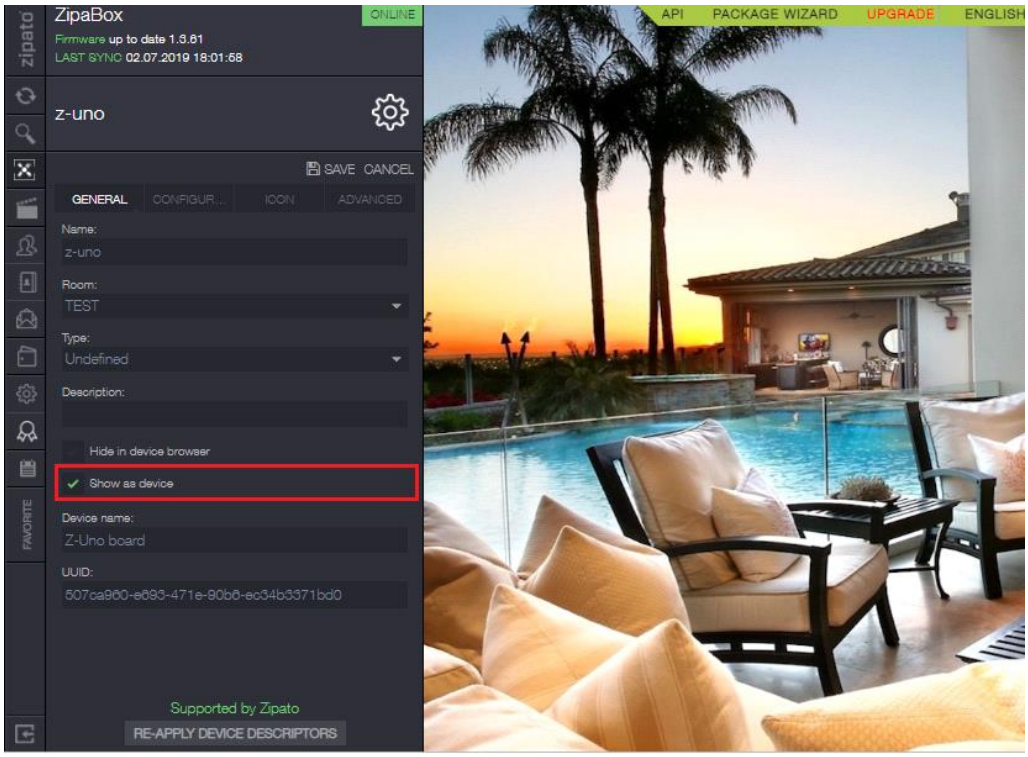


Figure 48 Activation of z-uno display in zipabox

After you have to go to configuration tab and set parameter number 12 to 1, it will help the controller to recognize all type of sensors, connected to the device. You can see configuration tab on the figure 49.

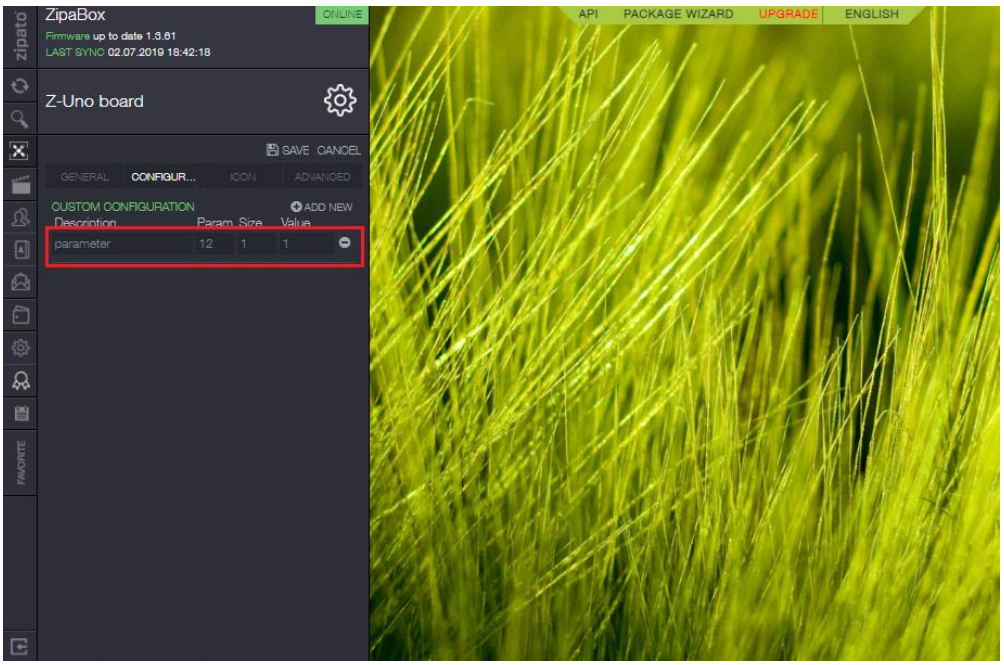


Figure 49 Configuring z-uno as a multichannel device in zipabox

Now we can find it in the device browser and make sure that it works. On the next [figure 50](#) we can see all parameters, that zippato can recognize from unknown device.

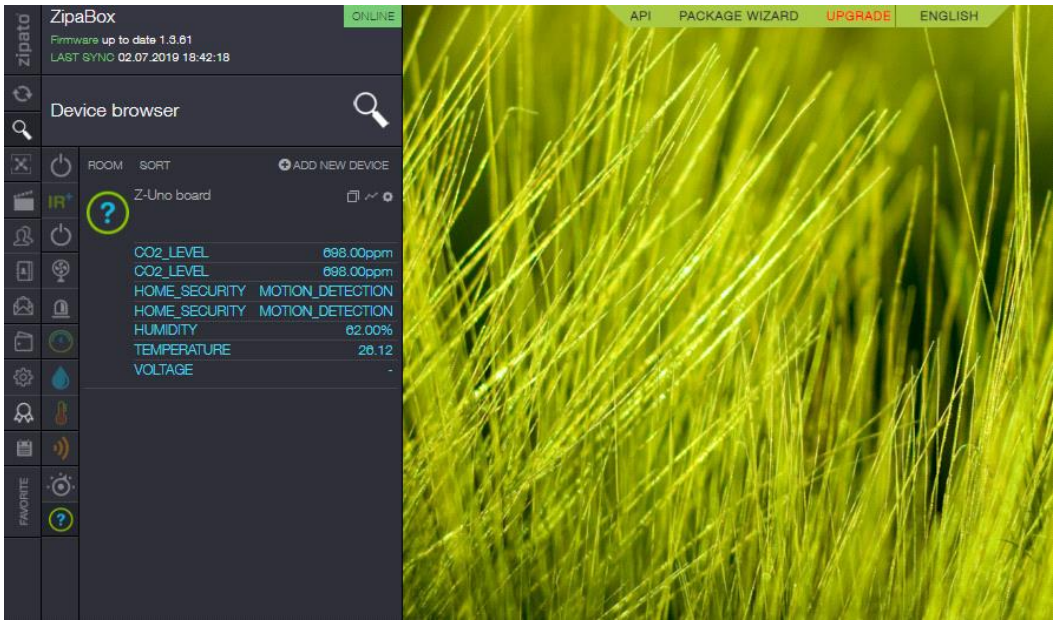


Figure 50 Z-Uno on device tab of zipabox

As we can see, all the parameters from our sensors are shown, with the exception of the door and window sensors, which unfortunately is not displayed. Also, zipato does not add to the panel of devices each of the connected devices separately and there is no possibility to control the relay and turning on the light. But they still can be controlled by creating scenes, as shown on the [figure 51](#).

Figure 51 Z-uno switches in Zipabox

All channels in the scene creation editor are displayed correctly and work with all z-wave devices.

This means that if you need to automate some place, let's say watering a flower with low soil moisture, then you will never have to manually turn on or turn off the faucet and the device will perfectly cope with this task. However, if you want to make a switch on the basis of the device, then you will be disappointed.

However, if we connect the z-uno with several channels, for example, with a door sensor and a temperature sensor, we will see all the parameters, as on the following figure 52. It means that zipato controllers do not fully support devices with a large number of channels.

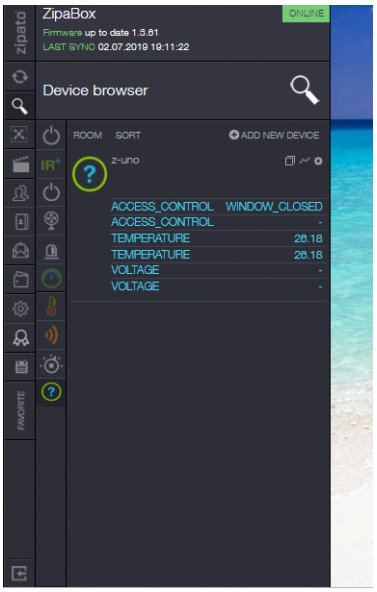


Figure 52 Z-Uno with two channels in zipabox

Vera controller

First of all, after connecting the device to the vera controller, we also need to change the parameter 12 of the connected device to the value 1. The meaning of this action is that initially the data channel of the device is in the notification mode, so for better recognition of the channels by the controller we need to change this value to 1.

Let's start by adding our device by connecting a door and window sensor and a temperature sensor to it. As we can see on the figure 53, everything works correctly and all values are displayed.

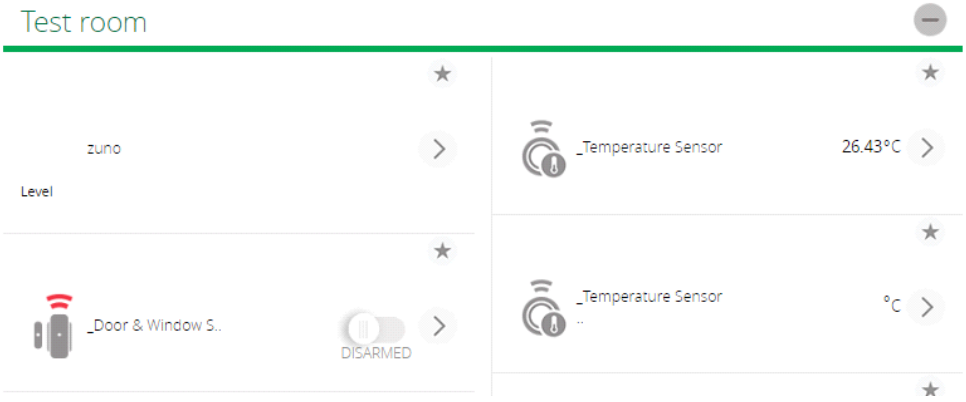


Figure 53 Z-uno with two channels in Vera edge

After connecting the device with nine channels, on the image 54 we can see all the buttons, the values are also displayed correctly, all the switches and dimer work properly, except for the door and window sensor. This sensor is displayed, but does not change its state. Also, when working with nine channels, there is a big delay, that is, after pressing the light off button, the action will take place after ten seconds. This tells us that the situation with the vera controller is better than with the zipato controller in terms of display and the ability to control devices, but worse in terms of speed.

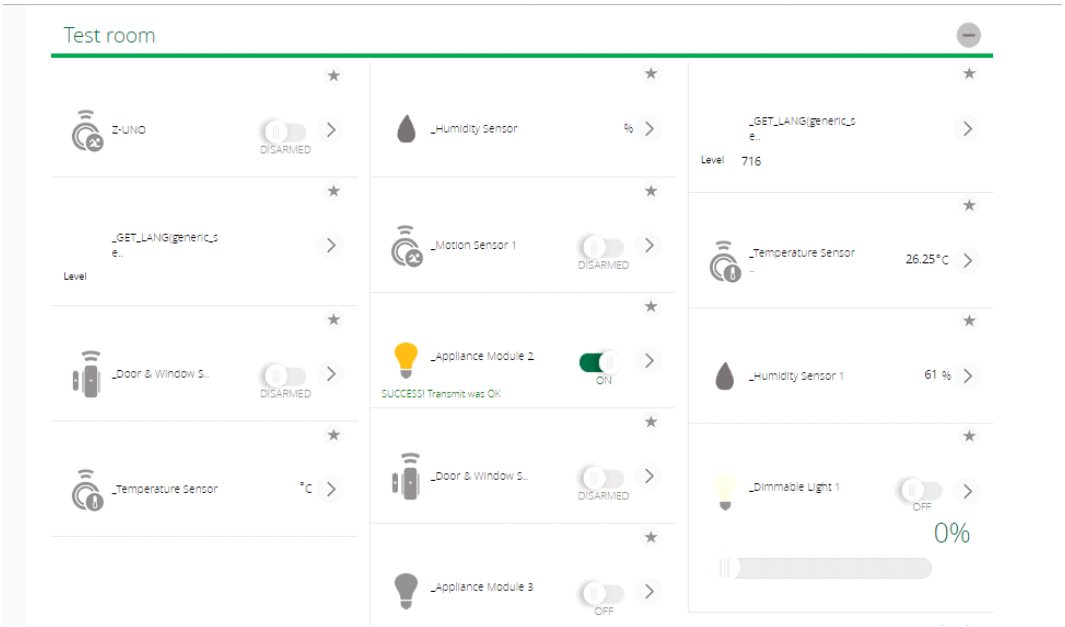


Figure 54 Z-Uno with eight channels in Vera edge

Fibaro

Fibaro controller showed the best results, except for the raspberry controller. It displays all the channels, except for the door and window sensor, the speed at a perfect level, which is almost excellent result. On the following figure 55 we can see the device with 9 channels, connected to fibaro home center.

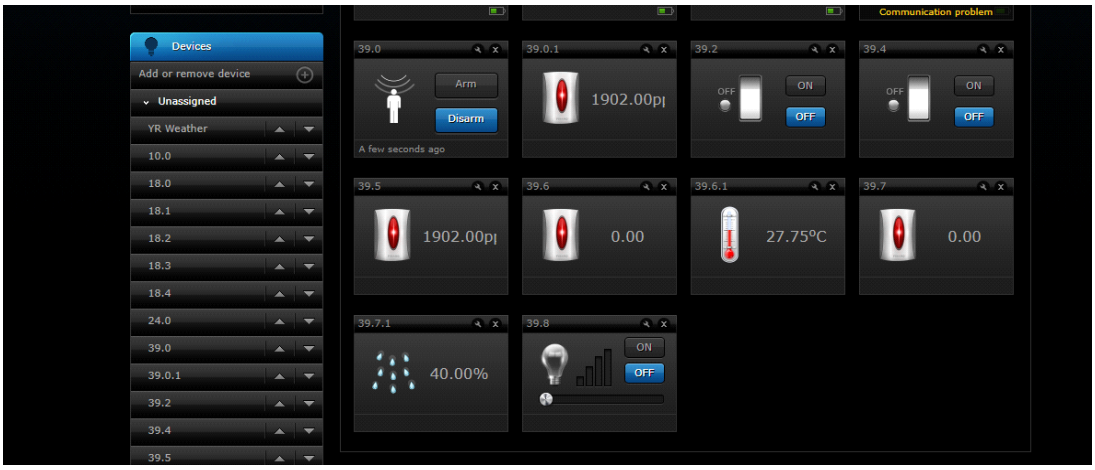


Figure 55 Z-Uno with eight channels in Fibaro home center lite

4. Conclusions

The z-wave protocol is a very convenient way to automate your smart home, but as you can see, it has several disadvantages.

Let's start with the shortcomings: the price of most certified devices is too high, that is, to implement these functions using other protocols, less money is often required than to implement these functions using z-wave protocol. The number of certified devices is also limited, which means that it is impossible to carry out certain tasks that might interest you. It also remains an open question about the speed of many systems: consider a specific example - often the vera controller may not work with the equipment at a stated distance, and the fibaro controller can update its software for five hours.

However, the advantages of this protocol are much greater. Firstly, the protocol is open, and with the help of source codes you can, for example, write the software for your controller by yourself. Secondly, this protocol does not have compatibility problems of different devices connected to the same network, what happens in other protocols. In addition, most controllers allow you to create scripts using various programming languages, which allows you to customize the functionality of the equipment more flexibly.

I had the best experience with the raspberry controller, in particular, installing the Z-Way software on it. The company's policy is extremely simple - to make a controller that will not experience any difficulties with any z-wave equipment. For many companies, this logic is lacking and they intentionally improve the quality of the work of the equipment they produce with their controllers. There are also a huge number of different applications for realizing any of your needs, and if you still don’t find a suitable application among hundreds of applications, you can always create one by yourself.

In this project, ways of integrating low-cost sensors were considered, in particular with the help of z-uno. However, it was presented only as a device working with arduino devices. In fact, you can purchase a special shield for this board, which allows you to install it in your office and connect all the usual devices simultaneously with the Arduino sensors. It is also possible to connect batteries to the board, on which it can work for about a year. This device is almost perfect, except for compatibility with other controllers. I found a comment from developers

about this, and they explain it by the fact that the development team is very small and there is simply no time to adjust the hardware compatibility with all controllers. However, imagine that you buy one device for any purpose, it is cheaper than any other officially device, usually with two channels, and the only thing you need to do is download the desired program the device. The input threshold for this device also hampers this. A large number of customers, in my experience, face problems even with devices that need to be taken out of the box and plugged in, and the situation with the z-uno will seem to them quite horrible, since many simply do not have time to tune the equipment. The solution to this problem also exists - the creation of a database of sketches, so that all the client needs to do is press the download button.

Now let's imagine that you need to automate any place where you need to use unusual sensors. For example, you want to remotely monitor the stagnation of your plant and set up automatic care for it. In this case, the z-uno is ideally suited for this task: we can connect all the necessary sensors, such as an earth humidity sensor, a light sensor, several relays, and so on, so that in the dark the necessary light automatically turns on, the irrigation automatically turns on when insufficient humidity and all indicators will be displayed on your phone. In this case, all devices will be in one place and the possibilities of the the device will be revealed to the maximum.

The case of raspberries is not so simple, since connecting low-cost sensors requires an understanding of the Linux system, which is a problem for many, including me, and a lot of time is wasted because of this. It is also worth remembering that the price of a raspberry board with a z-wave transmitter exceeds the price of cheap controllers for a smart home for most companies. Most people want everything to be simple, fast and clear, and few people will have to overpay for the fact that you will have some opportunities for activating which you have to spend a lot of time. However, in the case of z-way, it is worth it.

Future developments

Both tested devices are constantly improving, the new raspberry model came out last month, the z-uno is different from its first model, many additional features appear, like changing the transmission frequency, smaller size, increasing the range and so on. I believe that with proper marketing, which is now simply lacking and refinement of compatibility with all controllers, this device can greatly change the way people see z-wave devices, forcing other manufacturers to

either reduce the overestimated cost of equipment or create a similar device that has a large number different channels and having the same great functionality.

This means that in many cases, it is already quite difficult to find a replacement or alternative to this device, and if you wish, using this device you can save a lot of money instead of buying expensive equipment.

Z-wave is not going to stop its development, but on the contrary, the number of compatible devices is only growing, number of those who want to make their home smart is only increasing, as the number of companies offering their products. Two ways to save your money on the buying of expensive equipment, as well as the whole world, will not stand still. What specifically will be introduced or development plans, the companies do not provide, but we can observe the development process and be confident in the future of these devices.

Documentary References

- Internet resource https://progress.online/tehnologii/1686-istoriya-vozniknoveniya-i-razvitiya-umnogo-doma
- Internet resource http://arprime.ru/avtomatizacia/domashnyaya-avtomatizatsiya-umnyy-dom-i-sistemy-upravleniya
- Internet resource http://www.3acom.ru/umnyy-dom.html
- Nikolai Zhogov - Article : Communication protocols for smart home
- Egor Morozov – Article : Smart home - theory and implementation based on the Z-Wave protocol
- Internet resource http://www.rovdo.com/z-wave-stack
- Article : Quick introduction in Z-Uno https://z-uno.z-wave.me/getting-started/quick-introduction-in-z-uno/
- Article: Z-Uno Quick start guide https://z-uno.z-wave.me/getting-started/quick-introduction-in-z-uno/

Appendix A. How to install z-way to Raspberry PI.

To start working with raspberry, we need to install software on it. To do this, we first need to download it. We can do this by visiting official raspberry site. There we can see three files that are distinguished by the presence of the user interface and the presence of some utilities, any of them will suit us. Click to “Download ZIP”.

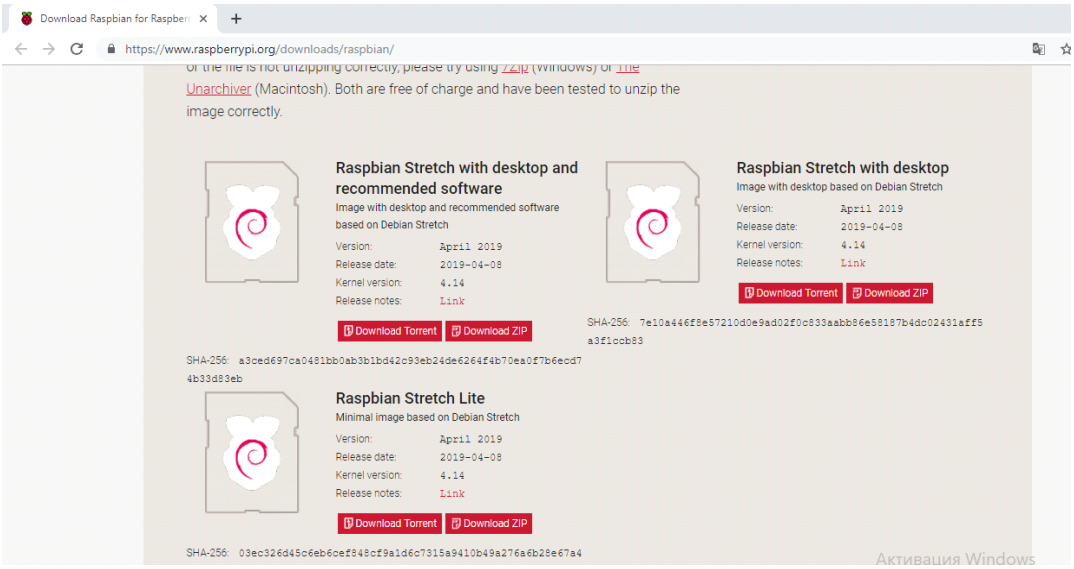


Figure 56 Raspbian download page

After downloading, we need to unzip the archive into a separate folder, and in the end, we should find the disk image there.

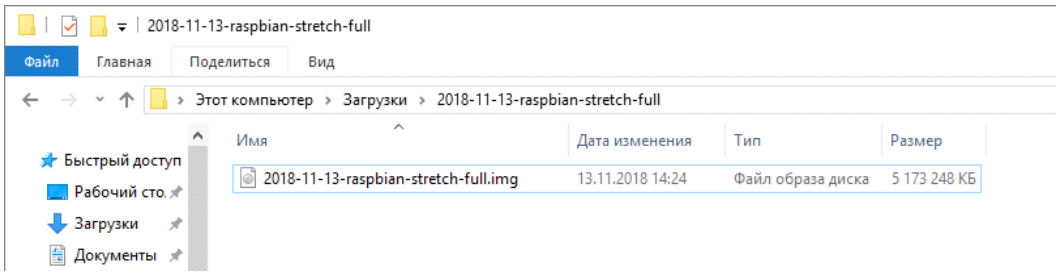


Figure 57 Image of Raspbian software

Next, we need to burn the image of this disk to our ssd card. To do this, we use the program “Etcher”.

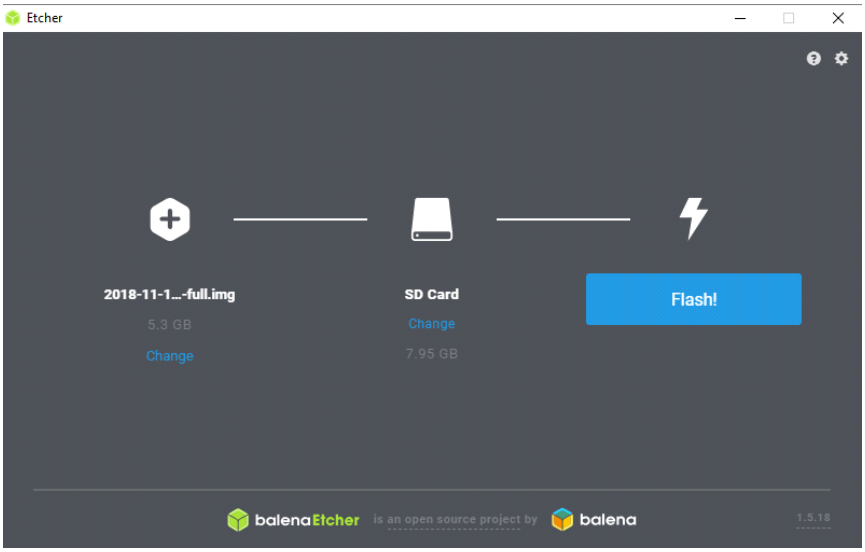


Figure 58 Etcher application

In this program, we need to specify the disk image we downloaded, the ssd card connected to the computer using a card reader and press the flash button.

Then it is necessary to disconnect from the computer card reader with an ssd card and turn it on again. In “my computer”, you should see two parts of our system, we need to open a removable disk called “boot”. In this disk, we have to create a new text document called "ssh" and remove the extension from it.

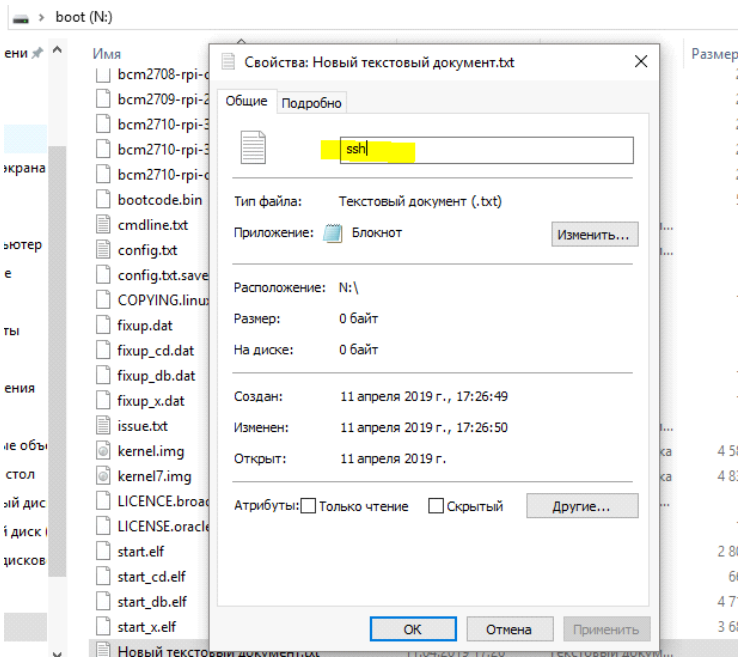


Figure 59 Creating the ssh file

Now we can connect our ssd card to the raspberry board. Ssd card slot is on the bottom of the board.

Figure 60 Port for micro sd card in Raspberry PI

Next, we have to connect our raspberry board to the Internet using a cable, wrap it to the same network to which your computer is connected, and connect the power supply and the Razberry board to it.

Figure 61 Conected Raspberry PI

To get access to the raspberry operating system, we need to install a program for remote access to our device. This program is called “PuTTY”.

We also need to know the IP address of our device. For this, there are many programs, such as “Angry Ip scanner”.

After opening this program, we will see this window, where we will need to click on the start button.

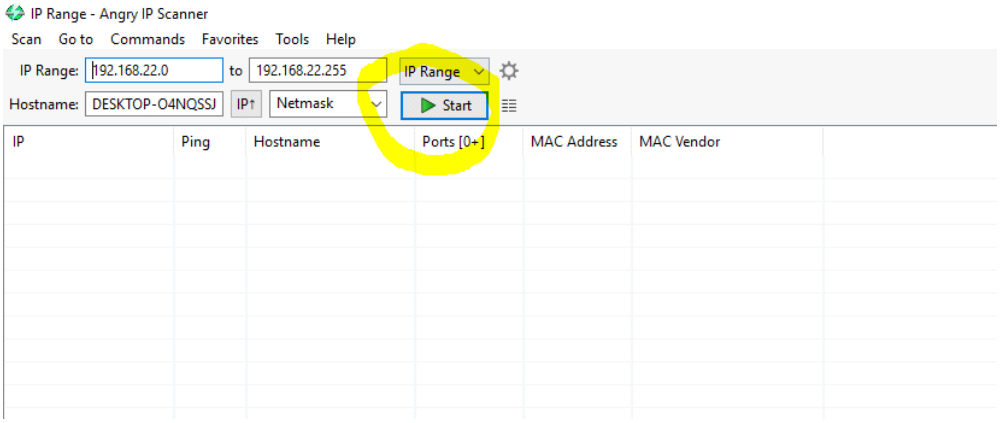


Figure 62 Angry IP interface

In the list of ip addresses, we need to find the address of our raspberry board.

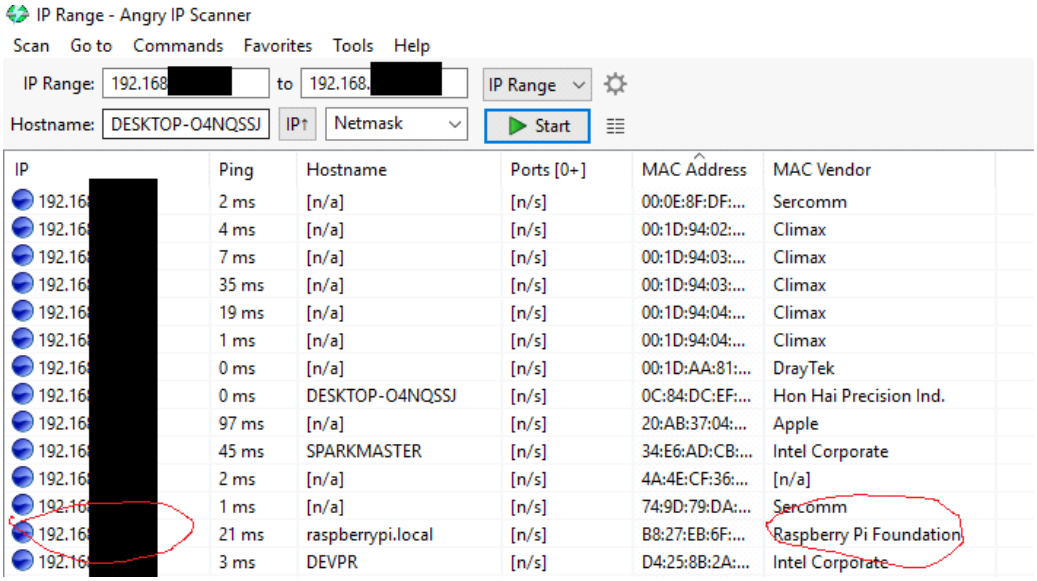


Figure 63 Raspberry`s IP in Ip scanner

Now let's open the PuTTY program, the following window should appear, where we specify the IP address of our device and select the ssh communication protocol.

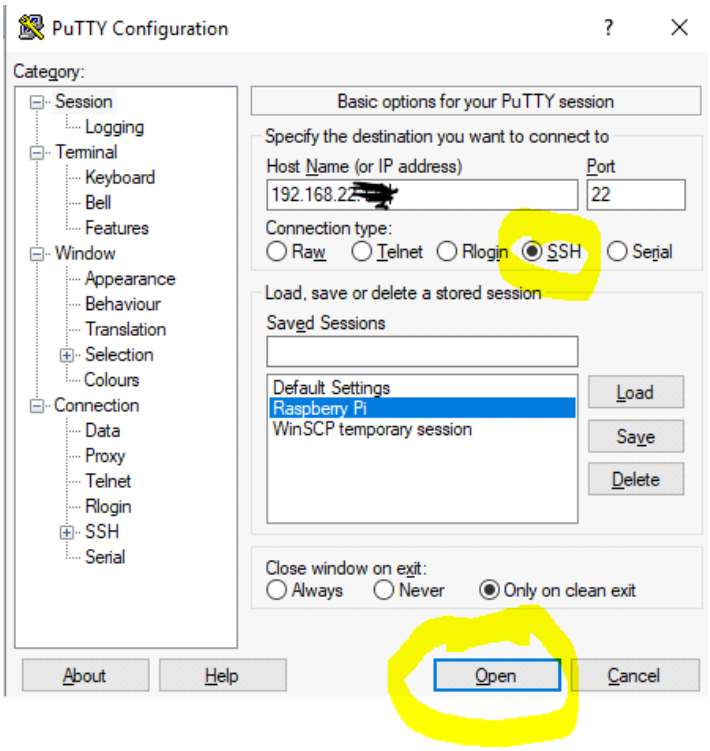


Figure 64 PuTTY`s interface

In the next window, we need to enter the login and password from our system. Since we did not change it from the factory settings, then login is “pi”, password is “raspberry”.

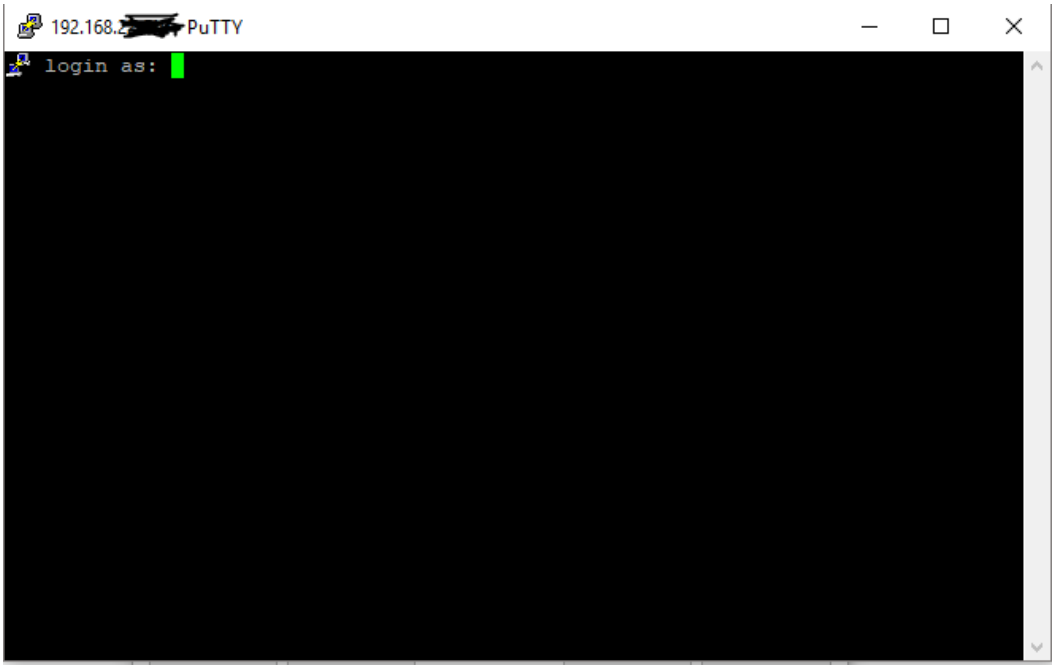


Figure 65 Initial page of Raspbian software

In the console that appears, we should enter the command to install the z- wave package for our raspberry board, for this, we will enter the command

```
$ wget -q -O - razberry.z-wave.me/install | sudo bash
```

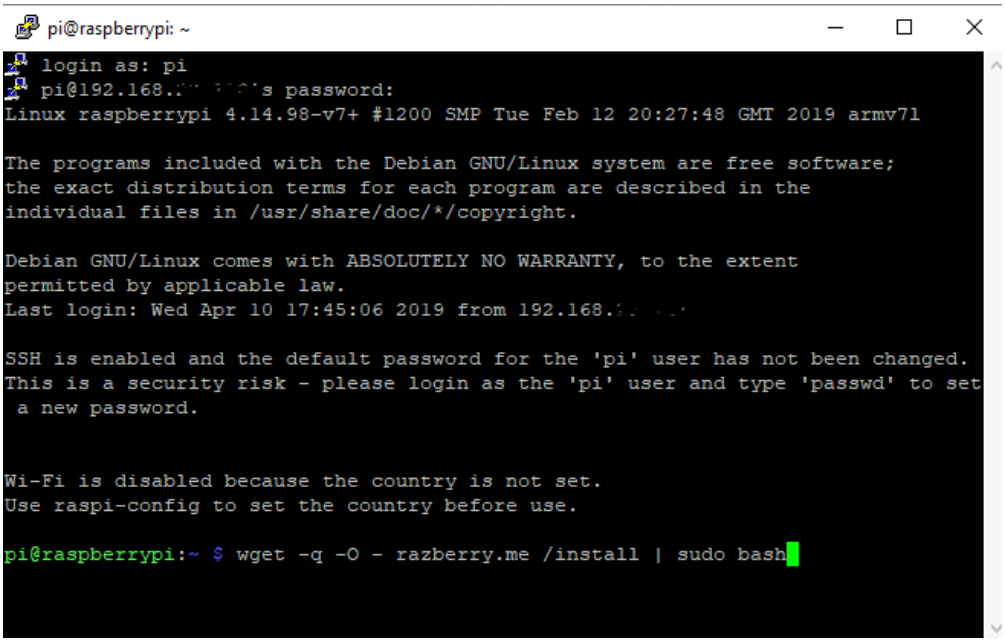


Figure 66 Installing z-wave.me to raspbian

After installing of this package, u have to restart your system to apply all changes, type this command to reboot the system.

```
$ sudo reboot
```

After installing the package, we can connect to the z-way site and gain access to the control panel of our controller. To do this, we should go to z-way finder, you can see the link in the following screenshot, click on the direct connection button and click on the IP address of our device.

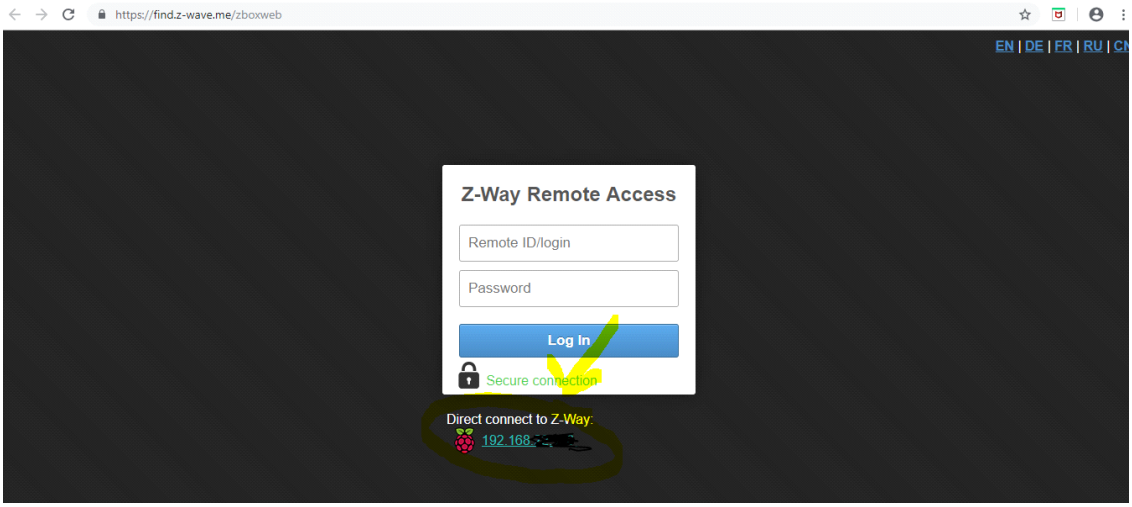


Figure 67 Z-wave.me finder

Appendix B. Setup of Z-Uno.

The first step is to check the condition of the device. At the beginning, a simple program is preinstalled that allows you to adjust the flashing intensity of the built-in lamp. To do this, simply connect the Arduino board to your computer for power supply. Let's try to add this device to the Vera Edge controller. To do this, go to the menu for adding a new device and select a generic z-wave device, as there is no special installation path, as for some devices.

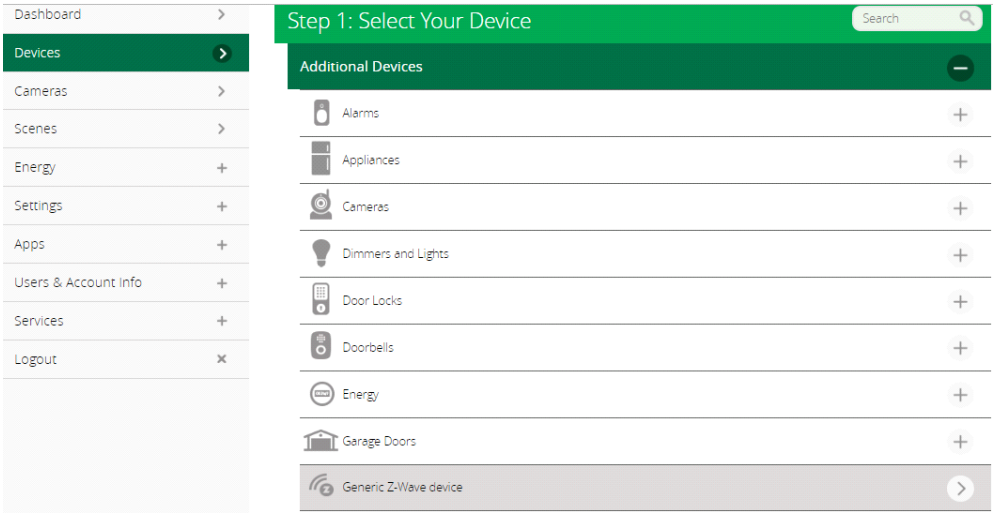


Figure 68 Adding a generic z-wave device in Vera interface

After adding a device, on the main panel we can see a new dimmer device and when moving the slider, the intensity of the flashing of the built-in lamp changes, which means the device is working properly and we can proceed to further adjustment.

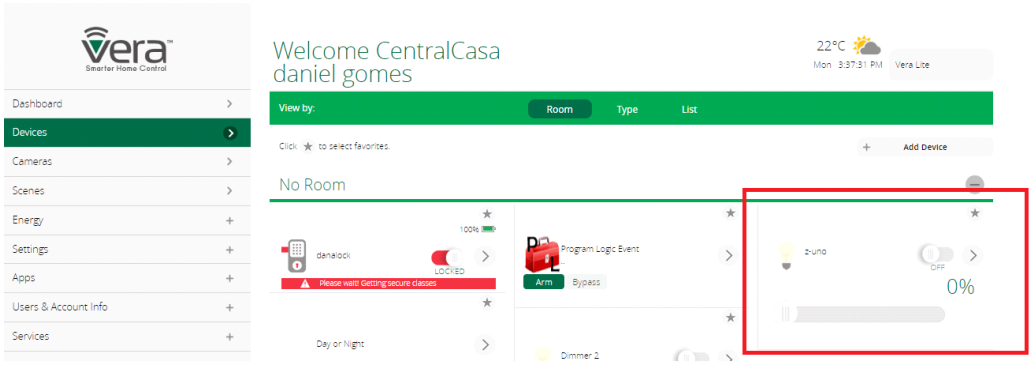


Figure 69 Z-Uno in Vera elements tab

In order to upload our own program, which is called a sketch, to z-uno, we will need to install Arduino IDE, a program for compiling sketches created in the Arduino language.

The important point is that z-uno works with only one version of this program, which is 1.6.5, and it is not the latest version, so you need to find it on the official website of Arduino.

After installing we can run the application.

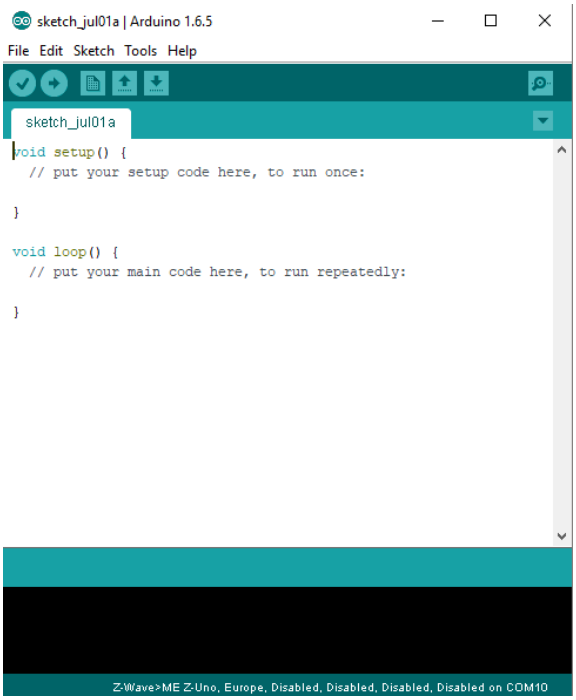


Figure 70 Arduino IDE interface

Then we have to add z-uno package to Arduino IDE. Open File and there Preferences**.** In the field "Additional Boards Manager URLs" put: `http://z-uno.z-wave.me/files/z-uno/package_z-wave.me_index.json` and press OK.

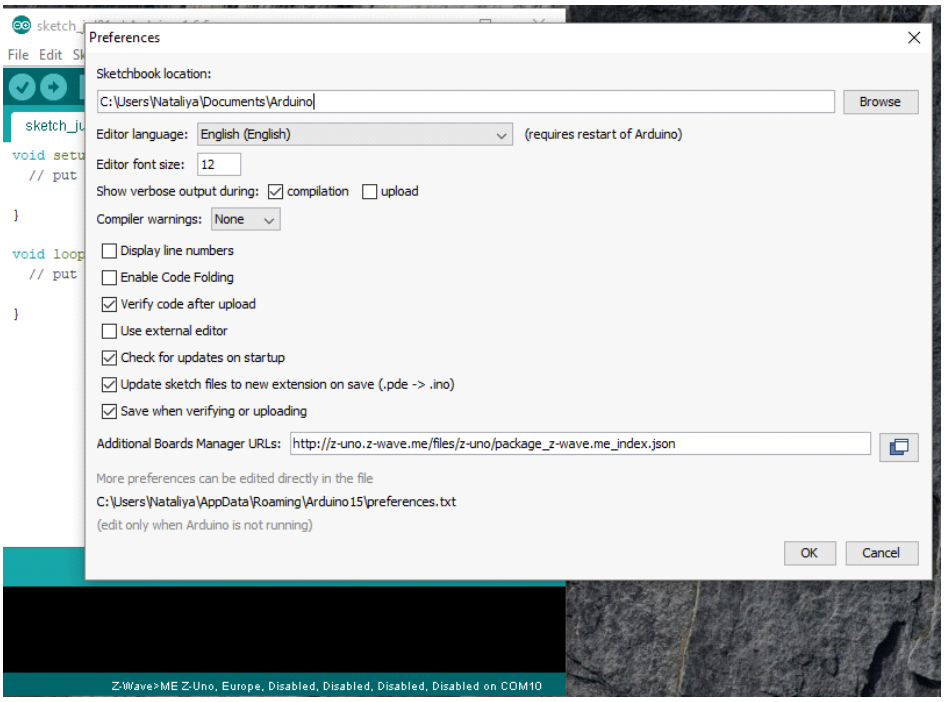


Figure 71 Installing z-wave.me package to Arduino IDE

Then go to Tools → Board → Boards manager. Scroll down, you should see a package "Z-Uno by Z-WAVE>ME". Install it

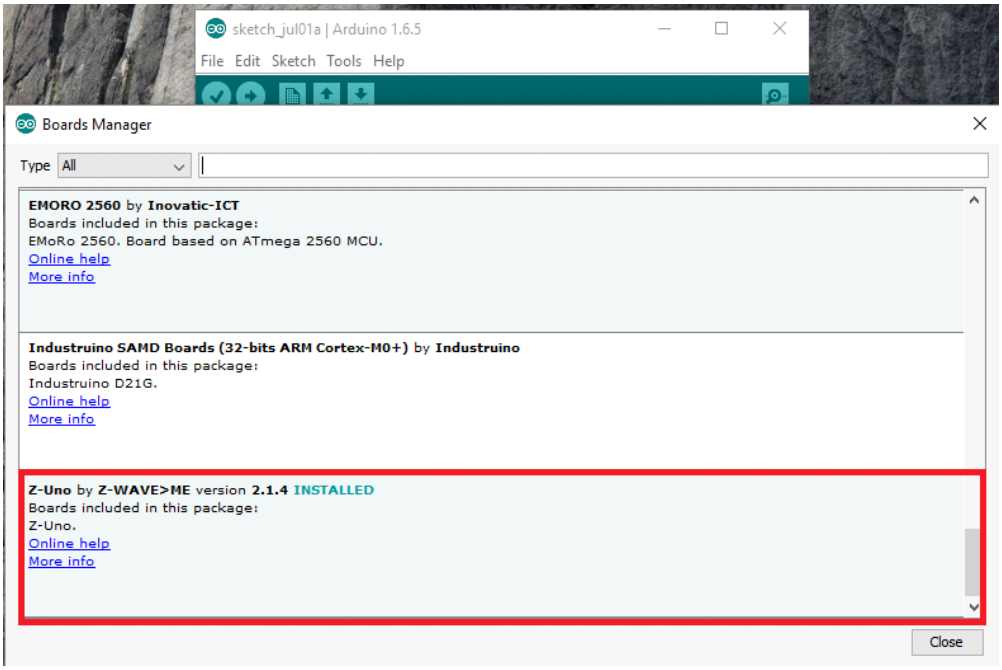


Figure 72 Installing Z-uno package

Now we have to restart Arduino IDE to apply all changes and then select "Z-Wave>ME Z-Uno" in Tools → Board and "Z-Uno Programmer" in Tools → Programmer section.

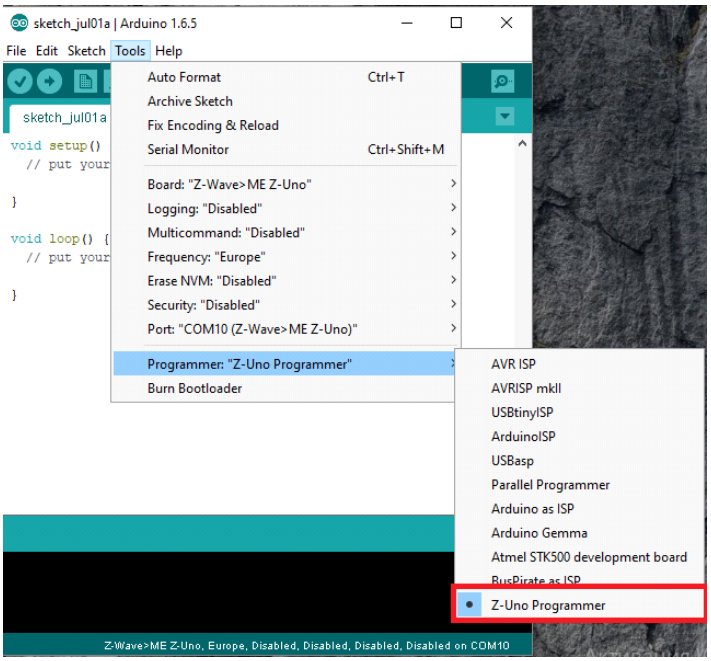


Figure 73 Activating of z-uno in Arduino IDE

The next step is to install driver for z-uno to your pc.

- Download and extract archive with z-uno driver.
- Connect z-uno to your pc with a USB cable.
- Open "Control Panel->Device Manager"
- Select "Unknown device"

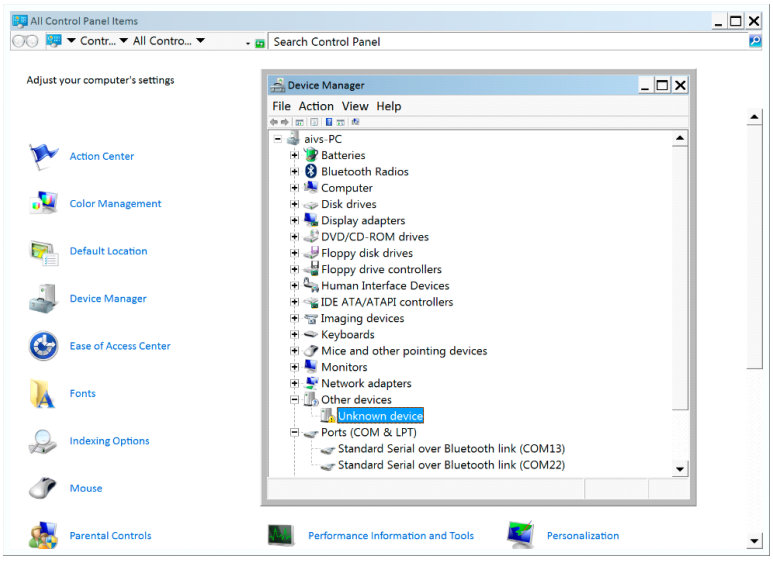


Figure 74 Z-uno in device manager with out drivers

- Press right mouse button on "Unknown device" and select "Update Driver Software ..."

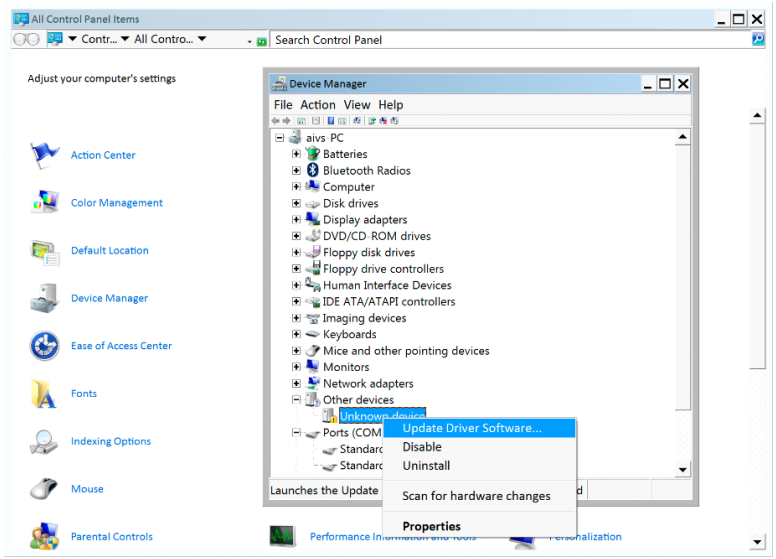


Figure 75 Installing of driver for z-uno

- Press "Browse my computer for driver software"
- Browse folder with Z-Uno driver F
- Driver installed successfully
- New COM port appeared

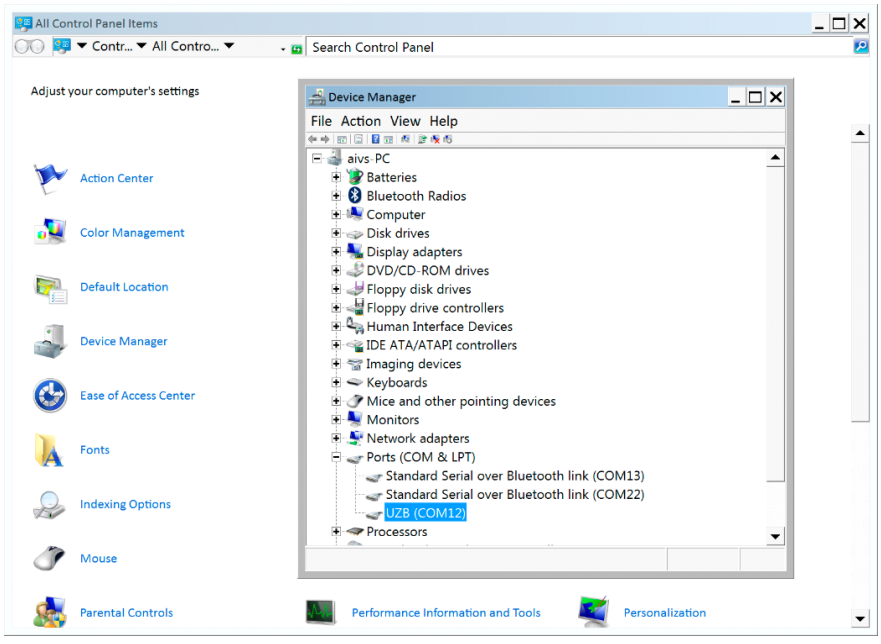


Figure 76 Z-uno in device manager with drivers

Now we can use our Z-Uno in Arduino IDE. We should see "Z-WAVE>ME Z-Uno" in Tools → Port section. Select it

Press Tools → Burn Bootloader. This will update your Z-Uno software to the latest stable version. This can take some time. You can see it's working if Service LED blinks

As you remember, z-wave protocol works on different frequencies in different countries, so we need to choose the frequency we need in Tools → Frequency

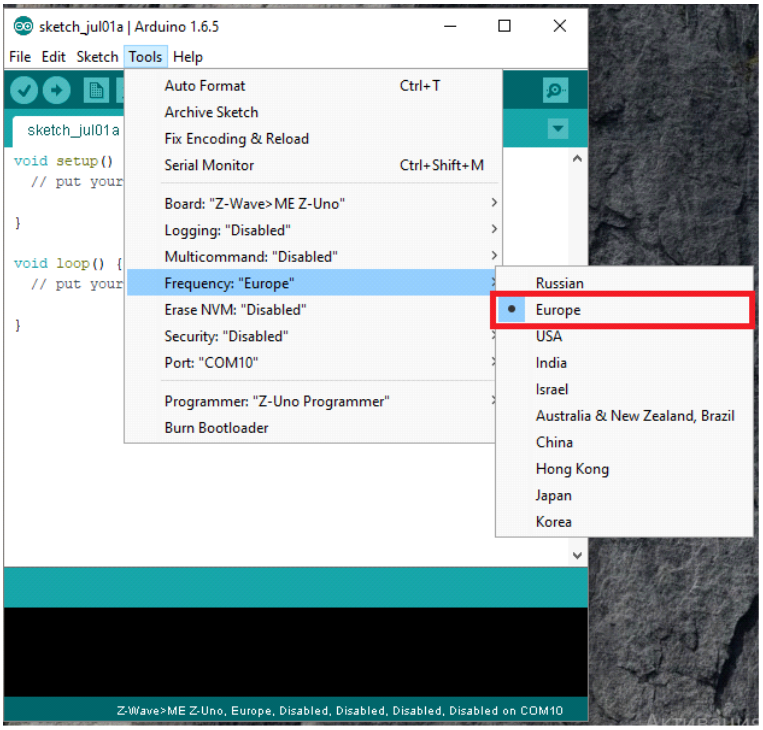


Figure 77 Changing of frequency of z-uno

Now everything is done to start work with z-uno.