Home Automation Course

Professor Olaf Drögehorn

TAVATA

Intelligent Meeting Room

Group 4

Rady, Mina

Ruci, Xhesika

Ida, Fumador

Jalolov Mukhammadjon

Contents

INTRODUCTION

In the context of the home automation seminar, we attempted to introduce a system that applies home automation technologies for a purpose that serves the cause of sustainability.

We identified current inefficiencies in the way medium or large facilities handle their meeting rooms management in a way that causes inefficient energy consumption as well as degraded levels of comfort during meeting times. More importantly, we identified an additional gap in current meeting room management routines which is the lack of security

of meeting rooms. This need is paramount in the case of relatively large facilities where meeting rooms may contain expensive equipment or sensitive/confidential information. There is a need to always ensure that meeting rooms are only open/accessible during scheduled meetings and they are locked otherwise.

SUSTAINABILITY

Social Sustainability

Our system offers a harmonious ecosystem during meeting by ensuring high level of comfort during meeting in an organisation. This comfort improves efficiency and increases concentrations of participants of a meeting. When people are comfortable during the meeting it helps to prevent tension that can lead to rise in temper which can result in unfruitful arguments. Therefore our system will ensure that the people are always happy and in good mood during and after meeting. In effect they will always be motivated to attend the next meeting based on the experience in the previous meeting.

Economic Sustainability: In terms of economic benefits, this can bring significant money savings for customers. Automated control of devices can prolong appliances' lifespan as manual switches lead to metallic contact spares to erode faster. Prevention from access of unauthorised people in a meeting room can also have economic benefits as the room contains valuable data and equipment.

Environmental Sustainability: Our system does not have any potential negative impact on the environment. It can be regarded as a partly closed system with only virtual interfaces to Google calendar and front end displaying. It has definitely connections to energy source.

SYSTEM CONCEPTION AND FEATURES

Our system synchronizes with personal or organizational calendars and adapts room comfort and security measures accordingly.

The system offers the following facilitations for a "meeting's lifecycle":

- **Pre-meeting:** our system synchronizes with existing organizational and personal calendar systems with (iCal format) such as Google Calendar. Consequently, all room automation actions is done according to that calendar. Moreover, as long as room is unoccupied, the system displays the upcoming schedule of reservation during the day.
- **During meeting:** system monitors and displays room comfort settings

(temperature and humidity). Furthermore, it displays the current meeting subject as defined in the organizational calendar entry during reservation.

- **After meeting:** the system waits for motion detection to report no-motion and then it turns powers off devices in the room.
- **Security monitoring**: the system monitors unauthorized access in the room through motion detection and door status. Whenever a meeting room is not occupied and no meeting is held, unexpected access to the room will alert the facility administrator by sending an email.

SYSTEM ARCHITECTURE:

Our system is based on the FHEM software platform for home automation. The platform allows us smooth integration of various sensor readings and controls by streamlining the different schemes on the physical layer.

FHEM is hosted on a Home Manager board which is a Raspberry PI based control system.

We have integrated the following hardware components:

- FS20-PIRI-2 Motion sensor
- FS20DI Dimmer
- FHT80b Thermostat
- HMS100TF Humidity/temperature sensor.
- FS20 TFK door sensor.
- A light bulb to display energy on/off behaviour

In order to achieve integration with organizational calendar/email systems, we created the following FHEM components:

- A calendar virtual device which provides an iCal interface with the Google Calendar of the organization.
- An internal email function using Perl script that contains our automatic email server configuration (SMTP URL, port, username, password).
- Internal scripting triggers (at,notify) that store different room status readings in external files to be accessed by other components external to FHEM.

In order to achieve an LCD display that contains the current meeting room status, we added the following software components:

- Installed Apache server on the Home Manager board with php support.
- A php monitoring page that reads the different stored information about the room which was processed by FHEM and displays it in a friendly format for the public.

The figure below depicts the outline of our system architecture.

Door Sensor   Motion Sensor   Dimmer   Thermostat   Humidity/Temperature
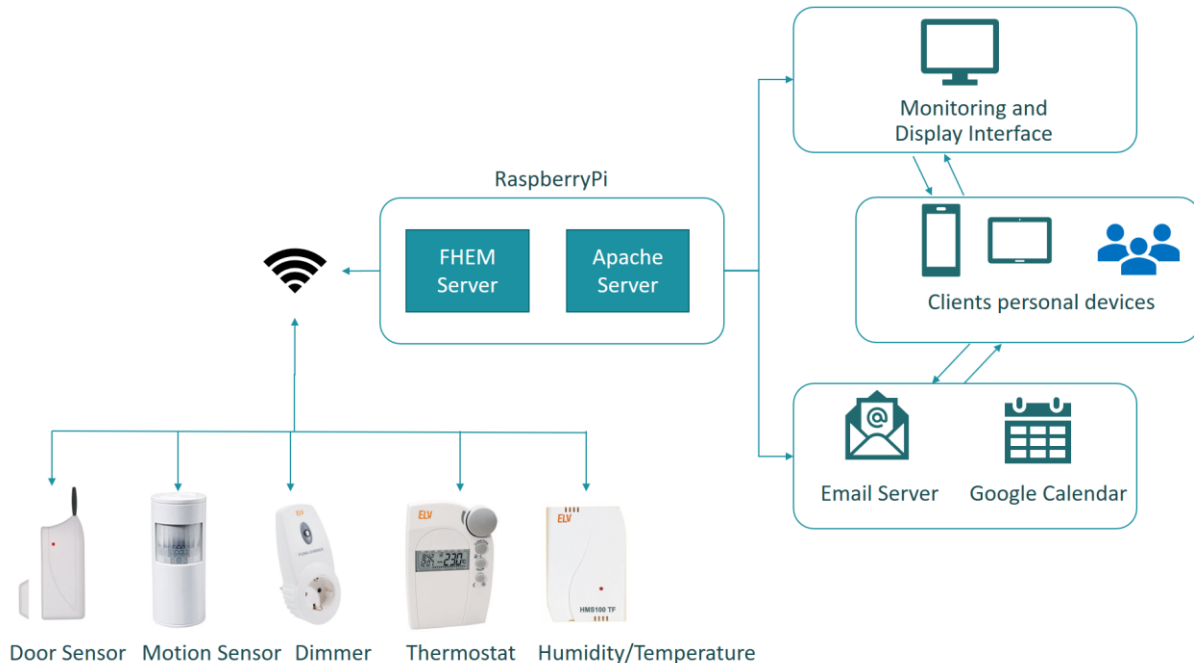
PROJECT IMPLEMENTATION

We have created several notify triggers as well as scheduled routines on the FHEM server using PERL scripts. Those components serve three main functional components of the system which behave as follows:

Notifications:

Our system has the following notify event triggers:
- When a scheduled meeting begins, energy is turned on for the room and the public display monitor displays the current meeting title as assigned in the calendar during reservation. Additionally, the thermostat is sent a command with the desired temperature to it adjusts the heater control actuator according to the temperature readings.
- When a scheduled meeting ends, the system checks the motion detector status to ensure that room users have left the room, only then energy is disconnected from the room.
- If door opens, turn on light automatically.
- If motion detected or door is opened out of schedule hours, send warning e-mail to building administrator.

Display Control:

This is a scheduled routing at executes every given period of time (as configured by administrators) to:
- Display current temperature and humidity of the room
- Check current schedule:
  - If room is a scheduled meeting,display current meeting title as set at the calendar during reservation
  - Otherwise, display upcoming meeting schedule of current day.

Watchdog:

This is also a scheduled routing at executes every given period of time (as configured by administrators) to:
- Check door status and check room reservation status from the calendar
- If there is no scheduled meeting and no motion, turn off lights/devices.
- If there is no scheduled meeting and door is open send warning e-mail to administrator. However, if an e-mail was sent already in a previous check, the system does not send any further email until the security breach is resolved (i.e. door closed). This is meant to avoid spamming administrators with emails about a single incident.

GENERAL IMPRESSIONS ON FHEM:

FHEM provides common interface to interact with different home automation technologies. It offers different open interfaces as well to allow other systems to be interoperable with it. Mainly JSON and XML. It also allows integrating PERL codes and scripts. However, it provides many limitations as all configuration has to be done within the FHEM framework.

Moreover all FHEM forums are in German which made it difficult to get feedback from other developers facing similar problems.

FHEM Limitations:

- FHEM runs on single-threaded Raspberry PI. This makes it quite challenging to trigger multiple notifications for the same event.
- FHEM does not allow typical HTTP hosting for any files outside of its framework, we had to install APACHE server to implement our monitoring interface.
- FHEM does not allow safe modularity or scalability. Since PERL scripts can only be used within its framework, the only way to create truly modular perl components is to use an embedded tool: 99_myUtils.pm which has to be completely compliant to the particular limitations and dependencies of FHEM.

System Pricing

We estimated the price of our system (per one meeting room) by comprising three cost criterias: the hardware cost of the appliances our system uses, the installation cost of the system in the meeting room and the maintenance cost (two-years contract). The hardware cost of our system includes the cost of the sensors, FHEM server and actuator. The installation cost includes deployment of the system in the facility, while the maintenance cost includes the maintenance of the system over a period of two years.

Appliances	Price (Euros)
1. FHEM Server	99.95
2. Door Sensor	39.95
3. Motion Sensor	39.95
4. Temp/Humidity Sensor	39.95
5. Thermostat	39.95
6. Dimmer	59.95
Total Hardware Cost	319.7
Total Installation Cost	225.29
Maintenance Cost	155
Total Price	699.99

SAVINGS CALCULATIONS

For calculating potential contribution to energy savings, as well as money and reduced carbon dioxide footprint, we based our calculations on official and up-to-date data. Although our calculations were assumption-based, the outcomes clearly showed that installing this system in a meeting room environment can bring considerable outcomes. Theoretically, even if turning home appliances off during one unused hour can save energy waste and financial losses. We considered 10-hour timespan for our system.

- Considering the average CO2 emission factor = **0.527 kg / kWh**
- The electricity price **0.220 EUR** per kWh
- Around 12 meetings per month.
- Assuming appliances are left on for 10 hours after meeting

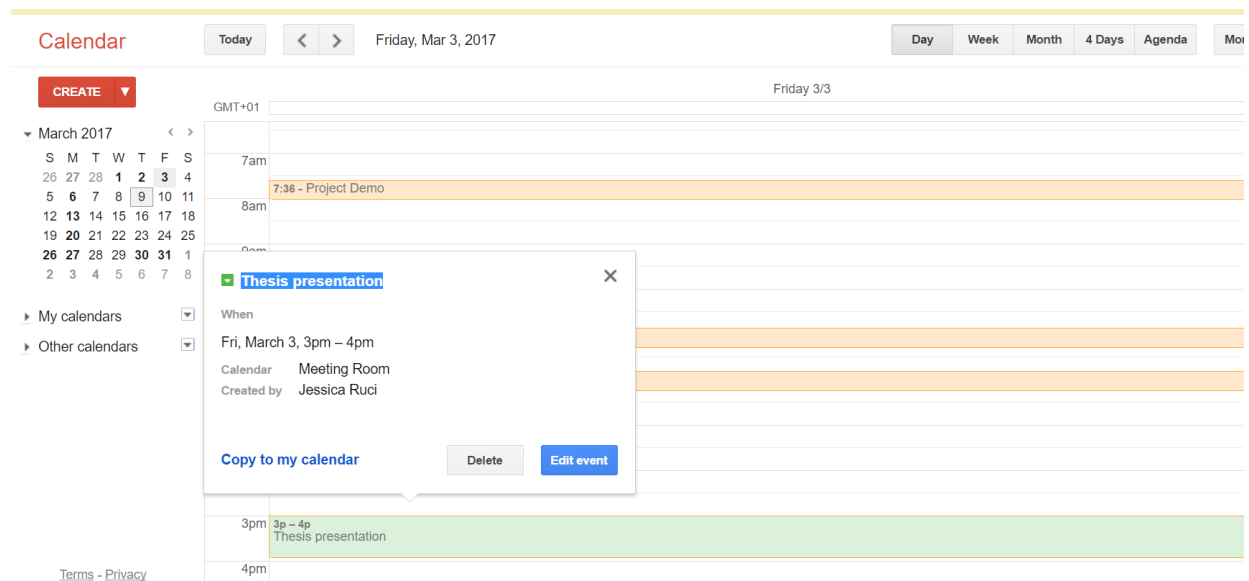| Appliance	Money saving per month	CO2 emissions	Per year	CO_2 Per year
Heaters (2) **2400 W**	63.36 EUR	151.7 kg	760.32 EUR	1820.4 kg
LED lights (20) **300 W**	7.92 EUR	18.9 kg	95.04 EUR	226.8 kg
Air conditioners (2) **2400 W** (**in summer time**)				
Total:	**71.28 EUR**	**170.6 kg**	**855.36 EUR**	**2047.2 kg**

The table above represents our average results. For the period of one year almost 1000 euros can be saved and 2 tons of carbon emissions could be prevented entering into the atmosphere by automating an average meeting room. Accordingly, if the system scales

up to manage facilities of large organizations which may contain at least ten meeting rooms and higher, the total incurred financial savings could easily exceed 10,000 Euros per year.

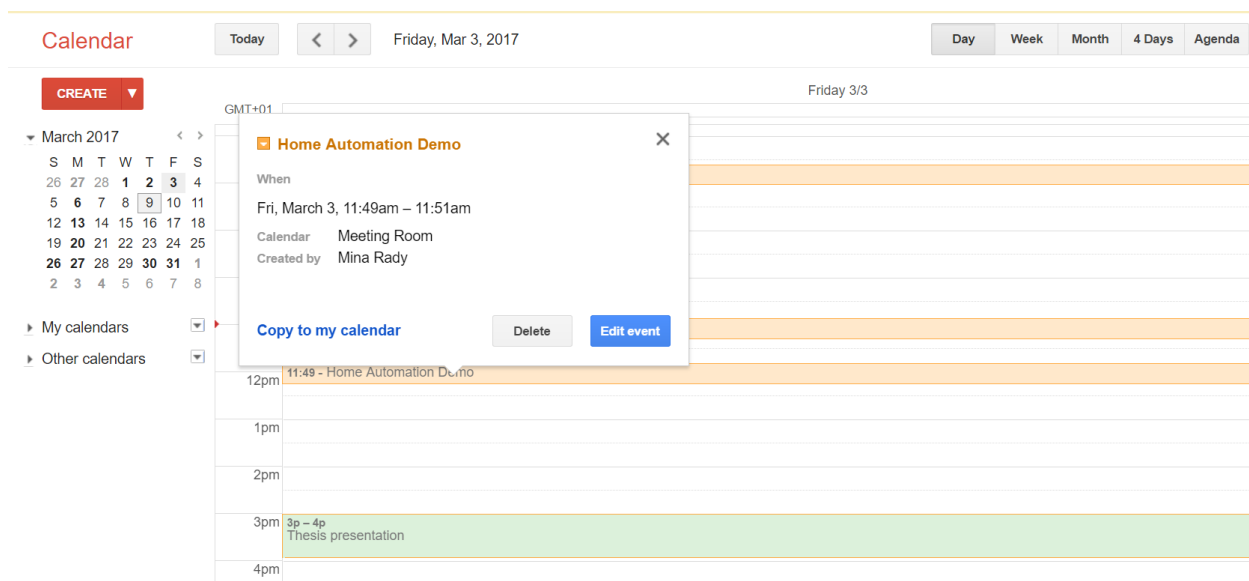## System implementation results:

Google calendar reservation

In this example, we show collaborative reservation by two different users from their accounts to the meeting room at two different times.

Public monitor

The public display monitor shows current room temperature and humidity (during demonstration the

humidity sensor reporting was erroneous). It also shows the upcoming reservation while the room is empty.

Temperature: 26.3
Humidity: 0
Room currently not reserved
Schedule:
03.03.17 16:00 Thesis presentation
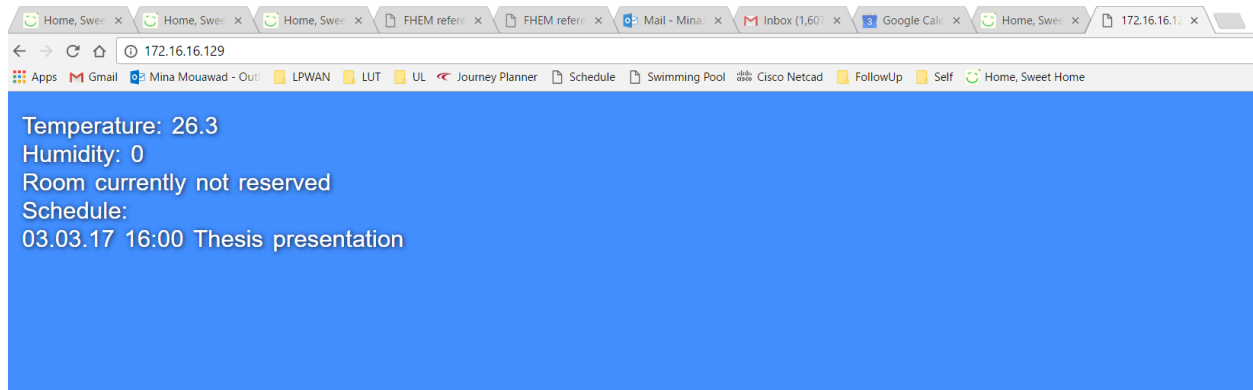
Email warning system

The system successfully reports by email the exact time when an unauthorized presence is detected (in this example, door was detected to be open outside of meeting room reservation schedule).
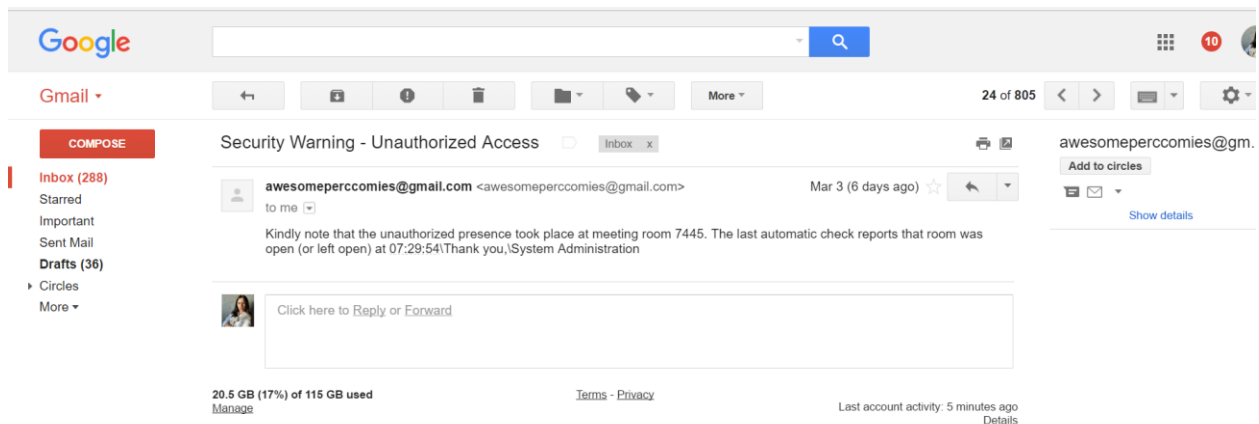

1. Overview of Wireless Protocols

The prevalence of wireless connectivity in industrial control and sensor system is increasing because it extends communication ranges at lower cost than wired alternatives

There exists a bewildering choice of connectivity options for electronics engineers and developers working on products and systems of Internet of Things, IoT. An overview of popular and talked-about technologies driving connectivity in the smart home. Choosing a smart home protocol can be tricky business. Obviously, you want one that will support a large number of devices, as well as one that offers the best possible device interoperability (the ability for devices to talk to each

other). But there are also other factors to consider, such as power consumption, bandwidth and, of course, cost.

INSTEON

Z WAVE

ZIGBEE

WI-FI

BLUETOOTH

THREAD

APPLE HOMEKIT, announced by Apple in 2014, it is a software framework, that enable all smart home devices to connect directly to the IPhone and iPad and be controlled by a dedicated application.

1.1. Z Wave Technology: Asynchronous Wireless Networking Protocol

1.2. What is Z Wave

Originally developed by Danish Startup Zen-Systems and then it was a acquired by a large American corporate, Sigma Designs, in 2008. The Z-Wave protocol is a low bandwidth half duplex protocol designed for reliable wireless communication in a low cost control network. The protocols main purpose is to communicate short control messages in a reliable manner from a control unit to one or more nodes in the network. The protocol is not designed to transfer large amounts of data or to transfer any kind of streaming or timing critical data. The protocol consist of 4 layers, the MAC layer that controls the RF media, the Transfer Layer that controls the transmitting and receiving of frames, the Routing Layer that controls the routing of frames in the network, and finally the application layer controls the payload in the transmitted and received frames.

Z Wave Protocol Stack and Technical Specifications

The z-wave protocol layers main function is to communicate very short messages of few bytes long from a control unit to one or more z-wave nodes. It is a low bandwidth and half duplex protocol to establish reliable wireless communication. z-wave protocol stack need not have to take care of large amount of data as well as any kind of time critical or streaming data.

As shown in the figure Z-wave protocol stack consists of 5 layers: Physical layer, MAC layer, Transport layer, Network layer and Application layer. The security layer is not defined in z-wave open protocol specifications and hence it is implementation specific. Following are the major functions of these protocol layers.

➤Physical layer takes care of modulation and RF channel assignment as well preamble addition at the transmitter and synchronization at the receiver using preamble.

➤MAC layer takes care of HomeID and NodeID, controls the medium between nodes based on collision avoidance algorithm and backoff algorithm.

➤Transport layer takes care of transmission and reception of frames, takes care of retransmission, ACK frame transmission and insertion of checksum.

➤Network layer takes care of frame routing, topology scan and routing table updates.

➤Application layer takes care of control of payloads in the frames received or to be transmitted.
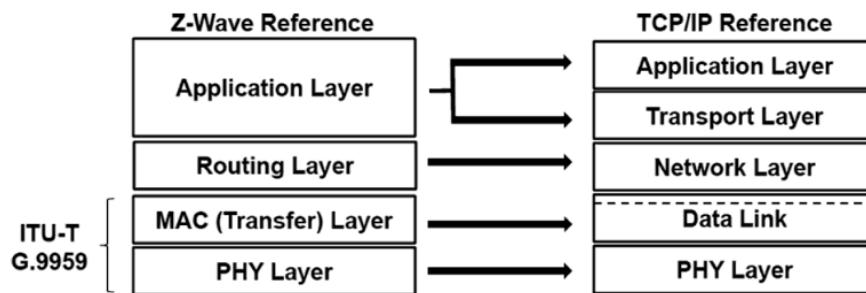
Figure : ITU-T G.9959 Z Wave model mapped to TCP/IP reference model

• Low-latency transmission of small data packets. Supports 9.6 Kb/s, 40 Kb/s up to 100 Kb/s data rates - control and sensor applications

•   Communication distance between two nodes – about 30-40 meters. Up to 4 hops Message

- FSK Modulation with Manchester channel encoding

- Max payload data – 64 bits, 8 bit address blocks

- It uses AES-128 type of encryption to provide secured wireless network- spec. for the encryption of electronic data.

Network and Topology

Z-Wave is a low powered mesh networking technology where each node or device on the network is capable of sending and receiving control commands through walls or floors and use intermediate nodes to route around household obstacles or radio dead spots that might occur. Z-Wave uses a source-routed mesh network topology and has one or more master (primary) controllers that control routing and security. Downfall of this topology is the increased frame length since the route should be included inside of the payload Devices can communicate to another by using intermediate nodes to actively route around and circumvent household obstacles or radio dead spots that might occur. The following example assumes that other devices exist on the network to create the mesh.

Example: A message from node A to node C can be successfully delivered even if the two nodes are not within range, providing that a third node B can communicate with nodes A and C. If the preferred route is unavailable, the message originator will attempt other routes until a path is found to the "C" node.

This allows a Z-Wave network to span much farther than the radio range of a single unit; however, with the use of several hops a delay could occur between the control command and the desired result. (Z-Wave, 2011)
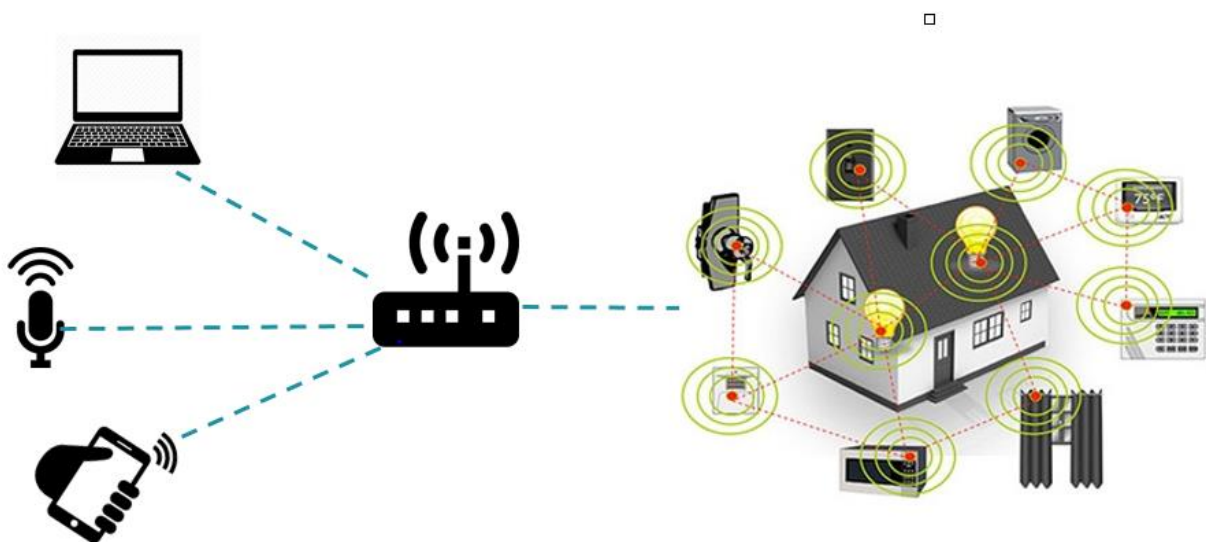
1.2.1.   Types of Network Nodes

The Z-Wave protocol has 2 basic kinds of devices; controlling devices and slave nodes. Controlling devices are the nodes in a network that initiate control commands and sends out the commands to other nodes, and slave nodes are the nodes that reply on and execute the commands. Slave nodes can also forward commands to other nodes, which make it possible for the controller to communicate with nodes out of the direct radio wave reach.

	Neighbours	Route	Possible Functions
Controller	Knows all neighbours	Has access to complete routing table	Can communicate with every device in the network if route exists
Slave	Knows all neighbors	Has no information about routing table	Can only rely to the node which it has received the message from. Hence, cannot send unsolicited messages
Routing Slave	Knows all neighbors	Has partial knowledge of routing table	Can reply to the node which he has received the message from and can send unsolicited messages to a number of predefined nodes he has a route to.

Figure 1: Controller and Slave nodes

3.3 Controllers: A controller is a Z-Wave device that has a full routing table and is therefore able to communicate with all nodes in the Z-Wave network. The functionality available in a controller depends on when it entered the Z-Wave network. In case the controller is used to create a new Z-Wave network it automatically become the primary controller. The primary controller is the "master" controller in the Z-Wave network and there can only be one in each network. Only primary controllers have the capability to include/exclude nodes in the network and therefore always have the latest network topology. Controllers added to the network using the primary controller are called secondary controllers and don't have the capability to include/exclude nodes in the network.

3.3.4 Bridge Controller: A Z-Wave network can optionally have a bridge controller. A bridge controller is an extended static controller, which incorporates extra functionality

that can be used to implement controllers, targeted for bridging between the Z-Wave network and other networks. The bridge controller device stores the information concerning the nodes in the Z-Wave network and in addition it can control up to 128 virtual slave nodes. A virtual slave node is a slave node that corresponds to a node, which resides on a different network type. An example of a bridge controller could be a bridge between an UPnP network and a Z-Wave network to link broadband and narrowband devices together in a home entertainment application.
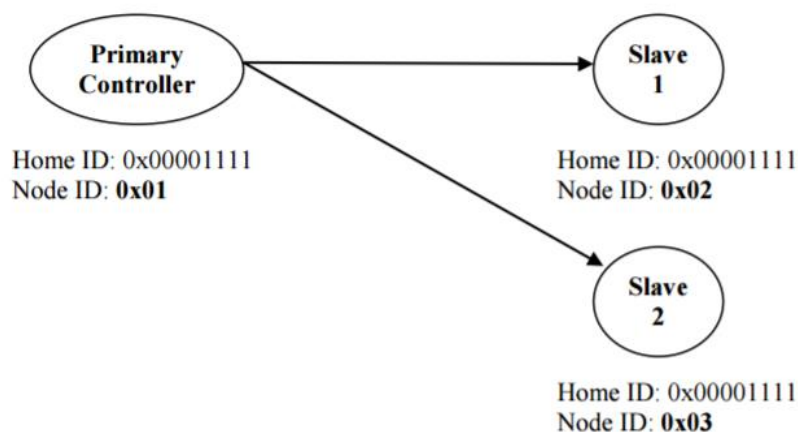
Home ID and Node ID

Home ID: To separate networks from one another the Z-Wave network uses a unique identifier called the Home ID. It refers to the ID that the Primary Controller assigns the node during the inclusion process. • This is a 32-bit code established by the primary controller • Additional controllers will be assigned the same Home ID during the inclusion process • All slave nodes in the network will initially have a Home ID that is set to zero (0) • Once the slave node contains a Home ID it must be excluded before you can assign it to a different network

Node ID: A node is the Z-Wave module itself. A Node ID is the identification number or address that each device is assigned during the inclusion process. The logic works very similar to that of an IP Address. • The primary controller assigns the ID to each node • There are a total of 232 nodes available on each network • Important Note: the Primary Controller is considered part of the network and must be subtracted from the overall node count. Therefore, the total numbers of slave nodes available are 231. In the example below, you can see where the primary controller has a home ID of 16 (0x00001111) and each node has an id of 02 and 03. Note the primary controller will always contain the Node ID of 01.

Primary Controller

Slave 1

Home ID: 0x00001111
Node ID: **0x01**

Home ID: 0x00001111
Node ID: **0x02**

Slave 2

Home ID: 0x00001111
Node ID: **0x03**

	Definition	In the Controller	In the Slave
Home ID	The common identification of a Z-Wave network	Already set as a factory default	No Home ID at factory default
Node ID	The individual identification of a node within a common network	Controller has its own Node ID predefined (typically 1x01)	Assigned by the primary controller

Table 2 : Home ID and Node ID comparison

2. Z WAVE Applications

1. Home control
2. Home security and monitoring
3. Home energy management

3. How Z-Wave works?

4. Z-Wave Protocol Stack and Technical Specifications

5. Comparisons with other technologies on the market. Why use Z-Wave?

One of the most popular of the wireless home automation protocols, Z-Wave runs on the 908.42MHz frequency band which is considered a quiet frequence. Because this is a much lower band than the one used by most household wireless products (2.4 GHz), it is not affected by their interference and "traffic jams." A significant advantage of Z-Wave is its interoperability. All Z-Wave devices talk to all other Z-Wave devices, regardless of type, version or brand. Further, the interoperability is backward- and forward-compatible in the Z-Wave ecosystem; that is, Z-Wave products introduced today will work with Z-Wave products from a decade ago and with products in the future (although possibly with some limits on functionality).

There are currently over 1,200 different Z-Wave -compatible devices on the market, giving consumers access to a wide range of options when automating their home. Known for their low power consumption, Z-Wave devices are designed for ease of set-up and use, a boon to budding

home automation aficionados. As with Insteon, a Z-Wave's mesh network makes all devices double as repeaters