	SEVENTH FRAMEWORK PROGRAMME Information and Communication Technologies
Grant agreement number	FP7-611650
Project acronym	DOREMI
Project Title:	Decrease of cOgnitive decline, malnutrRtion and sedEntariness by elderly empowerment in lifestyle Management and social Inclusion

Deliverable Number:	D3.2
Title of Deliverable:	Wireless Sensor Networks, Integration Middleware and System Configuration
WP related to the Deliverable:	WP3
Dissemination Level: (PU/PP/RE/CO)*:	PU
Nature of the Deliverable: (R/P/D/O):**	R
Contractual Date of Delivery to the CEC:	30/04/2015
Actual Date of Delivery to the CEC:	30/04/2015

WP responsible for the Deliverable:	WP3
Author(s):	Miguel Llorente(MYSPHERA), Luigi Fortunati(CNR-ISTI), Johanes Kropf(AIT)
Contributor(s):	Julián Guirao(MYSPHERA), Jose F. Hernández(MYSPHERA), Francisco Pérez(MYSPHERA), Daniel Vera(MYSPHERA)
Reviewer(s):	Oberdan Parodi(CNR-IFC), Stefano Chessa (UNIPI)

*Dissemination Level:
PU=Public
PP=Restricted to other program participants (including Commission Services)
RE=Restricted to a group specified by the consortium (including Commission Services).
CO=Confidential, only for members of the consortium (including Commission Services).

**Nature of Deliverables:
R=Report
P=Prototype
D=Demonstrator
O=Other

Abstract

This deliverable reports the first release of the DOREMI wireless sensor network (WSN), detailing its architecture, components and integration. This report is delivered together with the preliminary version of the DOREMI WSN, which has limited features but is 100% functional to perform integration and use tests with real users in a Living Lab environment. The document is focused in the description of how each element of the WSN gets data from a specific parcel of the user and puts them available to the rest of the DOREMI system. Features and working details of the software and hardware components developed until the release of this document are detailed. Finally, the development status and the next steps are described in order to put this release in context with the whole works to be performed within WP3 in DOREMI project.

Keywords

Wireless sensor network, device specification, requirements, home automation, middleware, auto configuration, wireless standards, wearable devices, real time location system, balance assessment, occupancy detection, social assessment, Internet of the Things, prototyping, real time location.

ICT / Specific Targeted Research Projects (STReP)

FP7-611650

DELIVERABLE D3.2

Title

D3.2 – Wireless Sensor Networks, Integration Middleware and System Configuration

Authors: Miguel Llorente, Luigi Fortunati, Johanes Kropf

Organization: **MYSPHERA, CNR-ISTI, AIT**

Version: V1.0 / Final version

Date: 30/04/2015

Distribution: **WP3**

Code: **D3.2**

LIST OF BENEFICIARIES

Ben. No.	Beneficiary Name	Short name	Country	Enter Date	Exit Date
1	Consiglio Nazionale delle Ricerche	CNR	IT	1	36
3	TSB Real Time Location Systems SL	MYSPHERA	ES	1	36
4	AIT Austrian Institute of Technolohy Gmbh	AIT	AT	1	36

VERSION HISTORY

Version	Primary Author	Version Description	Date Completed
0.1	MYSPHERA, CNR-ISTI, AIT	ToC Release	09/04/2015
0.4	CNR-ISTI, MYSPHERA	Introduction, DOREMI gateway and smart carpet, Back-end components	21/04/2015
0.6	MYSPHERA, CNR-ISTI	Wristband, RTLS, WSN components, section 6	23/04/2015
0.8	MYSPHERA, AIT, CNR-ISTI	Section 6, Autoconfiguration system and minor changes	24/04/2015
0.9	MYSPHERA	Included changes after feedback from Oberdan Parodi (CNR-IFC) and Stefano Chessa (UNIPI)	29/04/2015
1.0	MYSPHERA, CNR-IFC	Final contribution and changes after feedback from technical coordination (Erina Ferro, CNR-ISTI)	30/04/2015

Table of contents

List of figures

List of Tables

1. TERMS AND DEFINITIONS

Communication Gateway: a computer connected to a sensor network, capable of receiving sensor data from devices and propagate it on a distributed network over the Middleware communication infrastructure.
DOREMI wristband: also called DOREMI bracelet, is the wearable device carried by the user tracking his/her physical activity and hear rate.
DOREMI Gateway: Fixed *Communication Gateway* placed in each DOREMI Test house.
DOREMI User: participant of the DOREMI Trial.
Middleware: software layer that enables secure communication of sensor data between software modules in a distributed computing environment.

WSN: Wireless Sensor Network

KPI: Key Performance Indicator

TCP/IP: Transmission Control Protocol / Internet Protocol

RTLS: Real Time Location System

LED: Light emitting diode

LAN: Local area network

BLE: Bluetooth Low Energy

PIR: Passive infrared

GPS: Global Positioning System

NAT: Network address translation

WLAN: Wireless local area network

CPU: Central processing unit

RAM: Random access memory

LED: Light emitting diode

TLS: Transport layer security
JVM: Java virtual machine

MQTT: Message queue telemetry transport

DB: Database

OS: Operating system
UTC: Coordinated universal time

NTP: Network time protocol

MAC: Media access control

DHCP: Dynamic host configuration protocol

JSR: Java specification request

API: Application programming interface

SQL: Standard query language

REST: Representational state transfer

JSON: JavaScript object notation

GUI: Graphic user interface

2. EXECUTIVE SUMMARY

This document is the second report delivered by work package 3 in DOREMI project. While the first document, coded as D3.1, mainly reports about the system requirements and overall design, this has the purpose to deeply describe each component of the preliminary version of the wireless sensor network and the global integration of all of them. The aim of the preliminary version of the wireless sensor network is to provide a full-integrated system validating each component separately and working all together. This version is provisional: several components miss non-critical features and other do not have their final design (as in the case of the wristband, which will maintain the functionalities decreasing the size and increasing the battery life), but it is very useful for the following reasons:

- It enables the validation of all devices separately and working together,
- it provides an infrastructure to collect data to feed the data analysis modules developed by WP4 members,
- it is usable by real users, providing early feedback and enabling to test the usability before starting the pilot phase,
- it prevents eventual issues on installation, deployment or maintenance processes during the pilot phase.

To describe this preliminary version, the document is divided in chapters. The first chapter (the *Introduction*) gives a global vision of the wireless sensor network (WSN), recapping the main ideas of the deliverable D3.1.

The second chapter, called *System Integration* describes each subsystem, giving the specific requirements, the specifications of the developed component(s), their integration with the rest of the WSN, and finally a first approximation of the installation, configuration and maintenance aspects, which will be more deeply analysed in the third document delivered with the final version, according to the maturity of the system. This chapter is divided in two subsections, a first one for the hardware components, *The Wireless Sensor Network*, and a second one for the software components, *The Back-end components*.

The final chapter is dedicated to analyse the current development status of the work package tasks, linking with the plan for the following six months, which constitute the final period of execution of WP3. The following steps are described including the pending actions with dependencies with other WPs.

3. INTRODUCTION

The DOREMI wireless sensor network (WSN) is a system that aims at retrieving data from users to measure the following key performance indicators, as indicated in *D2.2 – Active Ageing Lifestyle*:

- Physical activity
- Vital parameters
- Balance assessment
- Social interaction

By the correct measurement of these indicators, the whole DOREMI system gets feedback about the performance of the gamified environment, physical exercises and, in general, all the actions performed by the users to improve their quality of life.

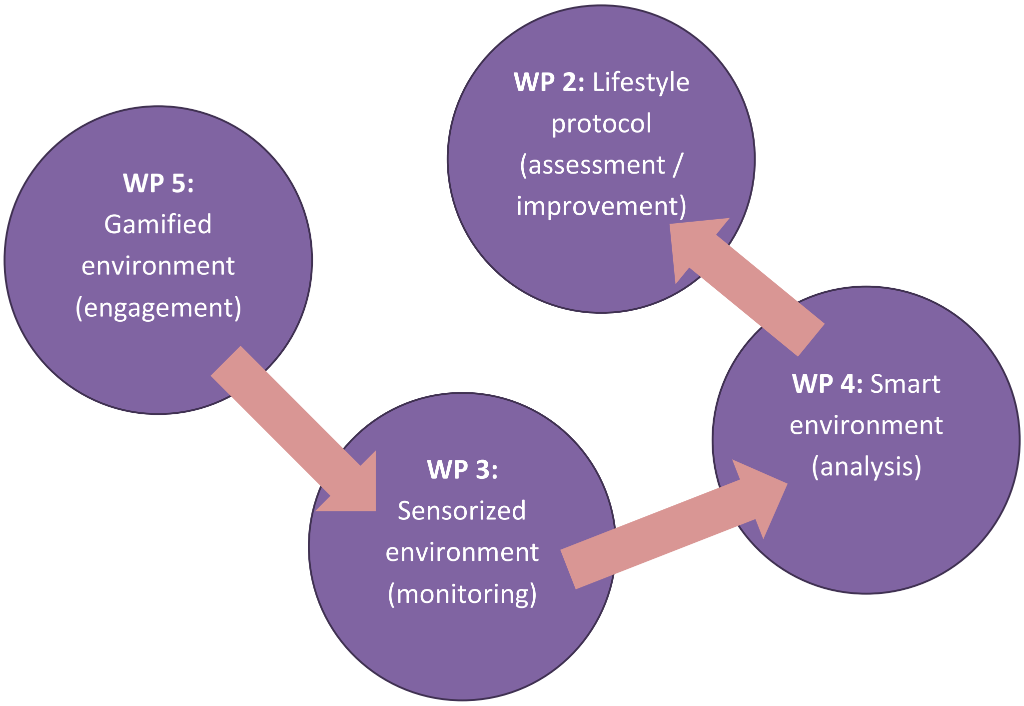


Figure 1 WP3 in the DOREMI context

The WSN is specified and defined in the document *D3.1-Specification of Wireless Sensor Network for LifeStyle Protocol implementation* released in July of 2014 and revised in February 2015. The architecture consists of the combination of several technologies and subsystems to enable the monitoring of the parameters composing the aforementioned key performance indicators (KPIs): step counting, indoor location (at room level), physical movements, interactions with people, outdoor location, heart rate, weight and balance. The relation between sensors and their monitoring purpose is summarized in the following table:

KPI	Input parameter	Sensing device	Data collection process
Physical activity	Step counting	DOREMI wristband	Automatic
	Indoor location	DOREMI wristband	Automatic

	Physical movements	DOREMI wristband	Automatic
Social interaction	Interactions with other people in home	Presence sensor	Automatic
		Door sensor	Automatic
	Interactions with DOREMI users while outside	Smartphone GPS	Automatic
	Routines	Indoor Localization system	Automatic
Vital parameters	Heart rate	DOREMI wristband	Automatic
	Weight	Smart carpet	Requires some user actions
Balance assessment	Daily balance evaluation	Smart carpet	Requires some user actions

Table 1 KPIs and WSN components

User monitoring must take place during every activity, either at home or when outdoor. According to this, two different scenarios have been described and the network has been consequently designed to follow the users during their daily life acquiring suitable information without obtrusion, since most of the monitoring is transparent to the user. The overall WSN diagram is as follows:

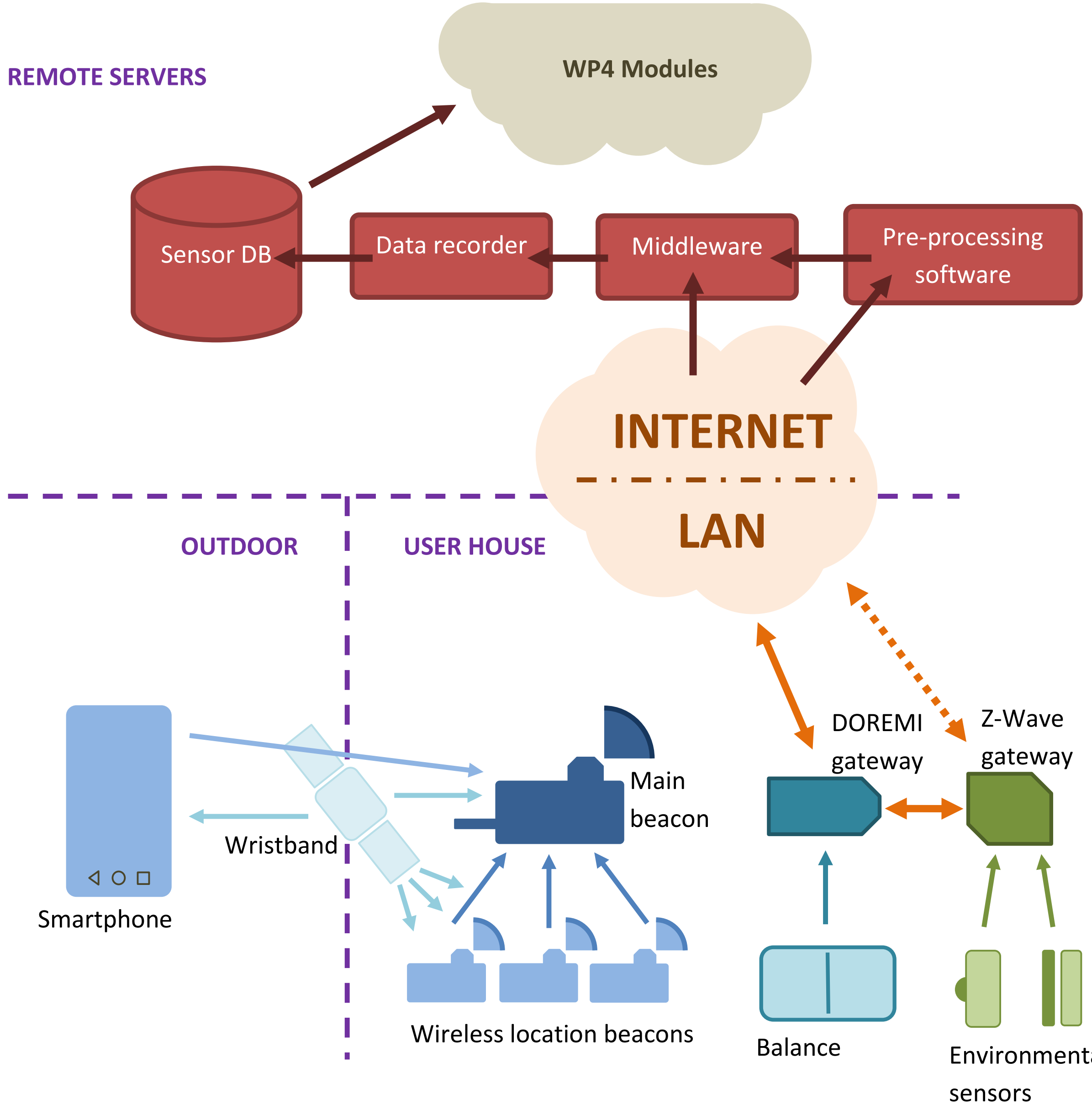


Figure 2 Architecture of DOREMI WSN

The communications in the WSN use three different technologies: Bluetooth Low Energy (blue spectrum arrows), Z-Wave (green arrows) and TCP/IP (orange arrows), as represented in Figure 2. BLE and Z-Wave are wireless technologies making easier the installation and deployment in the pilot phase. According to the Figure, wiring is only needed for the TCP/IP communications, so that three IP accesses are needed for the correct working of the DOREMI WSN: DOREMI Gateway, Z-Wave Gateway and RTLS main beacon.

All data generated in the WSN are sent to the middleware (in the case of the data coming from the main beacon, they are pre-processed in a remote module), an end-to-end communication system that enables secure transmission and retention of sensors data. The latter is stored in the sensor database through the data recorder module. That database is accessible by the WP4 software modules so the algorithms developed for DOREMI can be applied over the stored data.

4. SYSTEM INTEGRATION

This chapter provides a detailed description of the components involved in the data collection process, reporting the envisaged solution for the pilot, the interplay between each component and user interaction details.

4.1. The Wireless Sensor Network

The Wireless Sensor Network is a set of devices with the collective aim of retrieving and storing in a secure manner, on remote premises, sensed data related to the DOREMI user. The data collected by the Wireless Sensor Network pertains to:

- Weight and balance (Smart Carpet)
- Indoor activity (PIR and Door Contact sensors)
- Heartbeat and body movement (Wristband)
- Indoor Location (Indoor Location System and Wristband)
- Outdoor Location (Smartphone GPS)

To collect these data, the Wireless Sensor Network leverages both the devices installed in the apartment of the DOREMI user (*Indoor WSN*), such as the environmental sensors, the networking and the computing facilities, and the personal devices that are mainly used outdoor (*Outdoor WSN*), such as the wearable sensors and the mobile phones.

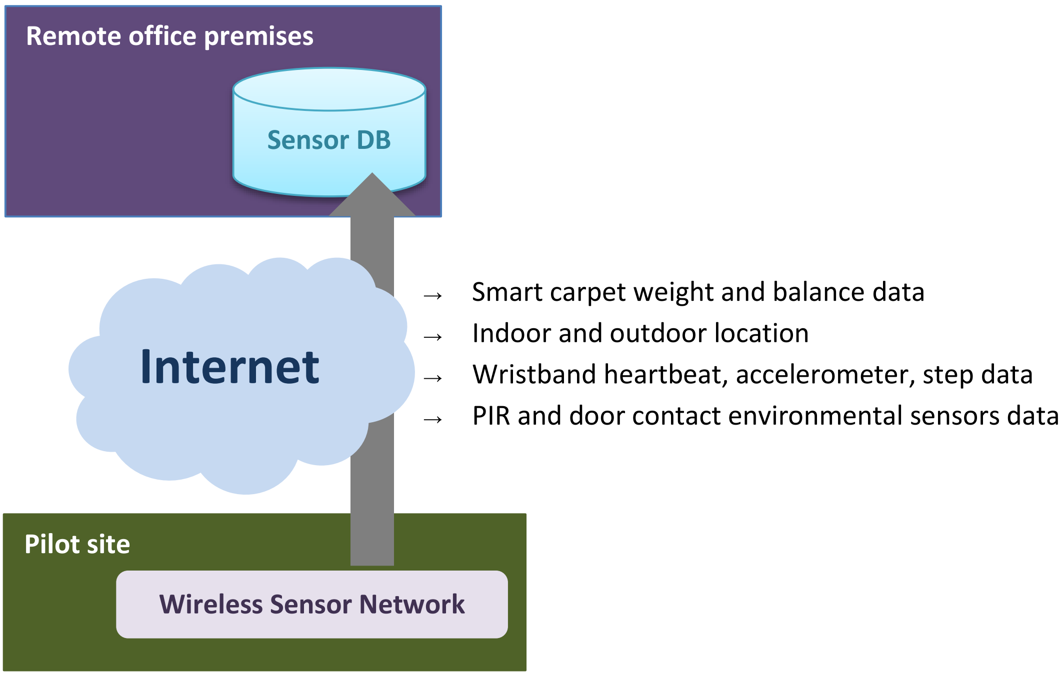


Figure 3 - Wireless Sensor Network scenario

In the following chapters, a description of each WSN subsystem is given, along with a specification on how the subsystem is integrated in the WSN. The last chapter provides a description of the required infrastructure facilities that enable the WSN use case.

4.1.1. Networking

Data collected by the WSN has to be stored on remote premises were computing facilities provide enough resources for nightly processing. The Indoor WSN must therefore provide networking facilities that allows devices to connect to remote servers through an Internet connection.

The networking solution has the following requirements:

- Must provide both cabled and wireless LAN capabilities
- Cabled LAN should support at least 3 hosts (DOREMI gateway, Z-Wave gateway and Main beacon).
- The Internet connection must provide at least 1Mbit Upload bandwidth
- Must allow configuration of IP addresses
- Must provide remote configuration capability
- Must support NAT
- Must support WI-FI bridging

In order to fulfil these requirements, a wireless access point/router, will be installed in each DOREMI pilot apartment. If the router has public IP addressing, this solution will allow the technical staff to control LAN and WLAN configuration from remote, when needed.

If the pilot site already provides an Internet connection through a cabled or wireless network then the solution already in place will be evaluated by the technical staff deploying the WSN. Otherwise, a subscription to a local Internet Service provider has to be considered for each apartment. Wireless Internet connectivity is discouraged as the wireless media is usually prone to interference a poor performance.

4.1.2. DOREMI Gateway

The DOREMI Gateway is the central computing platform of the indoor WSN with the role of collecting data from the environmental sensors of the Indoor WSN and safely storing them on a central database for later processing.

Sensors are not usually capable of storing generated data using remote services and generally they provide access to real-time data through a heterogeneous set of communication protocols (IP based protocols, Zigbee, Z-Wave, Bluetooth, etc.). In sensor network deployments, the gateway node generally stands as the central hub that interacts with available sensor devices using heterogeneous protocols, exposing their functionalities and data through a consolidated interface.

In DOREMI, the gateway is a computing platform that supports the extraction of scalar-typed data generated by the Z-Wave environmental sensors and the Smart Carpet, managing the persistence of collected time series data on a remote sensor database by using secure protocols.

As the DOREMI solution tries to be not too invasive in the home environment, the device must not take too much space and it must be affordable in terms of costs.

The DOREMI Gateway also needs an internet access in order to communicate with remote services over the Internet.

Specifications

The device selected as the DOREMI Gateway is the Raspberry PI 2 Model B. It is a credit card sized single board computer developed by the Raspberry Pi Foundation with the following general features:

- CPU: 900MHz quad-core ARM Cortex-A7
- Memory: 1GB RAM
- Storage: MicroSDHC slot (up to 64GB storage)
- OS: Support for Linux or Windows 10
- Networking : Ethernet networking enabled
- 4 USB ports
- Video output: HDMI
- Cost: 35 USD

Figure 4 – The Raspberry Pi 2

The OS selected for the deployment in DOREMI is the Raspbian OS, a Debian-based Linux distribution tailored for the ARM processor of the Raspberry Pi device.

On top of the OS, the device will run a software stack that allows sensor integration developers to run and maintain the software layer that retrieves data from the WSN sensors and publishes them on a remote sensor database. The software stack includes the Java runtime environment (Oracle JVM v7 or later) and the Karaf OSGi container (v3.0.x or later); it is detailed in chapter 4.1.2.1.

CNR-ISTI tested the performance of the Raspberry Pi version 1 and 2 models with the selected runtime environment by running the smart carpet integration layer under development. The RPi2 hardware was capable to support the maximum sample rate provided by the smart carpet device (~100 sample/sec) without showing significant sample loss or jitter. Tests done on older RPI models showed that the software stack could run even on those platforms with low performance losses. The choice on the Raspberry PI 2 is motivated by a larger memory pool (1GB instead of 512MB), a processor that grants Java server code optimization (only available in ARMv7), and a higher clock rate (900MHz instead of 700MHz).

The device will need hardware support that allows accessing to the Smart Carpet Bluetooth wireless interface and the Z-Wave sensor data IP service. For the Bluetooth case an USB Bluetooth dongle that support the Bluetooth protocol version 2 can be attached to the device.

About the connection to the local IP network, if the device is located near the indoor networking device, then it can be connected to the Access Point with an Ethernet cable. If the device has to be placed far from the access point or in another room, a wireless 802.11 b/g/n USB dongle is required. The first solution is preferred though as the wireless connectivity is generally prone to interference in environments where multiple wireless networks are setup.

Since most USB WiFi chipsets present compatibility issues with Linux OSes, several USB dongles were tested for compatibility and ease of installation. The following dongles were selected as eligible for the pilot:

- Edimax EW-711Un Wireless 802.1b/g/n nano USB adapter
- Hama Nano Bluetooth 3.0 USB Adapter Class 1 (code 049238)
- Inateck Bluetooth 4.0 USB Adapter Class 2 (BTA-CSR4B5)

Bluetooth network interfaces mostly differ in terms of supported Bluetooth protocol version and communication range (Class feature). Since the balance board falls in the Class 2 category (10 meters range), there is no motivation for adopting a Class 1 dongle (100 meters range).

The Raspberry Pi requires a set of other accessories to work. Power must be provided to the device with a certified 5V/2A power supply with USB cable and the required world adapter. Regarding the storage, an SD card is needed to host the OS. Tests done by CNR-ISTI proved that the 8 GB NOOBS SD card, suggested by the Raspberry Pi community, shows better read/write performances than other high-end SD cards; therefore, using that model is strongly encouraged. The Raspberry Pi will also need an opaque solid case that hides the board LEDs, such as the black case provided by OneNineDesign.

Figure 5 - OneNineDesign black RPi case, Edimax WiFi USB dongle, RPi Official power adapter

4.1.2.1. Middleware and sensor integration

The middleware is generally defined as a software layer that enables communication between software modules in a distributed computing environment. In DOREMI the Middleware is as set of distributed software components that enable secure data collection of sensor data generated by heterogeneous sensing devices, thus enabling the following use cases:

- Publication of sensor data and sensor device metadata;
- Discovery of available sensors and access to data streams.

The Middleware proposed for the DOREMI project is called Sensor Weaver and it is an evolution of the Middleware involved in the GiraffPlus European project [1].

The Middleware software components comprise the following elements:

- *Communication infrastructure*: Computing facilities and software that enable communication between nodes in the distributed networking environment.
- *Client libraries*: software libraries that simplify integration, hiding aspects such as configuration, low-level protocol communication, data format.

Considering that in the DOREMI pilot the application servers (like the DB node) and pilot sites will be in distinct locations, the Middleware will provide a secure distributed communication framework for the transmission of sensor data over the Internet.

The communication infrastructure of Sensor Weaver is based on the Message Queuing Telemetry Transport protocol (MQTT), a lightweight message transport specifically tailored for machine-to-machine communications and Internet-of-Things scenarios. The protocol is based on the publish/subscribe messaging pattern. In a message-oriented architecture, the communications between nodes are mediated by a message broker. This central node allows clients to publish data or to subscribe for receiving messages on a set of topics. In DOREMI, the MQTT message broker is provided in a virtual machine hosted on a server in office premises, deployed by AIT. Details on the subsystem are provided in chapter 4.2.2.

The DOREMI Gateway runs a set of software components that enable the integration of environmental sensors on top of the message-based communication infrastructure. They are:

- Raspbian OS
- Karaf OSGi container based on Apache Felix OSGi runtime
- Sensor Weaver Middleware OSGi bundles

Raspbian is a Debian-based operating system optimized for the Raspberry Pi hardware. It provides drivers support for most wireless technologies and hardware, like Bluetooth or WiFi.

On top of the OS, the DOREMI Gateway will run the Karaf OSGi container. Karaf is a small OSGi-based runtime that provides a lightweight container onto which various software components can be deployed, according to the Open Service Gateway Initiative philosophy. OSGi is a set of specifications that allows building modular applications based on components, also known as *bundles*. The specification defines how an OSGi framework implementation should deal with bundles lifecycle and manage the interactions between those. A bundle is a Java jar archive that contains additional information inside its MANIFEST.MF, which allows the OSGi framework to identify the bundle and to enable interactions with other bundle components.

In addition to support either the Apache Felix or Equinox OSGi framework implementation, Karaf provides a set of additional features that ease the maintenance of the node and the software bundles, such as remote console (Web or SSH) and JMX monitoring. These features are reported in detail in section 4.1.2.2.

Sensor Weaver offers a Java client library as a set of OSGi bundles for sensor integration. These components builds an abstraction layer, based on the concept of d*ata feed*, on top of the communication layer provided by MQTT. A *DataFeed is* a generic entity, characterized by some properties, which periodically produces data with a predefined format. Middleware clients can announce the presence of a new data feed by providing a set of properties and a message format with a key-value description model. When a new data feed is announced by a client, the middleware returns a data feed client object that can be used for publishing data according to the defined message format for announcing the unavailability of the data feed when the data source goes offline.

Other clients that consume data can discover data feeds matching a given set of properties and receive their data.

The middleware library features caching and secure communication channel support with the broker. In-memory caching allows delaying the dispatch of data messages in case the network connectivity is unavailable. The support for secure communication is based on end-to-end TLS connections with the MQTT broker.

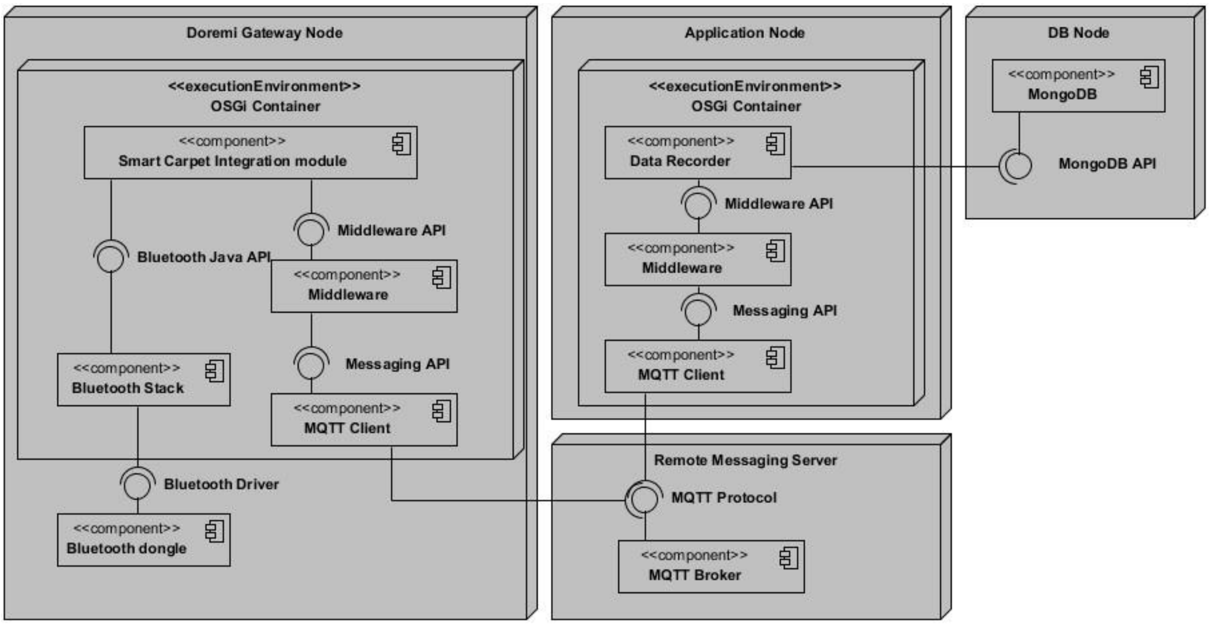


Figure 6 - Smart Carpet integration deployment diagram

Interactions between components work as depicted in the example deployment diagram above, in Figure 6 showing the computing nodes and component involved in the integration of the Smart Carpet device. The following example stands as a general description of the suggested integration approach for the indoor sensors: z-wave environmental sensors and smart carpet.

General sensor integration approach

In the smart carpet integration, the Wii Balance Board (which is a Bluetooth device) generates data through a Bluetooth communication channel. The Raspberry Pi hosts a Bluetooth dongle, its driver and the java software stack (Java Bluetooth device abstraction bundles) that enables data access in the OSGi container. The integration layer for the device is represented by the *Smart Carpet Integration module,* a bundle that performs the following actions:

1. When the bundle is started by the OSGI container, it searches for an available Balance Board.
2. When a device is found it creates a new Data Feed on the Middleware, attaching to its properties a unique device id, and identifier of the sensor type, and the device-specific message format.
3. The data feed is then announced on the Middleware, which releases a custom data feed client object.
4. The integration layer starts collecting data from the device. For each sample received, it sends a message using the data feed client object previously released.
5. If the device goes offline, the integration layer announces the unavailability of the related data feed by calling the specific method on the data feed client object. At this point, the integration module should continuously search for the Smart Carpet to come online. When one is found the workflow goes back to step 2.

The characterization of each device will include the minimal set of data that allows identifying the device itself, without including personal information regarding the user. This approach provides complete anonymity of data produced by the gateways and travelling over the Internet. Additional metadata regarding the device is safely hosted in the Auto Configuration system DB, which couples sensor identifiers with additional information such as sensor type, placement and related info.

Each data message sent by each data feed must include a timestamp along with the sample data. The format of choice for the timestamp is the UNIX timestamp (UNIX epoch) in milliseconds, referring to UTC time. On a related note, DOREMI gateways will be provided with NTP (Network Time Protocol) clients and forced to periodically update the clock to keep synchronized with the other systems.

Parameter	Format/Unit	Description
timestamp	32-bit integer	Time in UNIX Epoch (with milliseconds)

Table 2 - Mandatory timestamp parameter for each datafeed event message

On the other side of the wire, a virtual machine (*Application Node*) is running an OSGi container with Middleware client libraries and the *Data Recorder* bundle. The latter component leverages the middleware by listening for any new data feed available. When a new data feed is found, the component starts recording received data on the Sensor database, detailed in chapter 4.2.3. The sensor DB and the data recorder node run on the same secure local network. Sensor Waver documentation for developers and administration is available online[1].

Configuration for middleware and node certificate preparation

All the communications between the Middleware client nodes and the message broker are encrypted with TLS, hence all the messages exchanged and travelling over the Internet are cyphered. Each DOREMI gateway is supplied with a node certificate, used to authenticate the MQTT client on the Message Broker and to obtain the authorization required for publishing and receiving messages on a set of topics. The authorization mechanism enables the creation of data visibility scopes on Sensor Weaver and its use is completely transparent to the sensor integration developer. Therefore, a node can be configured to access a specific scope where travels data related to a single user or a set of them. The certificate released to every node regulates the access to a set of scopes.

Along with the configuration needed in order to handle the client certificate, the middleware libraries will need:

- Public IP address and port of the MQTT message broker.
- List of enabled scopes as a list of strings. In DOREMI, a single data scope can be used.
- Integration layer configuration: this could include information that identifies the source of data as, for example, the MAC address of the device to monitor.

Configuration of the MQTT message broker is provided in chapter 4.2.2.

4.1.2.2. Remote maintenance

Monitoring and maintaining the DOREMI Gateway during the pilot tests requires an open access to the DOREMI Gateway from remote hosts over the Internet. The networking solution sees each DOREMI gateway node connected to a private network installed in the premises of the pilot site apartment. Each node will therefore have a private IP address that makes the host not reachable from other hosts across the Internet.

The solution devised for DOREMI consists in the deployment of a Virtual Private network between the DOREMI Gateways and a remote VPN server installed in office premises (AIT). Each DOREMI Gateway will be prepared with certificates and networking configuration prior deployment that forces the setup of the VPN tunnel at system start up, providing a secondary VPN unique IP address to the host. The IP addressing of each host is established before the pilot at node preparation, by linking the certificate released to each gateway to a unique IP address and node hostname. Since the only requirement on the gateway for establishing the VPN tunnel is to have connectivity to the remote VPN server, the DOREMI gateway VPN network will be tested before deployment in pilot sites.

[1] http://wnlab.isti.cnr.it/sensorweaver

A dedicated DNS server will also be configured allowing clients to easily identify gateway IP addresses, removing the need of manually mapping IP addresses to gateway hosts.

Technical staff clients will be able to access the DOREMI Gateway hosts by first connecting their hosts to the VPN using the same approach followed by gateways.

The solution offers several advantages over other approaches, like SSH tunnelling or local network router NAT configuration:

- There is no need to configure TCP/UDP port-based rules for IP service access. In case a new service is setup on the gateway, no additional configuration is required in order to allow remote access to the related port.
- There is no need to configure additional devices or hosts. The Network Address Translation (NAT) solution, for example, would have required an ad-hoc configuration of each home router, posing burden on the deployment phase. Configuration of the VPN takes place during DOREMI Gateway preparation phase, before deployment in pilot sites.
- IP addressing of hosts is decided at pilot preparation time through certificate preparation and VPN server configuration.
- DOREMI gateways are accessed regardless of the IP addressing solution adopted in the home LAN (DHCP based or static configuration).
- All communications between hosts in the VPN are cyphered.

Once connectivity at the IP level is established from remote, a number of services will enable remote monitoring and maintenance.

A SSH server will provide access at the operating system level, allowing overcoming problems that could arise with the execution of the Karaf container.

Apache Karaf bundles a set of services that ease maintenance tasks on the OSGI container.

- *SSH console*: the service allows remote clients to access the Karaf console with the SSH protocol. The tool allows controlling the execution of software bundles and changing the node configuration at runtime.
- *Karaf Web console*: the service offers similar features to the SSH console service through a web interface.
- *Karaf Debugging service*: when the service is active, remote hosts can debug the execution of program code from remote, simplifying runtime errors troubleshooting.
- *Java Management eXtension support (JMX)*: allows fine-grained monitoring and management of resources allocated to the Java Virtual Machine.
- *Email error reporting*: the Karaf logging configuration can be configured to send error log reports to a set of email addresses.

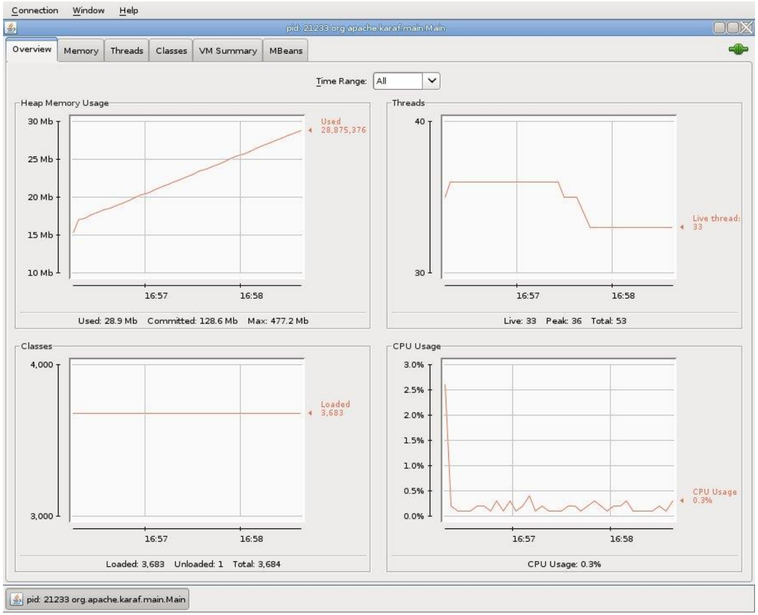


Figure 7 - An overview of a JMX enabled Karaf container provided by the JConsole client application

Even if the DOREMI Gateway will provide enough services to let the technical staff maintain the integration components from remote, local technical support is required, due to actions that cannot be performed remotely, such as physical resets or button combinations in devices.

4.1.3. Smart carpet

Description and requirements

The smart carpet is the home device that enables the day-by-day weight measurement and the periodic balance assessment.

The device provides the following main features:

- It provides enough data to assess the weight of the individual with negligible error and assesses its balance conditions;
- wireless: unwired communication channel would allow free placement of the device inside the user apartment;
- low cost: the device must not be expensive;
- ease to use and in a safe way.

Specifications

The smart carpet solution identified for the DOREMI project is the Wii Balance Board gaming device produced by Nintendo. It is a force platform that evaluates forces on the z-axis at the four corner ends of its surface by using four strain gauge sensors placed inside the 4 scale feet.

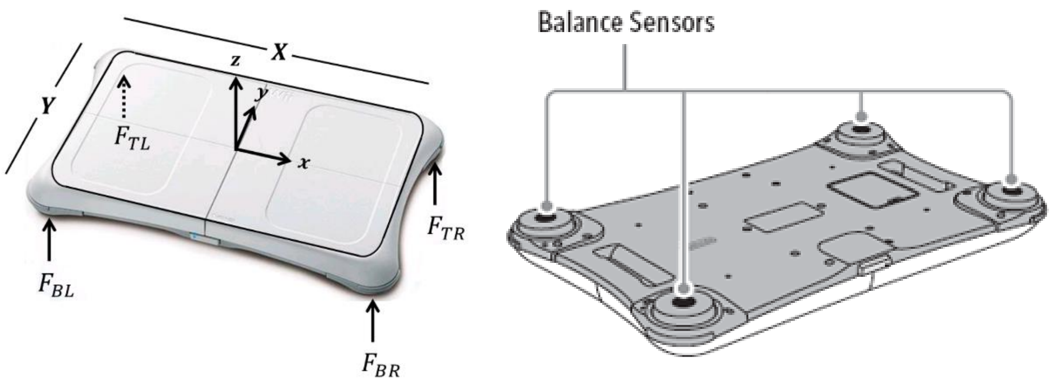


Figure 8 - Wii Balance Board measured forces and sensors

Several scientific works show that, even if the Wii Balance Board cannot be considered as laboratory-grade equipment for balance analysis, it provides good reliability at a very low cost [2][3][4].

The Balance Board allows to access data wirelessly via a Bluetooth 2.0 communication channel. The Bluetooth receiver needs to stand at 10 meters max in line of sight from the device to communicate without data loss. At first, several communication tests were performed using drivers developed by the Java community. The drivers were later adapted to a new, more modular and less resource consuming version, compatible with the OSGi platform. The nominal sample rate of 100 sample/sec degrades when the device is too far from the Bluetooth receiver. It is therefore advisable to place the device at a distance of maximum 4 meters in line of sight from the receiver to achieve the maximum sample rate. Each sample produced by the balance board includes weight (in kilograms) at each of the four corners (real number), status of the button (Boolean for pressed, not pressed), and status of the light (Boolean for on, off).

The modified Java driver also allows to control the status of the front-button light and to check the battery status. Another limitation regards the weight and precision of measurement. The balance board can produce meaningful measurements in the range 0-150 kg.

```
topRightWeight: 23.364783653846153,
topLeftWeight: 13.067469879518072,
bottomRightWeight: 37.923854848304586,
bottomLeftWeight: 13.874850299401198,
buttonPressed: false
```

Figure 9 - Sample data from Wii Balance Board

Usage and limitations

The device is activated by pressing the front button or the button placed on the bottom plane of the device, on the battery compartment. Once a button is pressed, the device requires a few seconds to establish the communication with the Bluetooth receiver. As in DOREMI we need to provide a device that is readily available to use, the device activation constraint is removed by keeping the device always active during the pilot. This behaviour is made possible by using a rechargeable battery always connected to a power socket with a power cable.

The board will be safely placed in the living room or in a place where the device does not hinder the mobility of the user, near the DOREMI Gateway and a power socket and with the frontal light/button facing the user. The user is expected to step on the board from a standing position, to execute some activities and then to make a step back. The protocol regarding the set of activities to perform while on top is related to items of Berg Balance test and is out of the scope of this document.

Maintenance

The device will be activated during the deployment phase by technical staff installing the WSN and left active during the whole duration of the trial. Technical staff from remote will monitor the status of devices by accessing the provided DOREMI Gateway maintenance tools.

If the device gets disconnected from the power source or an unforeseen event causes the device to turn off, the technical staff in pilot site, or the DOREMI user, is required to turn the device on again by pressing the power button on the back plane of the device, located on the battery compartment. A fixed light notification will provide visual feedback to the user, informing him that a communication channel has been established with the Gateway receiver. Technical staff from remote will monitor the procedure from their workstation by accessing the DOREMI gateway node logs and will restart individual application bundles if needed.

Integration

CNR-IFC has already provided 20 official Wii Balance Board and backup units. The additional hardware required in order to integrate the device:

- A Bluetooth 2.0 USB dongle
- A rechargeable battery extension with continuous power supply support over USB

The USB Bluetooth dongle will be selected between the two devices listed in chapter 4.1.2.

CNR-ISTI tested the rechargeable battery kit “Snakebyte Wii Balance Board Play & Charge kit”. The device was installed on a balance board and kept active for one month without experiencing data loss or disconnections with the Bluetooth receiver. An USB power adapter with two USB ports could be used for powering both the Raspberry Pi and the balance board since the devices needs to be placed near each other.

Figure 10 - Snakebyte play&charge rechargeable battery and the Skross UBS world power adapter

The DOREMI Gateway will provide integration software components for the smart carpet. According to the general integration approach described in chapter 4.1.2.1, several OSGi bundles will be provided in order to access device data and to publish them on the Middleware communication bus. The set of bundles comprises: an OSGi-compliant JSR-82 implementation based on Bluecove, Wii Balance Board Java Driver, Balance Board OSGi Device abstraction layer, and middleware integration layer.

The first module provides a low-level Java interface enabling Bluetooth communications for the L2CAP - Logical Link Control and Adaptation Protocol (i.e. the data access protocol implemented by the Wii Balance Board). The Java Driver and the OSGi device abstraction bundles abstract on the low-level protocol, providing a high-level model and simpler interface. The smart carpet middleware integration layer bundle exposes data generated by the device on the Sensor Weaver Middleware using the Data Feed device abstraction.

Each balance board will therefore result in a Data Feed being published by the integration layer on the Middleware. The Data Feed will report, as a unique data feed property, an ID based on the MAC address of the balance board Bluetooth interface. Each balance board will be paired with a DOREMI Gateway before the pilot deployment in order to avoid the possibility that a Gateway could connect to the wrong device placed in a near apartment. The following Table recaps the properties and parameters published with the DataFeed.

Property	**Value**	**Description**
sourceId	MAC of Wristband	Bluetooth interface MAC address
sensorType	**wiiboard**	Sensor type identifier

Parameter	**Format/Unit**	**Description**
timestamp	32-bit integer	Time in UNIX Epoch (with milliseconds)
topLeftWeight	32-bit floating point / Kg	Weight measured at the top left corner
topRightWeight	32-bit floating point / Kg	Weight measured at the top right corner
bottomRightWeight	32-bit floating point / Kg	Weight measured at the bottom right corner
bottomLeftWeight	32-bit floating point / Kg	Weight measured at the bottom left corner

4.1.4. Environmental sensors

Description and requirements

The environmental sensors are intended to get suitable data from the daily life of the user to evaluate the social interactions in an unobtrusive, user-unaware way. These sensors are called environmental since they are installed in the rooms of the user's house and do not require any user intervention, avoiding interferences in his daily life.

There are two types of environmental sensors: presence detectors, based on passive infrared technology (PIR) and door detectors, based on magnetic contacts. Measurement from these two kinds of devices are combined to assess, among others, the number of social interactions at home and an approximation to the number of people interacting. The selection of the sensors, as the rest of subsystems has been performed considering requirements from the lifestyle protocol (WP2), the smart environment (WP4) and the WSN (WP3).

Despite of specification in the deliverable D3.1, based on the indications of the first version of D4.1, two kind of devices have been discarded for the pilot phase due to different reasons. The microphone will not be used for privacy concerns raised by the pilot responsible partners. Nevertheless, data collection experiences have been done using microphones, so the impact of these devices will be studied, independently of the decision of not using them in the pilots. About the pressure sensors or people counters, although these devices have been identified as good candidate to obtain precise information about the presence of hosts at the user home, a strong requirement of the DOREMI project is the affordability for the final user. This recommends avoiding the use of these devices as these are noticeably more expensive than the rest of the components of the network and their inclusion in an eventual product would drastically affect the price, producing a 50% cost increase in the best case.

The limitations imposed to the specifications of the selected environmental sensors have been studied in collaboration with WP4 members, and the performed changes have been agreed with the partners. These changes have been reflected in D4.1 and in D3.1 revised versions, where specifications have been let as they were in the first version, in order to report the work performed during the first period of the project, but they have been updated in the section 14 System *configuration and deployment in pilot sites.*

The devices used in DOREMI are commercial products of the Z-Wave catalogue. This technology has been selected due to its maturity and wide availability of devices, accomplishing the requirements for the project (API to access the full data, low energy consumption, wireless, and ease of deployment). The full set of Z-Wave devices is available in the official website of the Z-Wave Alliance[2].

The Z-Wave technology requires an additional element to set up and manage the network and to retrieve all the data generated by each Z-Wave sensor. In DOREMI, this element is also responsible to

---

[2] http://products.z-wavealliance.org/regions/1/categories/8/products

offer data access to a middleware integration layer running on the DOREMI Gateway as explained in chapter 4.1.2.1. A commercial Z-Wave gateway has been selected to perform this task, once checked for compatibility with the selected sensors.

The PIR sensor has the following requirements:

- Must detect movements of a person in a room of 4m. X 4m.
- Dimensions must not be longer than 10 cm. x 10 cm. x 5 cm.
- Must not emit sounds in normal operation.
- Must support ceiling and wall mounting.
- Must publish its online/offline status as well as the battery life.
- Battery must last at least 3 months and must be easily replaceable.

While the requirements for the door sensor are the following:

- Dimensions must not be longer than 10 cm. x 10 cm. x 5 cm.
- Must not emit sounds in normal operation.
- Must support door and doorjamb mounting.
- Must publish its online/offline status as well as the battery life.
- Battery must last at least 3 months and must be easily replaceable.
- Must detect a *close* status when the distance between the active and passive parts is up to 2 cm.

Finally, the Z-wave gateway has the following requirements:

- Proven full interaction with the selected sensors.
- Dimensions below 15cm x 15cm x 10 cm.
- Do not emit sounds or noise in normal operation.

Specifications

The selected presence sensor is Everspring SP814, which has the following specifications:

- Operating frequency: 868.62 MHz
- Battery: 1.5V AA size x 3. Battery life depends on the activations of the sensor. Estimated 6 months.
- Operating Range : Up to 30 meters line of sight (indoor)
- Warm Up Time: About 2 minutes
- PIR Detection Coverage :
    - Wall-Mounted: Up to 10m x 110° (at 1.8m mounting height & 25°C)
    - Ceiling-Mounted: Up to 5m x 360° (at 2.8m mounting height & 25°C)
- Dimensions: 85 x 85 x 45 mm

Figure 11 Everspring SP814

The presence sensor works as a motion detector. It is normally in a 'no detection' state. When motion is detected, it sends a 'detection' signal to the gateway. A 'no detection' message is automatically sent by the presence detector after a configurable time after which the sensor has not detected any movements, and it does not offers information about the actual absence of presence. The data format is given by the Z-Wave gateway configuration and can be easily modified. This has been fixed as follows:

- 'detection' message: Motion <room_code> <time_stamp>

Where `room_code` is one of the following: `{bedroom, bathroom, livingroom, inside, outside}` and `timestamp` is in format *yyyy-MM-dd hh:MM:ss*.

An example of 'detection' message is the following:

```
Motion outside 2015-04-13 03:14:15
```

The door sensor selected is Everspring HSM02, with the following specifications:

- Operating frequency: 868.62 MHz
- Battery: CR2450 3.0V 620mAh Lithium Battery
- Range: Minimum 30 m line of sight
- Operation gap up to 30mm
- Dimension : 31 x 70 x 11.5mm

Figure 12 Everspring HSM02

The door sensor sends an asynchronous message to the Z-Wave gateway whenever the sensor is opened or closed. In this case, both messages are relevant, since a door can be normally opened or closed and the data are generated exclusively when the state of the door changes. As in the case of the presence sensor, the data format is given by the Z-Wave gateway and it is customizable. In DOREMI, the format of the messages is the following:

- When the door opens: ContactDoorOpen <room_code> <timestamp>
- When the door closes: ContactDoorClose <room_code> <timestamp>

Where `room_code` is one of the following: `{entrance, bathroom}` and `timestamp` is in format *yyyy-MM-dd hh:MM:ss*.

The following are two examples of messages generated by this sensor:

```
ContactDoorOpen bathroom 2015-03-20 15:07:48
```

```
ContactDoorClose entrance 2015-02-14 23:32:12
```

The Z-Wave gateway chosen to manage the subsystem is the Vera Lite, a cost-effective, programmable and configurable device with a graphic user interface (GUI) that enables remote management for the sensors.

Further information about these two devices is available in their datasheet, attached as appendixes.

Usage and limitations

This subsystem does not interact directly with the user, being this transparency to the user one of its most important characteristics. Consequently, there is no expected usage or learning phase to the user.

Maintenance

The environmental sensors are wireless devices fed by standard batteries. Since the power consumption directly depends on the number of activations of each device, the battery life of each sensor can differ. While it is not a normal situation, the battery of some devices could drain during the pilots, due to an unexpected high frequency of activations or because of batteries manufacturing issues. A battery drop can be easily detected by remote technical staff by accessing the Z-Wave gateway. If a device shows a critical battery level or has completely run out of battery, the technical staff of the pilots will be contacted in order to replace it. This operation is very straightforward since the batteries are standard and the devices offer easy access to battery compartments (see Annexes).

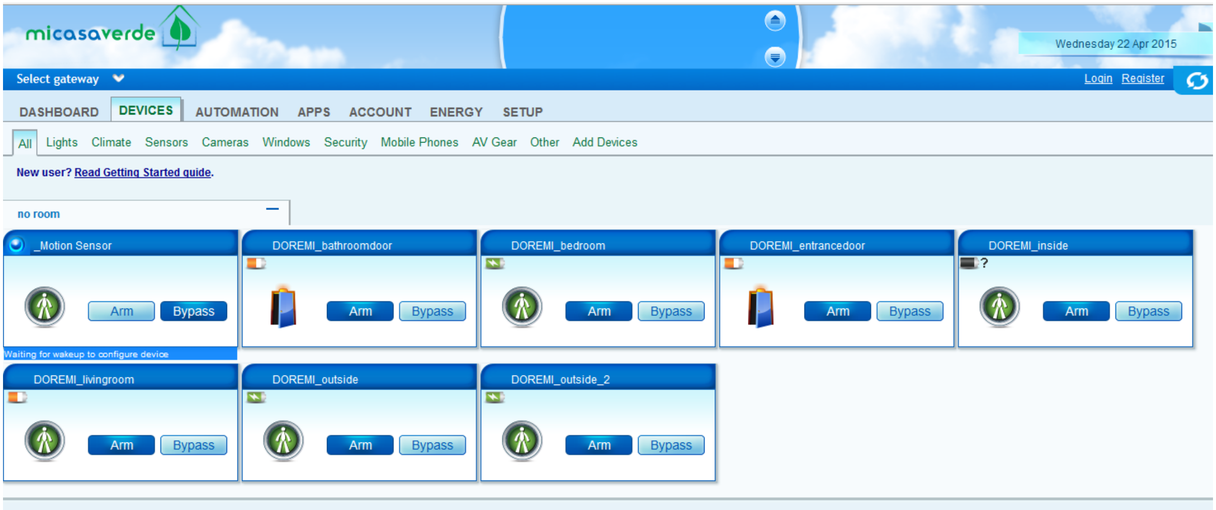


Figure 13 Vera Lite User Interface for sensors management

Information from the network is stored in the memory of the Z-Wave gateway, so it does not need to be reconfigured after an occasional power line drop. Additionally, Vera Lite has a backup power feed mechanism with standard AA batteries, so it could keep working after some time even without electricity in the house. In any case, this feature is useless in the context of the DOREMI project, since several WSN elements, such as the DOREMI gateway or the location beacons cannot work during a power line drop, thus in this case the full system stops working.

Integration

The Z-Wave gateway is the element that retrieves and integrate all the environmental sensors. It is integrated with the rest of the DOREMI WSN by a software component in the DOREMI gateway. The Z-Wave gateway is physically connected to the LAN of the user's house with an Ethernet wire to home router. This way all data collected can be automatically sent through a socket to a middleware client running in the DOREMI gateway. This software module publishes the data in the middleware where are managed as any other.

Hence, each Z-Wave gateway will have a Data Feed representation publishing the data of the environmental sensors in the following format:

Property	**Value**	**Description**
sourceId	ID of sensor	ID given to the Z-Wave sensor
sensorType	**PIR**	Sensor type identifier

Parameter	**Format/Unit**	**Description**
Value	Boolean	True when activated
timestamp	32-bit integer	Time in UNIX Epoch
RoomId	String	Id of the room

Property	Value	Description
sourceId	ID of sensor	ID given to the Z-Wave sensor
sensorType	**DoorContact**	Sensor type identifier

Parameter	Format/Unit	Description
Value	Boolean	True when closed, False when opened
timestamp	32-bit integer	Time in UNIX Epoch
RoomId	String	Id of the room

4.1.5. Smartphone

Description and requirements

In this project, a smartphone will be used as a mobile gateway for the data generated by the wristband. The use of a smartphone presents many advantages and some drawbacks. Some of the advantages of using mobile phones in DOREMI are:

- Availability of outdoor location, suitable for social assessment, routines determination and even wandering detection.
- Simpler circuitry in the wristband that allows more hardware resources available for the core functionalities.
- The use of a smartphone is generally considered a good practice for the elderly living alone because of security reasons, since it provides an instantaneous way to communicate to the elder and, if needed, to assist on emergency.

It also introduces the following pitfalls:

- It forces the user to take the device each time he gets out home. The user has to remember about this;
- It needs a recharge cycle each night, adding an additional routine in the user´s life.

For DOREMI, it has been considered that the advantages are more important than the problems than the use of smartphone as a gateway introduces.

The smartphone will be compliant with the wireless communication technology of the DOREMI wristband (BLE), and it will manage the communication in outdoor-indoor and indoor-outdoor transitions, as well as it will be responsible for collecting and storing the data provided by the wristband and GPS position.

Specifically, the smartphone will manage the following:

- The smartphone detects the DOREMI wristband and reads data that the BLE device transmit. When the smartphone detects data emitted by the DOREMI wristband, it is saved in the phone memory.

- The user leaves home: the user leaves home wearing the DOREMI wristband and carrying the smartphone. As the devices loses connectivity with the home wireless network, the smartphone starts GPS location collection.
- The user goes back home: The smartphone establishes an Internet connection with the WIFI of the user's house and the smartphone app stops the data collection and starts the data transmission to the back-end.

All these features are performed by a custom application developed for the DOREMI project. This application runs in background in the phone, avoiding any user interaction that could add complexity to the DOREMI protocol or reduce the overall usability.

Specifications

The selected device must meet the technical specifications needed to receive the DOREMI wristband data and GPS position. These specifications are:

- Bluetooth 4.0 Low Energy integrated.
- GPS location service.
- OS Android 4.3 (API Level 18) or higher.

Furthermore, this device meets other criteria that have been considered, such as:

- Affordable device, ideally under 100 euro.
- Computational capability for process and storage the data
- Clear user interface.

The selected device is the Motorola Moto E, which meets all the specifications and is offered at a price of 79 Euro on Amazon.

Figure 14 Motorola Moto E

The specifications are:

- OS Android 4.4.4
- Bluetooth v4.0 LE
- Qualcomm® Snapdragon™ 410 processor with 1.2 GHz quad-core CPU.
- 1GB RAM
- 8GB internal memory
- Battery (2390 mAh) Mixed usage up to 24 hours.
- MicroUSB charger

The Motorola Moto E is sold without charger, thus one must be purchased for each unit.

4.1.5.1. Integration: Android application

This section describes the features of the Android application.

Application initialization

As mentioned above, the Android application has been developed to minimise the user interaction.

For this reason, when the application has been installed properly it runs in background. This is thanks to a component of Android, the “Broadcast Receiver” that receives and to responds the events generated by the system. In this application the module "Broadcast Receiver" is responsible for starting up the service that read the BLE frame. When the service starts it first checks the Bluetooth interface status. If the Bluetooth interface is switched off, the app enables it automatically.

Scenario Detection

The smartphone application is able to detect when the user is not in range of MYSPHERA RTLS and it also detects when the Smartphone arrives home.

The method chosen to detect the scenario, where the device is located, is through the Wi-Fi connection status. Therefore, when the app starts it checks if the phone is connected to the home Wi-Fi network. If the Wi-Fi is connected, the application decides that the smartphone is within the range of MYSPHERA RTLS (indoor mode). Else, if the Wi-Fi is disconnected or the device is not connected to the home network, the application decides that the smartphone is outside the range of MYSPHERA RTLS (outdoor mode).

Moreover, the application is able to detect the changes in the Wi-Fi connection status. This can be made because other "broadcast receiver" is the manager to receive the changes of the Wi-Fi connection and it decides the location of the device.

Read data from the wristband

The Android app is ready to receive and process Bluetooth Low Energy frames that is sent by the DOREMI wristband. As it has been mentioned above, the service that runs in the background is responsible for reading BLE frames, which are emitted from devices that it has around it. When the application detects that the frames are emitted by the DOREMI wristband, it processes the frames and stores the data in database.

The smartphone database is SQLite, a simple and lightweight relational database. The DB data model is depicted in the following figure.

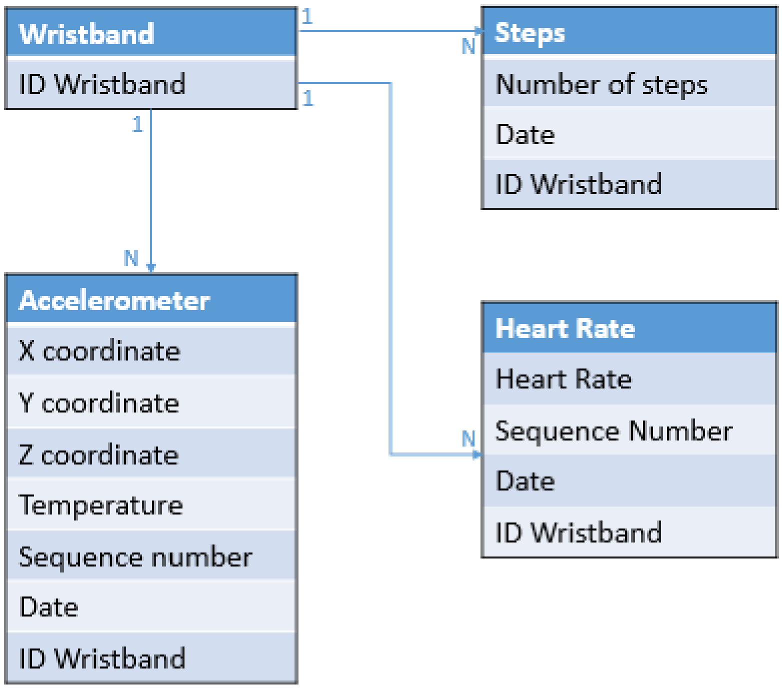


Figure 15 Data model for Smartphone

GPS location

The application has another module that store the GPS position when the device is located outdoor.

When the module "Scenario Detection" decides that the device is located outdoor, it notifies this module to start storing the GPS location. In the event that the device has the GPS disabled, the system enables the interface automatically.

The data of GPS location stored in the database are:

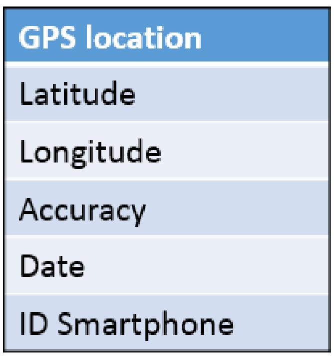


Figure 16 GPS location

Gateway

This application component is responsible to send the data stored on the device to the remote MYSPHERA server when the home wireless network is at range. The data is sent, using the http protocol, to a REST service hosted in the remote MYSPHERA server. A HTTPS security mechanism will be added by the final version of the application, in order to preserve the data from the users, even though these are completely anonymous.

When data is sent to the server, the application waits for a response declaring if the data have been stored correctly. If this happen, the app marks the data as sent in database, enabling local removal.

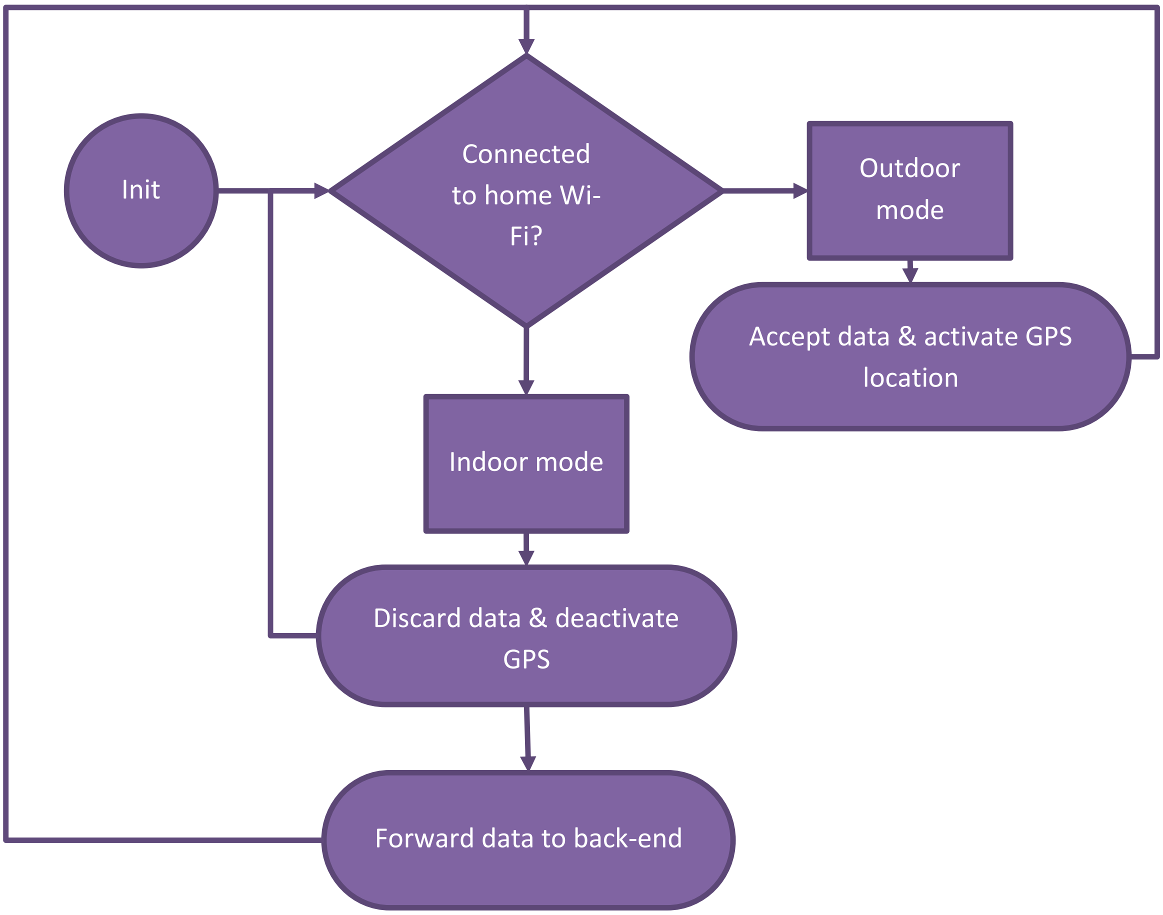


Figure 17 Smartphone modes

Deployment

The device runs an Android application that performs all the operation described in this section. The application runs in background and does not need any user interaction to work if the configuration is correct.

The application will came preinstalled in the devices when the pilot starts in each site. The system settings that the devices need are: configuration for the home Wi-Fi network, GPS and Bluetooth enabled.

The app behaviour will be assessed remotely by monitoring logs set by the developer. This remote logging can be performed by using a third party application such as DeployGate. Remote logging tools provide a way to substitute the Android built in logging, allowing to monitor remotely the logs generated by each instance of the application installed.

Maintenance

The user must perform some periodical actions and keep a specific configuration to enable the correct working of the device. Specifically, the user must:

- Charge the smartphone each night.
- Keep Bluetooth interface enabled.
- Keep Wi-Fi interface enabled.
- Keep GPS interface enabled.

As it is mentioned above, the interaction between the user and the application is minimal. Thanks to the application running in background. If the user does not change the wireless connection or location permits in the device, the user does not have to interact with the app.

The final version of the app will have a special alert to remember the users to charge the smartphone.

For technical information and diagnosis, a very simple user interface displays technical information about the wristband such as ID, parameters and last events.

4.1.6. Wristband

The activity recognition wristband is a custom device aimed at taking continuous measurements of physical activity parameters, as well as vital signs of the DOREMI user during his daily life. This device works as a continuous monitor, independent and unobtrusive for the patient; thus, beyond the sensing, the electrical characteristics and the data format or availability, there are other important aspects to be considered in the design: form factor, battery life, usability, and comfort. The user will wear the device all day, so a correct tailoring and a special attention to the comfort will be especially relevant in the manufacturing process.

Compared to other commercial solutions, DOREMI wristband has some key strengths. It is specially designed for elderly, being easy to attach it to the user arm and keeping the user interface as simple as possible. It comes ready to use, the user only has to keep it charged and wear it all over the day, avoiding complex configuration processes.

It implements MYSPHERA technology to locate the user at room level when at home. This brings useful data for daily routines analysis.

Compared to most of the well-known commercially available products the device provides raw accelerometer and heartbeat sensor readings instead of preprocessed and aggregated data. This feature is particularly important in the context of the DOREMI project as it enables the development of activity recognition tasks customized on the elderly.

Requirements

- Movement (Mandatory): Gathers acceleration data at no less than 5 samples per second.
- Step Counter (Mandatory): From acceleration data, it estimates a step counter with less than 20% error (same as average commercial pedometer). It is provided at a rate equal or higher than 6 times per minute.
- Heart rate (Mandatory): Gathers heart rate with less than 25% error (same as average commercial heart rate monitor) at no less than 3 samples per minute.
- Location (Mandatory): Sends a radio signal to the location system at no less than 6 times per minute.
- Communications (Mandatory): Sends all gathered data (Acceleration raw data, step counter and heart rate) wirelessly to the server.
- Data integrity (Mandatory): No data is lost in the bracelet. The communication process ensures all data is finally stored in the server, including a timestamp for every sample.
- User Interface (Mandatory): Includes a LED to notify the user error situations, lack of connectivity and battery status.
- Battery operated (Mandatory): The device is powered from a rechargeable battery providing an autonomy of at least 16 hour of continuous operation. Battery charge level is regularly monitored and the following states are notified to the user and the server: low battery, battery charging and battery fully charged. The device has a connector for charging purposes.
- Wireless charging (Desirable): The device is charged wirelessly when placing it on a charging base.
- Strap and housing (Mandatory): The wristband is able to be attached to the wrist of the user in an effective way order to acquire data with an acceptable precision being comfortable to the user.
- Water proof (Desirable): the device is sealed and resistant to water.

Specifications

- Strap: A Velcro strap is a solution comfortable easy to attach and detach. It keeps the wristband tight to the user arm.
- Case: Waterproof. Made of hypoallergenic plastic. Easy to wear. It has a window in the bottom side for the heart rate sensor and the LED notifications are visible on the top side.
- Battery operated: At least 16 hours of battery life. Rechargeable.
- Easy to charge: The actions to be done by the user to charge the device are as simple as possible. Wireless charging is preferred.
- Battery charge monitoring: Monitors the battery charge level, providing the curve to the system and notifications to the user.

- Bluetooth Low Energy: Implements this radio communications technology to interact with the rest of the system.
- Accelerometer: Low power, small size, 3 axis accelerometer.
- Hearth rate sensor: Optical sensor suitable for pulse oximeter applications (photoplethysmography).
- Heart rate algorithm: Wristband runs a heart rate calculation algorithm on the raw data provided by the optical sensor.
- Step Counter algorithm: Wristband runs a step counter algorithm on the raw data provided by the accelerometer.
- LED: One LED as user interface.
- Location data: Provides data for the location system.
- Connection management: Wristband communicates with both the DOREMI smartphone and the location system, ensuring no data is lost.
- Data provided: Wristband provides heart rate, accelerometer raw data, step counter, battery status, as well as the necessary information for the location system algorithm. The sample rate[3] for every type of data is:
    - Heart rate: 10 seconds
    - Accelerometer raw data: 0.1 seconds
    - Step counter: 10 seconds
    - Battery status: 1 minute
    - Identification: 3 seconds

Wristband interface

DOREMI wristband produces the following data packets that are emitted over the air through the Bluetooth Low Energy radio protocol:

1. **Wristband identification (for location purposes):**

SPH Header	ID SPH	IEEE Address	RSSI	SPH Header	ID SPH	Sequence Number	RSSI reference
0x08	0x3D	6 bytes	1 byte	0x03	0x2C	1 Byte	1 Byte

This packet includes the necessary parameters for the location algorithm.

2. **Heart rate**

SPH Header	ID SPH	HR	Nº Seq

[3] Sample rates are subject to change in the final prototype.

0x03	0x2F	1 byte	1 bytes

This packet includes the last calculated heart rate (current heart rate) and a sequence number as a timestamp.

3. Step Counter

SPH Header	ID SPH	Step counter
0x03	0x2E	2 bytes

This packet includes the step counter actualized count.

4. Accelerometer raw data

SPH Header	ID SPH	X	Y	Z	Temp	Nº Seq
0x07	0x2A	1 byte	1 byte	1 byte	1 byte	2 bytes

This packet includes the accelerometer raw data (X, Y, Z axis), a temperature byte (for calibration purposes) and a sequence number as a timestamp.

5. Battery status

SPH Header	ID SPH	Battery status	Charge level
0x03	0x36	1 Byte	1 Byte

This packet includes the battery status (in charge, fully charged, low battery) and the battery current charge level.

Usage and limitations

The use of the wristband is very straightforward. It is attached to the wrist just after the user wakes up and until going to bed. Only while having a shower it has to be removed in order to avoid problems due to water. It is important that the wristband is tight and correctly placed on the wrist.

The device has to be charged every night, connecting it to the provided charger.

Sometimes the device lights a LED to give a notification such as battery low indication for instance. The user must understand these notifications to address the corresponding actions to ensure proper functioning.

Maintenance

The main maintenance issue of this device is the need to recharge its battery periodically. Given the early stage of development, it is not possible to give more details about this process at this stage. Several charging methods are being research in order to provide the best for the end users and the more convenient to reach an acceptable form factor and an affordable price.

Integration

The wristband is designed to work with the rest of the system, not as a standalone device. It must have communication either with the RTLS or the DOREMI smartphone. It implies that the wristband needs to be in range of any of the stated subsystems. In most cases, this range is 10 meters as maximum.

The communication is established automatically, so there is no configuration needed in the wristband side.

Battery recharging is done using the provided charger or charging dock. The time necessary until fully charge is specified in the user manual, that will be provided with the final prototype in the MS5 (October '15). The wristband or the charging dock notify when the wristband is charging and when the recharge is completed.

4.1.7. Location system

MYSPHERA's Real-time location system is used to automatically identify and track the location of objects or people in real time within a building or other contained area. Wireless RTLS tags are attached to objects or worn by people, and fixed reference points receive wireless signals from tags to determine their location.

The RTLS technology used for DOREMI is Bluetooth Low Energy, which is a wireless personal area network technology designed for applications in the healthcare, fitness, beacons, security, and home entertainment industries. Compared to Classic Bluetooth, Bluetooth Low Energy is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range.

- Bluetooth Low Energy wireless technology features:
- Ultra-low peak, average and idle mode power consumption
- Ability to run for years on standard coin-cell batteries
- Low cost
- Multi-vendor interoperability
- Enhanced range

The latest Bluetooth specification uses a service-based architecture based on the attribute protocol (ATT). All communication in low energy takes place over the Generic Attribute Profile (GATT). An application or another profile uses the GATT profile so a client and server can interact in a structured way.

The server contains a number of attributes, and the GATT Profile defines how to use the Attribute Protocol to discover, read, write and obtain indications. These features support a service-based architecture. The services are used as defined in the profile specifications. GATT enables you to expose service and characteristics defined in the profile specification.

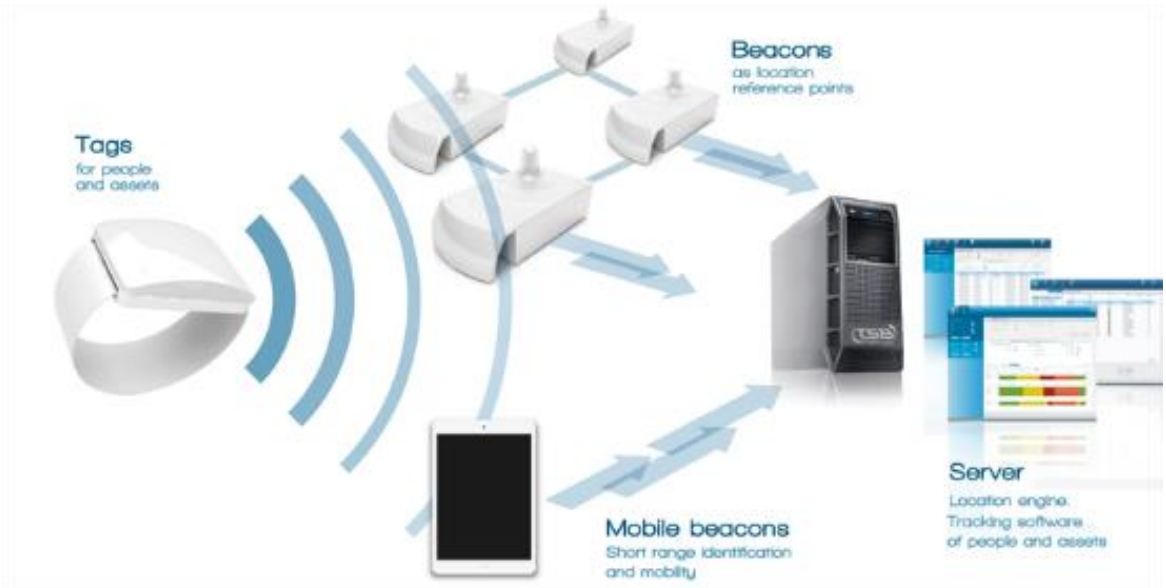


Figure 18 MYSPHERA RTLS working diagram

MYSPHERA's real time location system uses a wristband as a tag which communicates with fixed beacons that recover the data from the wristbands and collect it into a server. An algorithm is used to triangulate the location of the tag using the power of the signal received by the beacons. The tag-beacon communication occurs approximately every three seconds.

The Location system needs fixed beacons connected to a power socket in every room and a concentrator beacon connected to a router so data is sent to the server. MYSPHERA's software will manage data received from the system and send it to the middleware.

4.1.7.1. Wireless beacons

The hardware of the wireless beacon has been developed completely for DOREMI. Despite MYSPHERA counted with a previous version of beacons for RTLS, the impossibility of wiring with Ethernet the apartments of pilots have forced to develop a new type of wireless beacons that only need to be plugged to power sockets and uses BLE both for network communications and data transmission. Currently it has developed a first version of firmware that allows reception data from wristbands and sending this data to the main beacon.

The wireless beacons are powered by USB cable and charger. This allows much versatility in installation.

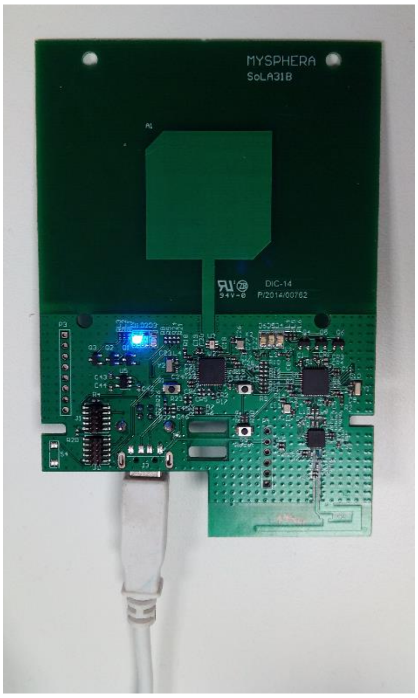


Figure 19 Wireless beacon

The Hardware includes the following components:

- Bluetooth transceiver CC2541
- Bluetooth transceiver CC2540
- Range Extender CC2590
- 2 Sets of 3 LEDs for user interface

4.1.7.2. Main beacon

The main beacon is responsible for collecting all information of wireless beacons and transmit them to the server. In MYSPHERA infrastructure, the main beacon is a transparent proxy between home and the server.

This device is connected via Bluetooth to all authorized wireless beacons to collect all the information transmitted by the tags. Main beacon is continuously collecting the information and sending it to the server to store or calculate the necessary data.

Main beacon will be Intel Galileo Board Gen 2 with dongle Bluetooth BLE. At this moment, we have a basic Linux running a very simple BLE application. It connects with wireless beacons and get all data and send these data through the serial port. The next step is to send this information to the server.

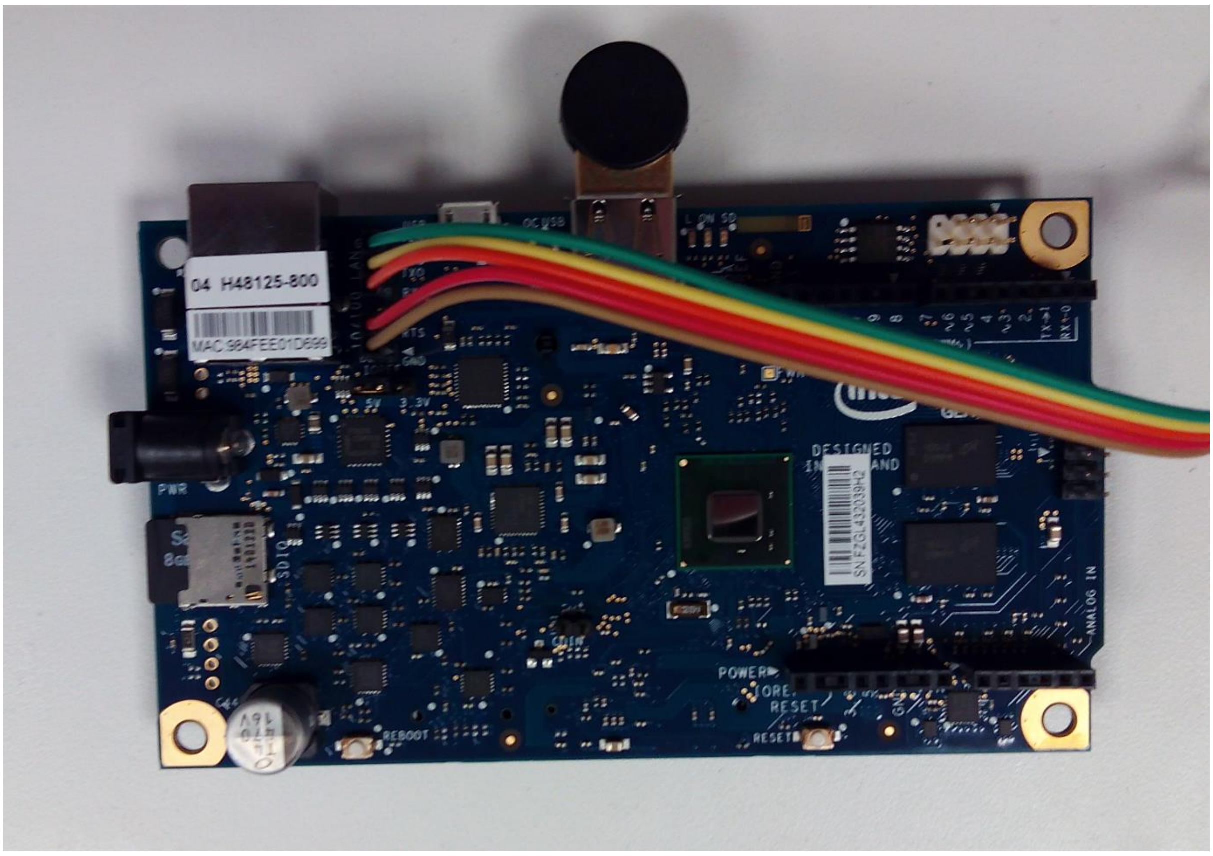


Figure 20 Main beacon prototype based on Intel Galileo Board

4.2. Back-end components

Back-end components are software services at the bottom of the DOREMI architecture layers. These services will be deployed in virtual machines provided by AIT.

4.2.1. Virtual Private Network server

The VPN server is the service that enable technical staff to access DOREMI Gateways from their workstations, overcoming the private IP address limitation of the indoor WSN LAN hosts. The VPN solution eases the overall deployment of the DOREMI Gateway as no further configuration is required on other systems, exception made for the gateway and the VPN server.

Requirements

The VPN server requires a virtual machine with the following resources:

- CPU: 1,5 GHz single-core processor equivalent or better
- Memory: 512 MB or more
- Storage: 10 GB
- OS: Ubuntu server LTS

- Networking: an interface with public IP addressing is required.

The OpenVPN software will be installed on the aforementioned VM. A PKI (public key infrastructure) will then be established, using the OpenSSL software suite, by producing a Certificate Authority certificate and separate public and private keys for the VPN server and each client. These certificates will also be used for authenticating clients on the MQTT broker.

The OpenVPN server will be configured in order to:

- Provide a private IP address to each connecting host in the class “A” IPv4 network range (10.0.0.0/8) on the basis of the client certificate. Each client certificate pair with an IP private address and a hostname.
- Setup a TLS encrypted communication channel over TCP or UDP protocol
- Push DNS server configuration to each the client
- Enable client-to-client communications for hosts in the VPN

The same virtual machine will be configured with a DNS server. The latter will serve hostname resolution for DOREMI Gateway hostnames to clients connected to the VPN, allowing easier access to those machines by means of meaningful hostnames. Bind9 server application will provide the DNS service.

4.2.2. Message broker

The MQTT message broker is the service that provides the communication infrastructure for Sensor Weaver.

The identified implementation of MQTT broker for DOREMI is Mosquitto, a lightweight open source message broker that implements the MQ Telemetry Transport protocol version 3.1 and 3.1.1. It features the following characteristics:

- Support for encrypted communications
- Support for credentials based or certificate based authentication of clients
- Support for authorization on topic access
- Brokers bridging

In DOREMI, a dedicated Virtual Machine will host the Mosquitto service. The minimum requirements for the VM are:

- CPU: 3 GHz single-core processor
- Memory: 2 GB or more
- Storage: 25 GB
- OS: Ubuntu server LTS
- Networking: an interface with public IP addressing is required.

The configuration of the service includes:

- Listening port: default 1883
- Enabled persistence: the setting allows Mosquitto to recover the state in case the VM goes offline abruptly, allowing automatic recovering of clients' connection on server restart.
- Logging enabled.
- Anonymous access disabled. Authentication of clients based on certificates.

The Sensor Weaver java client libraries governs the connection with the MQTT broker. Each DOREMI Gateway will use a node certificate, the same used for VPN authentication, in order to connect to the broker.

4.2.3. Sensor database

The *Sensor Database* hosts data collected from WSN sensors across pilot sites. It is accessed by the *Data Recorder* software module described in chapter 4.1.2.1 and by WP4 data processing software components.

The DB solution involves the usage of a document-oriented database MongoDB. The latter eschews the traditional table-based relational database structure in favour of JSON documents with dynamic schemas. MongoDB organize its data in internal databases and, for each database, several collections of heterogeneous documents. The fact of not being tied to a certain schema makes the integration of data of certain types easier and faster in applications where no assumption can be made about the data generated by sources.

In DOREMI, the data model will include one internal database with two documents collections: the *Data Feed collection* and the *Data collection*.

The D*ata Feed collection* contains documents that reports the availability of a given data feed within a given time span. For each document contained, the content reports the properties published by the data feed, the message format and a unique ID used for linking documents in the data collection. Timestamps are provided in ISODate typed values (ISO 8601 standard format). An example document is shown below, where the variable schema part is the content of the element *datafeedDescriptor*.

```
{
        "_id" : ObjectId("54783709986c707baa444775"),
        "scope" : "doremiScope",
        "datafeedDescriptor" : {
                "id" : "BalanceBoard001",
                "properties" : {
                        "sensorType" : "WiiBoard",
                        "deviceId" : "BalanceBoard-58BDA3AC4DA1"
                },
                "messageElements" : [
                        {
                                "name" : "topLeftWeight",
                                "unit" : "kg"
                        },
                        {
                                "name" : "topRightWeight",
                                "unit" : "kg"
                        },
                        {
                                "name" : "bottomRightWeight",
                                "unit" : "kg"
                        },
                        {
                                "name" : "bottomLeftWeight",
                                "unit" : "kg"
                        },
                        {
                                "name" : "buttonPressed",
                                "unit" : "boolean"
                        }
                ]
        },
        "startTime" : ISODate("2014-11-28T08:49:13.793Z"),
        "endTime" : ISODate("2014-11-28T08:49:42.697Z")
}
```


Figure 21 - Sample Data Feed Collection document

The data collection contains data generated by sensors.


```
{
        "_id" : ObjectId("5478373e986c6ddfc36a43d4"),
        "id" : "BalanceBoard001",
        "timestamp" : ISODate("2014-11-28T08:50:06.492Z"),
        "values" : {
                "topRightWeight" : "0.0",
                "buttonPressed" : "false",
                "topLeftWeight" : "0.0",
                "bottomRightWeight" : "0.0",
                "bottomLeftWeight" : "0.0"
        },
        "offeringId" : ObjectId("5478373b986c6ddfc36a43d3")
}
```


Figure 22 - Sample Data collection document

Indexes are implemented on both the collections to speed up queries. Since the expected queries will ask for a given set of data generated by a sensor with a given id in a given time span, the *Data Feed* *collection* will have indexes on fields: datafeedDescriptor.id, startTime, endTime. The *Data collection* includes indexes for the following document elements: id, timestamp.

The data query time by simulating a scenario with 20 pilot DOREMI users and the following sources of data:

- 6 PIR sensors: variable sample rate
- 1 Door contact sensor: variable sample rate
- 1 Wristband accelerometer sensor: Fixed sample rate 20Hz, 18 hours of daily usage
- 1 Wristband heartbeat sensor: Fixed sample rate at 1Hz, 18 hours of daily usage
- 1 GPS: Fixed data rate at 0,2 Hz, 8 hours of daily usage
- 1 Indoor localization virtual sensor: Fixed sample rate at 0,1 Hz, 16 hours of daily usage
- 1 Smart Carpet: fixed amount of data generated each time the user steps on the board (100 samples/sec for 15 secs), 8 daily uses of the device.

The simulation scenario is overly pessimistic in terms of data produced by sensors and activation of devices by the end user.

The data simulation spans on 3 months producing a total of 31050 MB of data for each user (around 30 GB). Considering the simulation, 2 Terabytes of database storage should accommodate the amount of data of the DOREMI pilot with 20 users.

Query times were also tested according to the scenarios above and resulted in less than 1ms with indexes implemented.

Interactions with the rest of components

Client applications that need to access data can use the native MongoDB wire protocol based on the BSON specification. Official and community supported client libraries are available for the most used programming languages: http://docs.mongodb.org/manual/applications/drivers/.

Requirements

An instance of MongoDB will be hosted on a Virtual Machine with the following requirements:

- CPU: 3 GHz quad-core processor or equivalent
- Memory: 4 GB or more
- Storage: 2 TB
- OS: Ubuntu server LTS
- Networking: Private IP address. Firewall configuration must allow only incoming connections from a set of other VMs in the same local network.

4.2.4. Autoconfiguration system

The aim of the auto configuration system is to enable a (semi)-automatic central configuration of the sensor network simply and in a convenient way via a graphical user interface. It will provide the sensor configuration to the gateways installed in the users' homes which connects to the configuration system over the internet.

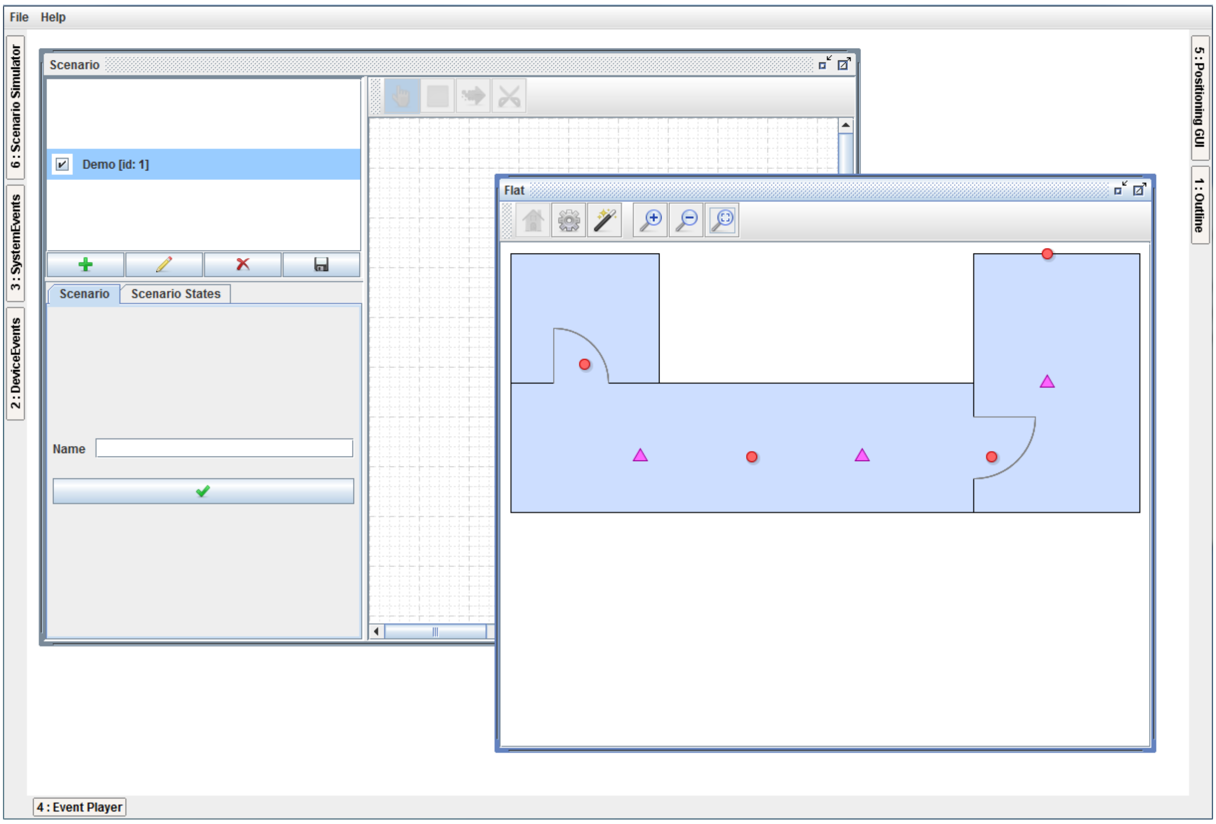


Figure 23: Graphical user interface of the autoconfiguration system

Services offered and APIs

The flat and sensor configuration is performed via a graphical user interface (Figure 23) and stored in a SQL database (PostgresQL or SQLite). A GUI dialogue (Figure 24) is used to configure the details of specific sensors or actuators. These are up to three hardware IDs, the device type, the protocol and the position within the flat. The meaning of the hardware IDs differs from protocol to protocol, but the first is usually the unique serial number of the device. The second and the third can be the packet identifier or sub-sensor number (used for multi-sensor devices). The representation within the auto configuration system does not reflect the hardware physically, but logically. That means, a physical device can have multiple logical representations within the system for different functions (e.g. a sensor device that supports brightness, motion and temperature measurements).

The auto configuration system provides interfaces to third-parties like REST, Websocket or RabbitMQ. By client request, the whole flat configuration is sent back as a serialized Java Object in JSON format via de dedicated protocol. The request is also sent as a JSON string and forward to the internal event bus by the user interface bundle. The flat configuration request is defined as follows:

```
{"topic":
"at/ac/ait/hbs/homer/event/homecontrol/flat/configuration/request",
```

```
  "properties": {

    "TAG": "e50a8883-2074-44dc-ab45-94611efbea44",

    "VALUE": 1

  }

}
```


Its response is of the form:


```
{
  "topic":
"at/ac/ait/hbs/homer/event/homecontrol/flat/configuration/message",
  "properties": {
    "FLAT": {
      "id": 1,
      "name": "Demo",
      "devices": [
        {
          "id": 1,
          "type": 1001,
          "category": "ACTUATOR",
          "roomId": 1,
          "coordinate": {
            "index": 0,
            "x": 10.0,
            "y": 10.0
          },
          "areaCoordinates": [],
          "metavalues": {
            "HARDWARE_ID": {
              "key": "HARDWARE_ID",
              "type": "NUMERIC",
              "value": "100"
            },
            "CONFIGURATION": {
              "key": "CONFIGURATION",
              "type": "STRING",
              "value": ""
            },
            "MANUFACTURER": {
              "key": "MANUFACTURER",
              "type": "STRING",
              "value": "KNX"
            },
            "GATEWAY": {
              "key": "GATEWAY",
              "type": "NUMERIC",
              "value": "1234567"
```

```
                }
            }
        },
}
```

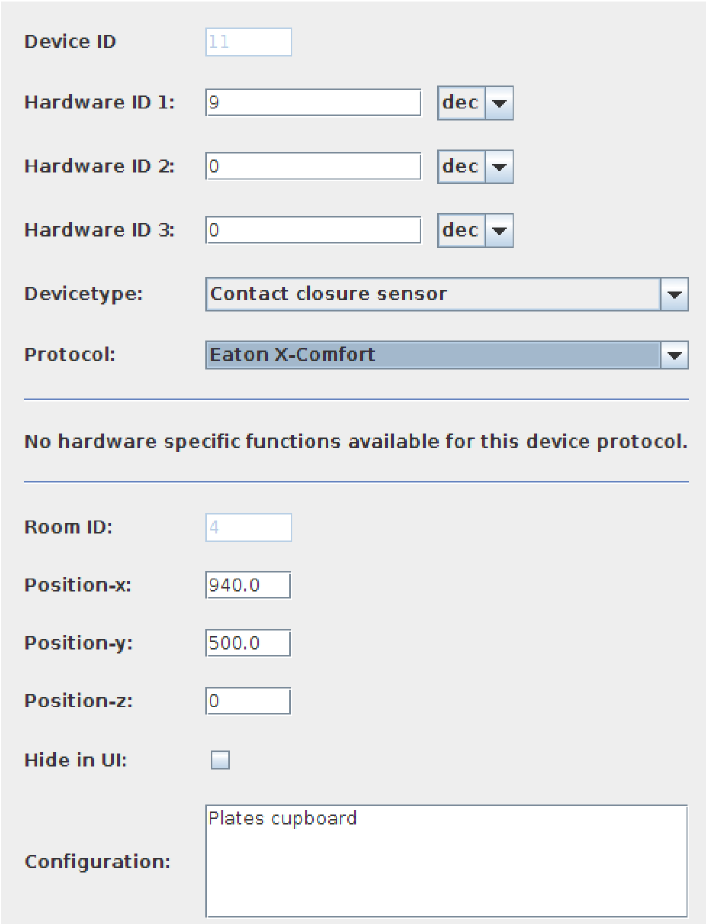


Figure 24: Device configuration dialogue

Configuration

The configuration for DOREMI will be straight forward. Among the large amount of available bundles for the system only a subset will be used, mainly database, GUI and a connector (most likely REST).

Hosting

The auto configuration is provided by AIT via a Virtual machine connected to the internet. The virtual machine runs an Ubuntu Linux operation system, the auto configuration system runs on a JVM. The auto configuration system will be accessible by the partners via SSH and/or TeamViewer, which provides also a graphical user interaction.

The autoconfiguration system is written in Java and is structured in OSGI bundles running inside the KARAF framework. (Figure 25). This allows the system get updated during runtime without a full shutdown and high reliability.

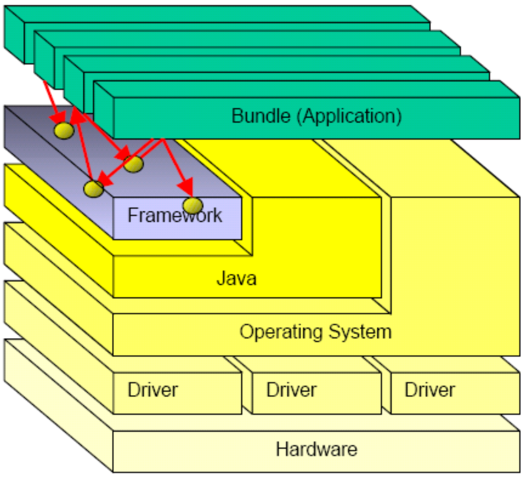


Figure 25: Architecture of Java Software inside the KARAF Framework

4.2.5. Localization and wristband data back-end

The wristband transmits data to RTLS beacons when the user is at home and to a mobile phone while the user is outdoor.

MYSPHERA Localization and data back end server will be in charge of collecting the data from the wristbands received from beacons as well as mobile devices. The server will remove duplicated information, calculate indoor localization, and forward the processed information to the rest of the components using the integration middleware API.

4.2.5.1. Middleware events

Each wristband will be published as several sensors using the middleware API, below the properties and parameters of each of these sensors are listed:

4.2.5.1.1. Heart rate

Property	Value	Description
sourceId	MAC of Wristband	It is the Bluetooth Mac address of the wristband
sensorType	**hr**	Id of Heart Rate sensor

Parameter	Format/Unit	Description
hr	Beats per minute	Heart rate in beats per minute
Timestamp	32-bit integer	Time in UNIX Epoch (with milliseconds)
seqNumer	8 bit integer	Used to detect duplicated and lost frames

4.2.5.1.2. Step counter

Property	**Value**	**Description**
sourceId	MAC of Wristband	It is the Bluetooth Mac address of the wristband
sensorType	**stepCounter**	Id of step counter sensor type

Parameter	Format/Unit	Description
stepCounter	Steps (16 bits integer)	Accumulated step number of steps
Timestamp	32-bit integer	Time in UNIX Epoch (with milliseconds)

4.2.5.1.3. Accelerometer

Property	**Value**	**Description**
sourceId	MAC of Wristband	It is the Bluetooth Mac address of the wristband
sensorType	**accelerometer**	Id of accelerometer sensor type

Parameter	**Format/Unit**	**Description**
seqNumer	16 bit Integer	Used to detect duplicated and lost frames
Timestamp	32-bit integer	Time in UNIX Epoch (with milliseconds)
x	8 bit signed integer	Value represents range of [3, -3] g
x	8 bit signed integer	Value represents range of [3, -3] g
z	8 bit signed integer	Value represents range of [3, -3] g
temperature	Celsius	Temperature used to calibrate some accelerometers

4.2.5.1.4. Indoor Localization information

An event will be generated when the wristband changes from one area to another

Property	**Value**	**Description**
sourceId	MAC of Wristband	It is the Bluetooth Mac address of the wristband
sensorType	**locationTrace**	Id of location trace sensor

Parameter	**Format/Unit**	**Description**
timestamp	32-bit integer	Time in UNIX Epoch (with milliseconds)
mapId	String	Id of the map enclosing the area
areaId	String	Id of the area where the bracelet is being located.

4.2.5.1.5. GPS localization information

The mobile phone carried by the users will provide GPS information when the user is outdoors.

Property	**Value**	**Description**
sourceId	MAC of Wristband	It is the Bluetooth Mac address of the mobile device
sensorType	**gps**	Id of the GPS sensor type

Parameter	**Format/Unit**	**Description**
timestamp	32-bit integer	Time in UNIX Epoch (with milliseconds)
latitude	double	Latitude in degrees

longitude	double	Longitude in degrees
accuracy	double	Accuracy in meters

4.2.5.2. Server requirements

The server is based on JBoss Application Server[4], it can run on Windows or Linux and has the following hardware requirements:

- RAM: 4GB
- HD: 40GB
- CPU: 2 CPUs

[4] http://jbossas.jboss.org/

5. DEVELOPMENT STATUS AND NEXT STEPS

By the delivery date of this document, the development status of the WSN follows the plan marked in the DoW and updated in D3.1. The release of this document coincides with the release of the fourth milestone of the project: preliminary version of the WSN environment, smart environment for context awareness and gamified environment. Up to now, all the developments are according to plan and in several tasks and subtasks, the progress have overtaken the plan (middleware, smartphone gateway). The following subsections summarize the development work done in the period between the last deliverable and the current. This chapter also describes the future actions to accomplish the WP3 and the overall project plans.

5.1. Execution of the development plan

5.1.1. DOREMI Gateway and Middleware

The Middleware components that enable secure communication of sensor data collected by the DOREMI Gateway is at a mature stage. Both the DOREMI Gateway components and the back-end services (MongoDB Sensor Data recorder) have been released in a stable version 2.0 in December 2014. Documentation is also available at http://wnlab.isti.cnr.it/sensorweaver. Development of additional services could involve:

- Development of a Web Application and DOREMI Gateway software bundles that enable real-time monitoring of all the WSN deployed in a single view.
- Development of a secure REST web service for batch upload of sensor data on the DB.
- Bug fixing and development of new Middleware features, which requirements could arise during the integration phase.

These activities do not hinder the integration phase as developers can already use the facilities offered by Sensor Weaver to integrate their sensors with the Middleware.

Further work will involve the development of scripts and tools that simplify the preparation of the DOREMI Gateways for the pilot sites. This task should not to be underestimated since the number of Gateways and the number of services to be installed and configured on each node can be cumbersome if deployed manually.

5.1.2. Wristband

The development of the custom wristband is one of the most challenging tasks in WP3, since it implies the design, development and manufacturing of a hardware device from scratch in a period of 18 months. Concerns about the convenience of the choice of this way to implement the data collection from the user have as well as related to the restrictive development plan have been raised in several occasions during the execution of DOREMI. These concerns have been considered as priority in the work package and in the whole project, since critical activities of WP4 and WP6 are highly dependent on the success of this WP.

A brief of the motivations to choose the development of an own wristband are the following:

- APIs from market devices offer a limited access to the data, impeding a free access to raw data in a high sampling rate (10 Hz.
- Most of the commercially available, by the time of the project kick off, required user intervention to take heart rate measurements. These are usually requested to be taken in resting conditions.
- There are not devices in the market providing indoor location among user quantification.
- The need of designing a device specially tailored for elderly people.
- The full access to the device allows quick changes to meet future user needs or to perform changes to refine the effectiveness of the DOREMI system.

The development plan concerns have been addressed by the creation of a reliable detailed plan of development, introducing and intermediate milestone in January of 2015 (Month 15). The plan was released in D3.1 second version and it is attached as an annex in this deliverable.

5.1.2.1. Lab prototype (Jan '15)

The first prototype a simple combination of modules to test interoperability between them and validate the parts selection. It consists of 3 modules interconnected by SPI. The following figure shows the basic outline of communications.

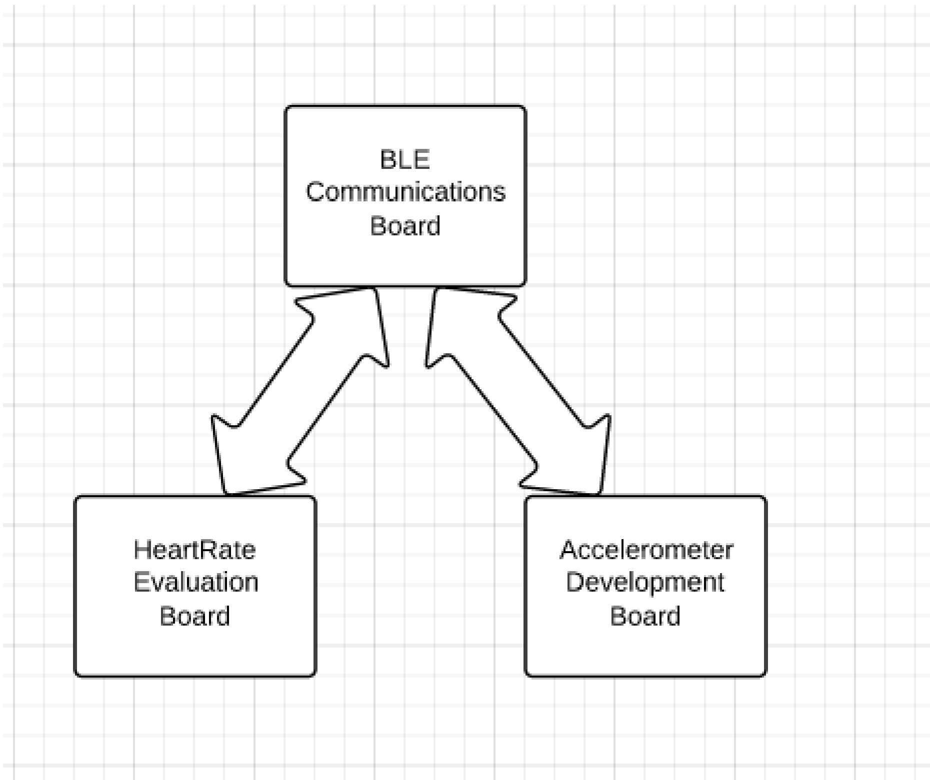


Figure 26 Lab prototype schema

For this prototype we used a BLE Communications Board from MYSPHERA indoor location system, a HeartRate evaluation Board (AFE44x0SPO2-FE) form Texas Instruments and Accelerometer Development Board (MSP-EXP430FR5739) form Texas Instruments.

The next figure shows the final prototype.

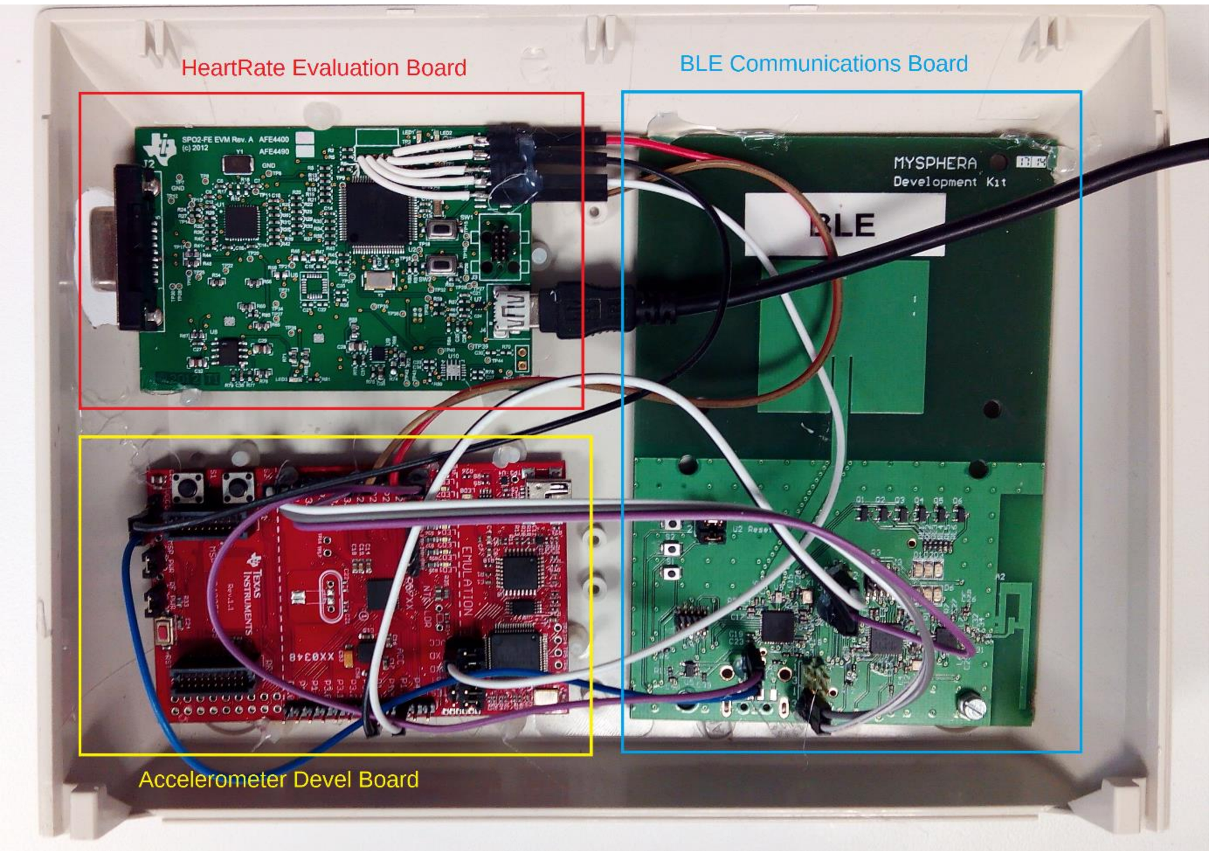


Figure 27 First lab prototype

In this prototype the following features are developed:

- **Activity detection**: The accelerometer is sampled at 10Hz.
- **Step counter**: With the accelerometer raw data, step counter is calculated. Typical error 20%.
- **HeartRate**: in this prototype this parameter is measured through a finger probe. Every 5 seconds sampling was done to determine the pulse.
- **Location**: Locations frames is sent every 3 seconds.
- **Communications**: Acceleration raw data, step counter and heart rate are sent wirelessly to the server.
- **User interface**: Accelerometer Development Board LEDs were used as user interface.

The wireless beacon used for receiving data is the prototype developed for DOREMI, detailed in section **¡Error! No se encuentra el origen de la referencia.** of this chapter.

Figure 28 Wireless beacon prototype

5.1.2.2. Integration and data collection prototype (Apr '15)

The first wristband hardware prototype is currently designed and it is being manufactured, expected to be released in the next few weeks.

The housing is made in resin from a 3D design. The dimensions of this first prototype are 47.5 x 52.5 x 15 mm.

The strap allows a fine tuning to attach the device properly.

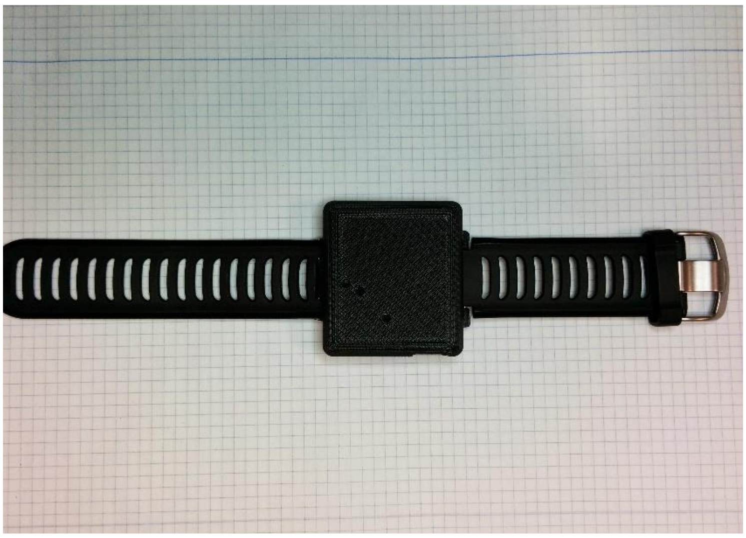


Figure 29 First integration and data collection prototype housing

The Hardware includes the following components:

- Analog front end (AFE4403) including the photoplethysmography sensor (NJL5310R)
- Low power accelerometer ADXL362
- Battery with 300mAh capacity
- Microcontroller MSP430F5528
- Bluetooth transceiver CC2541
- Battery charge monitor BQ27425
- Battery charge manager BQ24072
- Wireless charge manager BQ51020

This prototype aims to cover all requirements stated above in this document. It will allow to test the performance and convenience of the two options for charging the device: through micro-USB connector and wireless charge.

The algorithms designed for the lab prototype are to be tested with this prototype and adjusted afterwards to meet the accuracy required.

A more precise battery life estimation is to be determined from the tests over this prototype. The final prototype may have a battery with a capacity up to 33% higher, and a longer battery life.

About the size and shape of the case, there is room for improvement. The whole dimensions of the wristband will be shrank in the final prototype.

5.1.3. Smart carpet

The development of the components needed to setup the smart carpet in the DOREMI pilot is at a mature stage and a working version of the Middleware integration bundles is already implemented at the time of writing of this deliverable.

The initial use case devised in the DOREMI protocol for the Smart Carpet saw the device placed at the side of the bed. According to the use case, the user should step on the device every time he gets up from bed. The scenario involved the activity of stepping up on the balance from a sitting position and then walk off from the board.

Afters some issues raised by the pilot site management partners, the use case was discarded because of safety concerns involving the possibility that the user could stumble and fall in some circumstances. It was therefore decided to allow device positioning in a safer place and modify the use case scenario, discarding the sitting-to-standing user action.

The new solution involves placing the device in the living room or in a room where there is enough space to easily access the device without harm on user mobility. The preferred location for the balance board is the living room because the solution would allow the aggregation of most of DOREMI hardware (router, DOREMI gateway, z-wave gateway, smart carpet) in the same spot.

The initial idea of producing a modified version of the balance board with side ramps for greater accessibility was also discarded as the final product proved to be bulkier without providing any consistent additional safety to the user.

5.1.4. Network elements

The network elements are 100% developed in terms of hardware design and firmware development. It is still pending the choice of an appropriate housing for each element, to be done for the last release in October of 2015. The manufacturing of the number of devices needed for the pilot phase it is already planned to be available for the release date.

5.1.5. Auto calibration and configuration system

AIT's HOMER system, which acts as a base for the autoconfiguration system in DOREMI will be modified mainly in terms of its interfaces to fulfil the DOREMI requirements. A virtual server for running the system will be provided. Z-Wave sensors will be added to the hardware abstraction layer.

5.2. Integration tests in Living Lab

The immediate next step after the release of the preliminary version of the WSN together with this document will be the schedule a week to perform integration tests with the rest of preliminary versions of the DOREMI components (WP4, WP5).

This activity, which is enclosed in the task 6.2 of WP6, will be performed in CIAMI Living Lab (Valencia). An infrastructure to which MYSPHERA has full access.

CIAMI is an infrastructure that combines on one hand the simulation of a living space where anyone could live in total comfort and on the other hand, integrated technologies into the physical environment. Its purpose is to provide an intelligent space for testing technological prototypes with their potential users in real conditions.

Figure 30 Exterior picture of the CIAMI Living Lab

Figure 31 Indoor setup of the CIAMI Living Lab

This experience is expected to last about one week, counting with the installation and configuration tasks and the integration tests and data collection experiences. It will be scheduled between May ’15 and June ’15, depending on the availability of the involved partners.

The design of the tests in Living Lab are a competence of WP6 specifically allocated in Task 6.2. Despite these tests are not still completely defined and specified, they will imply, in general terms, the following activities:

- Systems integration: it will be tested the interoperability between the systems developed in WP3 (WSN), WP4 (context awareness) and WP5 (gamified environment) and the correct data flow.
- Installation tests: it will be measured the installation time in a real scenario, annotating issues and difficulties that could appear. Additionally, an exhaustive list of materials will be generated, including small equipment.
- Usage tests: the system working will be tested with real usage tests, including the following activities. A script for each activity will be developed in order to have a reference document:
    - Standing from bed and using the smart carpet.
    - Performing normal daily activities such as watching the TV, cooking or doing the laundry.
    - Meeting with people at the apartment.

5.3. Pending data collection

The activities of data collection have been continued since the moment the first results of WP3 have permitted obtain valid data for pilots. The status of data collection for each of the different subsystems is as follows:

- Smart carpet: completed
- Environmental sensors: completed
- Wristband: pending; expected to be done during the Living Lab integration tests.

5.4. Pilots deployment

The pilot sites preparation has already begun. Several virtual meetings and physical meetings have been held within the period between D3.1 release and the current delivery. As reported in D3.1 final version, chapter 14, blueprints from the pilot sites have been provided and a preliminary study of the installation in each site have been done. In that document are available schemas of the installation planned in each site.

The pilot sites in UK (Shenley Wood and Pannel Croft) have been visited by the WP3 staff, where several meetings with the different partners were held. As a consequence of this activity, several issues were raised and solved later, such as the convenient location of the smart carpet due to safety reasons or installation constrains in the sites. The main issues discussed were:

- Installation: prevent drilling any surface or perform deep changes in the distribution of the apartments.
- Installation: avoid install devices in public places, especially where these could be considered as a mark or signal of health problems of a particular user.
- Network: an Internet connection must be provided by the pilot sites for each participating user.
- Repair: if the installation and later removal of the DOREMI elements imply any deterioration or noticeable modification of the walls, ceiling or furniture of the apartment, they have to be repaired.
- Compensation: the participating users should receive a present or a benefit to compensate their invested time and their possible inconveniences.

It is expected a visit to the Italian pilot sites during the next 6 month period, in order to prepare this pilot and research for possible issues and prevent problems during the pilots.

The following is the draft list of the devices to be acquired for the deployment in the pilot phase.

Chapter	Item Description	Quantity per pilot	Total units
Infrastructure	**Wireless Router**	**1**	**22**
Real Time Location System	**Beacons**	**5**	**110**
Real Time Location System	**Main beacon**	**1**	**22**
Real Time Location System	**MYSPHERA Server**	**1**	**1**
Wearable Sensors	**Wristband**	**1**	**22**
Mobile device	**Smartphone**	**1**	**22**
Environmental Sensors	**Z-Wave Gateway**	**1**	**22**
Smart Carpet	**Bluetooth Dongle**	**1**	**22**
Smart Carpet	**Wired Battery**	**1**	**22**
Environmental Sensors	**Door contact sensor**	**2**	**44**
Smart Carpet	**Wii Balance Board**	**1**	**22**

DOREMI Gateway Station	**Single-Board PC**	**1**	**22**
DOREMI Gateway Station	**Wireless Dongle**	**1**	**22**
Environmental Sensors	**PIR**	**5**	**110**

6. REFERENCES

[1] Palumbo, F., Ullberg, J., Štimec, A., Furfari, F., Karlsson, L., & Coradeschi, S. (2014). Sensor network infrastructure for a home care monitoring system. Sensors, 14(3), 3833-3860.

[2] CLARK, Ross A., et al. Validity and reliability of the Nintendo Wii Balance Board for assessment of standing balance. Gait & posture, 2010, 31.3: 307-310.

[3] BARTLETT, Harrison; BINGHAM, Jeff; TING, Lena H. Validation and calibration of the Wii Balance Board as an inexpensive force plate. American Society of Biomechanics, 2012, 1.2: 3-4.

[4] BARTLETT, Harrison L.; TING, Lena H.; BINGHAM, Jeffrey T. Accuracy of force and center of pressure measures of the Wii Balance Board. Gait & posture, 2014, 39.1: 224-228.

7. APPENDIX

7.1. DEVELOPMENT PLAN

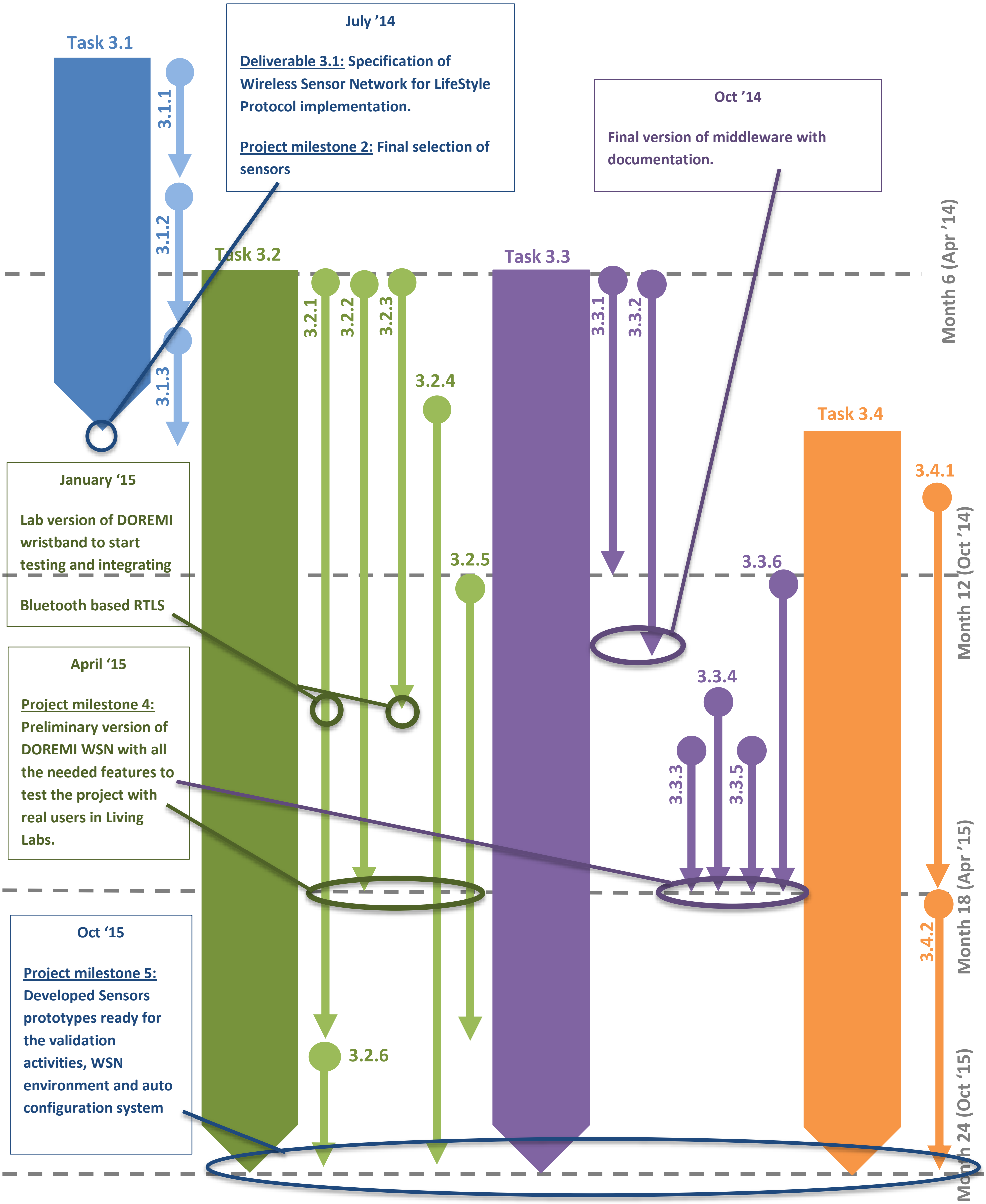
Task	Leader	Contributors	Start date	Due date	Outputs	Intermediate milestones
Task 3.1: Requirements, parameters and sensor selection	MYSPHERA	CNR-ISTI, AIT	01/14	07/14	Deliverable 3.1 Specification of Wireless Sensor Network for LifeStyle Protocol implementation.	
Task 3.1.1 Capture and specify functional and non-functional requirements	MYSPHERA	CNR-ISTI, AIT	01/14	03/14		
Task 3.1.2 Design the network infrastructure	MYSPHERA	CNR-ISTI	04/14	05/14		
Task 3.1.3 Select the sensors based on WP2 and WP4 criteria	MYSPHERA	CNR-ISTI	06/14	07/14	Project milestone 2: Sensor selection	
Task 3.2: Sensors development and production	MYSPHERA	CNR-ISTI	04/14	10/15	Deliverable 3.2: Wireless sensor network, integration middleware and system configuration Deliverable 3.3: Production of units for pilots	
Task 3.2.1 Development of DOREMI wristband	MYSPHERA		04/14	10/15		
Task 3.2.1.1 State of Art of OEM sensors	MYSPHERA		04/14	07/14	State of art of OEM sensors, OEM	

					sensors selection (included in D3.1)	
Task 3.2.1.2 Integration of OEM sensors with hardware development kits	MYSPHERA		08/14	01/15		Laboratory prototype with HDKs (01/15)
Task 3.2.1.3 PCB design with components	MYSPHERA		02/15	03/15	PCB schematics and layout	
Task 3.2.1.4 Development of firmware with detection algorithms	MYSPHERA		02/15	09/15	Firmware source code	Intermediate versions for Living Lab testing (04/15)
Task 3.2.1.5 Development of elderly friendly casing and strap	MYSPHERA		02/15	09/15		Intermediate versions based on 3D-Printing (for Living Lab testing) (04/15)
Task 3.2.1.6 Battery optimization	MYSPHERA		07/15	10/15		
Task 3.2.2 Development of DOREMI smart carpet	CNR-ISTI CNR-IFC		04/14	03/15	Smart carpet solution	
Task 3.2.3 Development of Bluetooth based RTLS	MYSPHERA		04/14	01/15	RTLS solution based on Bluetooth	
Task 3.2.4 Development of DOREMI gateway	CNR-ISTI		07/14	10/15	Gateway solution	Preliminary version for test in Living Lab (04/15)
Task 3.2.5 Development of DOREMI protocols	MYSPHERA		10/14	10/15	DOREMI networking protocols	Intermediate versions with reduced

						functionalities (04/15)
Task 3.2.6 Sensors production	MYSPHERA		08/14	10/15	DOREMI final prototypes	
Task 3.3: Sensor integration and middleware	CNR-ISTI	MYSPHERA	04/14	10/15	Final version of DOREMI middleware	
Task 3.3.1 Development of middleware communication platform	CNR-ISTI		04/14	10/15	Final version of middleware	Working version of middleware ready for sensor integration (12/14)
Task 3.3.2 Documentation of middleware communication platform	CNR-ISTI		04/14	12/14	Middleware wiki documentation	
Task 3.3.3 Integration of DOREMI wristband	MYSPHERA		02/15	04/15		
Task 3.3.4 Integration of DOREMI smart carpet	CNR-ISTI CNR-IFC		01/15	04/15		
Task 3.3.5 Integration of off-the-shelf sensors	MYSPHERA		02/15	04/15		
Task 3.3.6 Development of data storage solution	CNR-ISTI		12/14	04/15		Prototype data storage solution ready for integration (04/15)

Task 3.4: Auto configuration system and calibration	AIT	CNR-ISTI	07/14	10/15	Autoconfiguration system and calibration	
Task 3.4.1 Development of autoconfiguration system and calibration	AIT		07/14	04/15		
Task 3.4.2 Integration with middleware	AIT	CNR-ISTI	05/15	10/15		

In the next diagram it is shown the main tasks of WP3 with their timelines and most important outputs:


Task 3.1
3.1.1
3.1.2
3.1.3
July '14
Deliverable 3.1: Specification of Wireless Sensor Network for LifeStyle Protocol implementation.
Project milestone 2: Final selection of sensors
Oct '14
Final version of middleware with documentation.
Task 3.2
3.2.1
3.2.2
3.2.3
3.2.4
3.2.5
3.2.6
Task 3.3
3.3.1
3.3.2
3.3.3
3.3.4
3.3.5
3.3.6
Task 3.4
3.4.1
3.4.2
Month 6 (Apr '14)
Month 12 (Oct '14)
Month 18 (Apr '15)
Month 24 (Oct '15)
January '15
Lab version of DOREMI wristband to start testing and integrating Bluetooth based RTLS
April '15
Project milestone 4: Preliminary version of DOREMI WSN with all the needed features to test the project with real users in Living Labs.
Oct '15
Project milestone 5: Developed Sensors prototypes ready for the validation activities, WSN environment and auto configuration system

7.2. EVERSPRING SP-814 DATASHEET

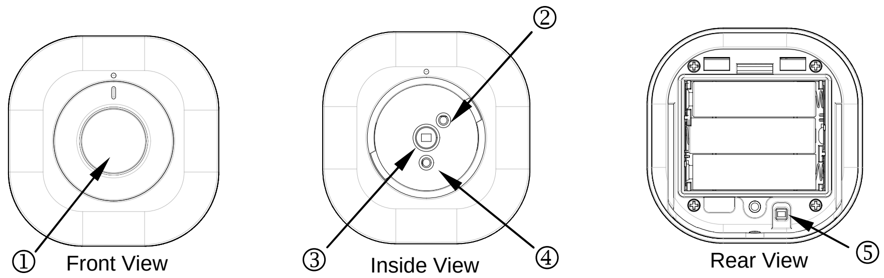
SP814 MOTION DETECTOR

The Motion Detector is a Z-Wave™ enabled device which is fully compatible with any Z-Wave™ enabled network. Z-Wave™ enabled devices displaying the Z-Wave™ logo can also be used with it regardless of the manufacturer, and ours can also be used in other manufacturer's Z-Wave™ enabled networks. This Motion Detector can control our modules via controller setting. Inclusion of this Motion Detector on other manufacturer's Wireless Controller menu allows remote turn-on of connected modules when the detector is triggered.

The Motion Detector is designed with two detecting sensors, Passive Infra-Red (PIR) sensor and light sensor, in order to fulfill the purpose of security and home automation. When the detector is cooperated with security appliances, it is acting as a security device by detecting changes in infra-red radiation levels. If a person moves within or across the device field of vision, a trigger radio signal will be transmitted to cause full alarm condition in order to frighten intruders away. Alternatively, when the detector is worked with home automation appliances, the detector can be set to perform the role of home automation device by detecting both changes in infra-red radiation levels and percentage of lux levels. Once night falls, the percentage of ambient illumination is lower than preset value. If a person moves within or across the device field of vision, a trigger radio signal will be transmitted so as to turn on the connected lightings for better illumination.

Two mounting methods are provided for varying detection range. The detector can be mounted on a wall for farther detecting distance but narrower coverage; while for ceiling mounting, shorter detecting distance can be made but desired coverage can be expected at user's disposal.

Product Overview



① Lens Cover (wall-lens cover and ceiling-lens cover)	
② Photocell Sensor	④ Two-Color Indication LED (red & green)
③ PIR Sensor	⑤ Link Key

Include to or Exclude from Z-Wave™ Network

In the rear casing, there is a link key which is used to carry out inclusion, exclusion, association or reset. When the detector is first powered up, the LED flashes on and off alternately and repeatedly at 2-second intervals. It implies that it has not been assigned a node ID and cannot work with Z-Wave enabled devices. Please get familiar with the terms below before starting the operations.

Function	Description
Inclusion	Add a Z-Wave enabled device (e.g. Motion Detector) to Z-Wave network.
Exclusion	Delete a Z-Wave enabled device (e.g. Motion Detector) from the network.
Association	After inclusion, you have to define the relationship between devices. Trough association, device can be assigned as master/slave, and specify which slave is going to be controlled by which master.
Reset	Restore Detector to factory default.

The table below lists an operation summary of basic Z-Wave functions. Please refer to the instructions for your Z-Wave™ Certificated Primary Controller to access the setup function, and to include/exclude/associate devices.

Function	Description	Indication
No node ID	The Z-Wave Controller does not allocate a node ID to the unit.	2-second on, 2-second off
Inclusion	1. Have Z-Wave Controller entered inclusion mode.	Detector beeps when link key is pressed.
	2. Pressing link key 3 times within 1.5 second will enter inclusion mode. The Detector will stay "awake" for 10 minutes to allow time for setting and device status enquiring.	
Exclusion	1. Have Z-Wave Controller entered exclusion mode.	Detector beeps when link key is pressed.
	2. Pressing link key 3 times within 1.5 second will enter exclusion mode. The Detector will stay "awake" for 10 minutes to allow time for setting and device status enquiring.	
Reset	1. Press link key 3 times within 1.5 second.	Detector beeps when link key is pressed.
	2. Within 1 second, press and hold link key until beep stops.	A long beep is sounded for 5 seconds.
	3. IDs are excluded and all of preset value will be reset to factory default.	2-second on, 2-second off

Function	Description	Indication
Association	1. Have Z-Wave Controller entered association mode.	
	2. When pressing link key 3 times within 1.5 seconds will enter association mode.	Detector beeps when link key is pressed.
	3. There are two groupings – 1 and 2. Refer to Z-Wave's Grouping as described on page 4.	

※Including a node ID allocated by Z-Wave Controller means inclusion. Excluding a node ID allocated by Z-Wave Controller means exclusion.
※Failed or success in including/excluding the node ID can be viewed from the Z-Wave Controller.

Choosing a Mounting Location

The Motion Detector can be mounted either on a wall or under a ceiling. Before selecting a position for Motion Detector, the following points should be noted:

1. Do not position the detector facing a window/fan/air-conditioner or direct sunlight.

2. Do not position the detector directly above or facing any source of heat, e.g. fires, radiators, boiler etc.

3. Ensure the detector is positioned in place where the light source detected by the detector is consistent with actual ambient illumination. Do not locate the detector in a shadowy place.

4. Where possible, mount the detector so that the logical path of an intruder would cut across the fan pattern rather than directly towards the detector (FIGURE 1).

5. For best results, locate the detector directly facing an entrance.

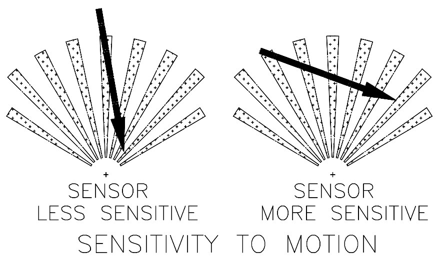


FIGURE 1

Installation

1. Undo and remove the screw from the bottom edge of the detector to detach the rear cover (FIGURE 2).

2. Unscrew the screw from the battery cover and remove the battery cover.

3. Insert 3 AA-size 1.5V alkaline batteries to the battery compartment, ensuring correct polarity is put (FIGURE 3).

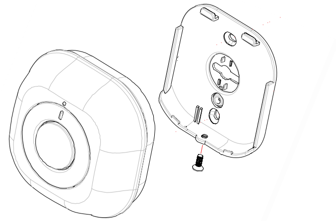


FIGURE 2

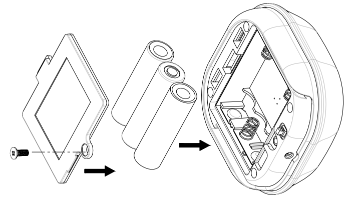


FIGURE 3

4. Two ways of mounting are applicable to the detector. Decide the detector is to be wall-mounted (FIGURE 4a) or ceiling-mounted (FIGURE 5a) based on the coverage angles shown in FIGURE 4b and FIGURE 5b. Hold the rear cover in position and mark the two mounting holes. Drill the holes, insert the plastic wall plugs and screw the rear cover to the wall or ceiling using the screws provided.

5. Engage the detector to the rear cover firmly.

(I) Wall Mounting

The recommended position for wall mounting is at the height of 1.8m (5.91 ft) from the floor. At this height, the optimum detection range is up to 10m (32.81 ft) with coverage range of 110 degrees (FIGURE 4b).

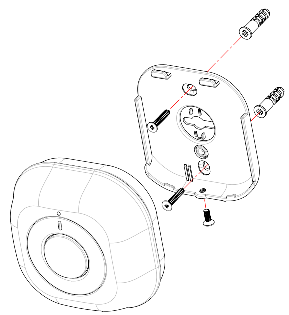


FIGURE 4a

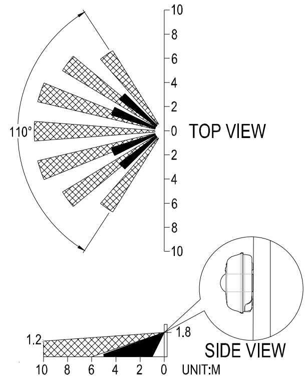


FIGURE 4b

(II) Ceiling Mounting

The recommended position for ceiling mounting is at the height of 2.8m (9.19ft) from the floor. At this height, the optimum detection range is up to 5m (16.41ft) with coverage range of 360 degrees (FIGURE 5b).

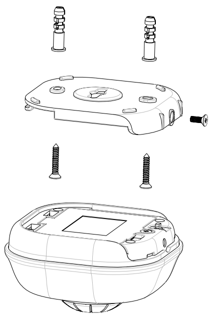


FIGURE 5a

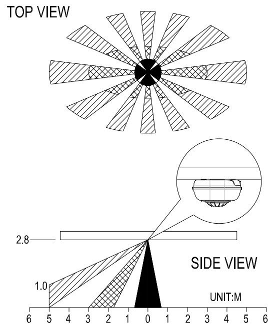


FIGURE 5b

Settings

Coverage Range Adjustments

Two types of lens covers are provided for the detector. Wall-lens cover (FIGURE 6a) is to be used when the detector is wall-mounted, whereas ceiling-lens cover (FIGURE 6b) is to be used when the detector is ceiling-mounted. The coverage range adjustment is only applicable to ceiling-lens cover; choose correct lens cover before mounting.

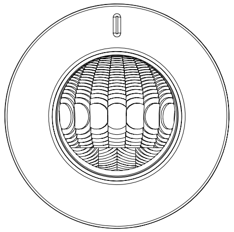


FIGURE 6a

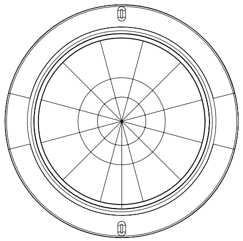


FIGURE 6b

The shading cap is composed of 12 segments for limiting the detection coverage, and each segment covers detection angle of 30 degrees (FIGURE 6c). Follow the grooves on the cap, cut the cap to a suitable size and place it onto the ceiling-lens cover (FIGURE 6d). The remaining segments are used for blanking off an undesirable detection area.

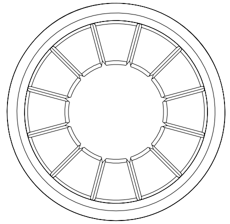


FIGURE 6c

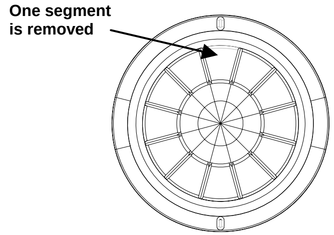


FIGURE 6d

Simply turn the cover anticlockwise to remove the wall-lens cover from the detector (FIGURE 6e).

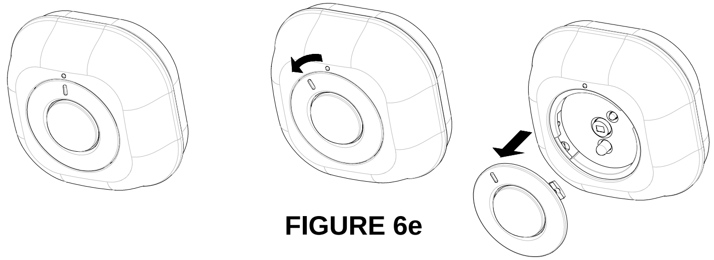


FIGURE 6e

Once the wall-lens cover is removed, reload the detector with ceiling-lens cover and turn it clockwise, ensure the mark on the cover is pointing towards and aligned with the mark on the detector (FIGURE 6f).

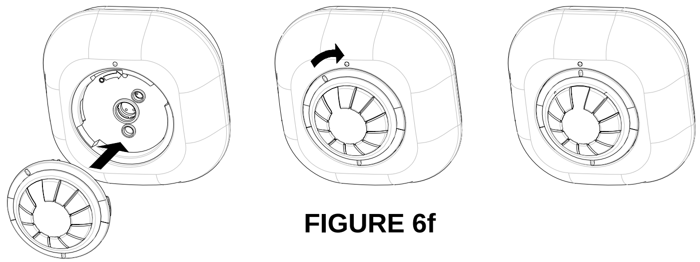


FIGURE 6f

Note: To detect movements with detection coverage up to 360 degrees, simply reload the ceiling-lens cover without shading cap. No movements can be detected if the detector is reloaded with a shading cap which maintains 12 lens segments.

Warm-Up

It will take approximately 2 minutes to warm up after battery has been connected. During this period, the detector beeps once every 3 seconds. When a long beep is sounded with red LED turns on steadily for 5 seconds, it implies warm-up procedure is completed and the detector is ready for detection.

Operation

Mounting location is a critical factor for deciding the type of lens to be used for the detector. Please decide whether the detector is going to be wall-mounted or ceiling-mounted before the operation procedure is carried on.

Wall Mounting

1. Place the wall-lens cover onto the detector.

2. By walking into a protected area within coverage of 110 degrees, the detector will now be triggered each time the detector senses movement. The orange LED on the detector will be illuminated and the associated appliances will be activated. For example, siren will be sounded or indication of movement detection will be shown on the controller. It implies that the unit is working properly.

Ceiling Mounting

1. Place the ceiling-lens cover (shading cap free) onto the detector.

2. By walking into a protected area within coverage of 360 degrees, the detector will now be triggered each time the detector senses movement. The orange LED on the detector will be illuminated and the associated appliances will be activated. For example, the siren will be sounded or indication of movement detection will be shown on the controller. It implies that the unit is working properly.

3. Place the shading cap onto the ceiling-lens cover.

4. Check whether same results can be gained by walking into a protected area within coverage that is at your disposal.

Programming

1. Z-Wave's Group (Association Command Class Version 2)

The unit supports two association groups with one node support for Grouping 1 and three nodes support for Grouping 2. This has the effect that when the unit is triggered, all devices associated with the unit will receive the relevant reports.

There are two kinds of reports: ALARM_REPORT and SENSOR_BINARY_REPORT.

1-1 Grouping 1 (Max. node = 1)

1-1-1 Power Applied Command

The unit will send ALARM_REPORT command to the nodes of Grouping 1 to inform the device that the unit is powered up.

ALARM_REPORT Command: **[Command Class Alarm, Alarm Report, Alarm Level = 0x02, Alarm Type = 0x01]**

1-1-2 Intrusion Event Report (Binary Sensor Report)

Once the Detector detected a movement, the unit will send SENSOR _ BINARY_REPORT to the nodes of Grouping 1 to inform there is an intrusion event. Once the movement is stopped, SENSOR_ BINARY_REPORT will be sent again to the associated devices.

BINARY SENSOR REPORT Command: Event Present: **[Command Class Sensor Binary, Sensor Binary Report, Value = 255 (0xFF)]** Event Clear: **[Command Class Sensor Binary, Sensor Binary Report, Value = 0 (0x00)]**

1-1-3 Low Battery Report (Alarm Report Class)

Upon Detector status being changed, the unit will check its battery status simultaneously. When the battery level of the unit drops to an unacceptable level, the unit will flash red LED once every 30 seconds, and emit ALARM_REPORT command to the nodes of Grouping 1.

ALARM_REPORT Command: **[Command Class Alarm, Alarm Type = 0x01, Alarm Level = 255(0xFF)]**

1-2 Grouping 2 (Max. node = 3)

1-2-1 Control other Z-Wave Devices

When the detector is triggered, the unit will send BASIC_SET command which contains a value that is adjustable, to the nodes of Grouping 2. For instance, the brightness level of a lamp module can be fixed according to the set value.

However, the BASIC_SET command will also be sent to the nodes of Grouping 2. For instance, a lamp module will be turned off after receiving the BAISC_SET command.

Basic Set Command: Event Present: **[Command Class Basic, Basic Set, Value = 255 (0xFF)]**
Event Clear: **[Command Class Basic, Basic Set, Value = 0 (0x00)]**

2. Z-Wave's Configuration

The following information is for someone that has some experience in setting up a Z-Wave system or someone that has computer software running a Z-Wave controller. Please get familiar with software of Z-Wave controller before getting started.

2-1 Basic Set Level

When Basic Set Command is sent where contains a value, the receiver will take it for consideration; for instance, if a lamp module is received the Basic Set command of which value is decisive as to how bright of dim level of lamp module shall be.

Example:
0: OFF
1-99: ON (Binary Switch Device)
Dim Level (Multilevel Switch Device)

Function	Parameter Number	Size	Range	Default
Basic Set level	1	1	0 ~99	99

Configuration Command

2-2 Enabling/Disabling Sensor Detecting Function

There might be times when users wish to suspend the detecting functions of the detector temporarily. By using Configuration Parameter #2, the detecting function can be set as enable or disable, where configured with the value of 0 means disable and 1 means enable.

Function	Parameter Number	Size	Value	Default
Enable/Disable Detecting	2	1	0 or 1	1

Configuration Command

Note: Reconnection of power supply will enable the sensor detecting function automatically.

2-3 Sensitivity Level (PIR sensor only)

In order to provide a best efficiency of the detector, it is recommended to test the detector with movements from a farthest end of the coverage area at first time of use. If movements cannot be detected sensitively, simply adjust the sensitivity level with Configuration Parameter #3. This parameter can be configured with the value of 1 through 10, where 1 means low sensitivity and 10 means highest sensitivity.

Function	Parameter Number	Size	Range	Default
Sensitivity Level	3	1	1~10	6

Configuration Command

2-4 Re-trigger Interval Setting (PIR sensor only)

The Configuration parameter that can be used to adjust the interval of being re-triggered after the detector has been triggered as Configuration Parameter #4. No response will be made during this interval if a movement is presented (FIGURE 7). The time interval can be set between 5 secs to 3600 secs.

Function	Parameter Number	Size	Range	Default
Re-trigger Interval	4	1 or 2 (if value > 127)	5~3600 (sec)	5

Configuration Command

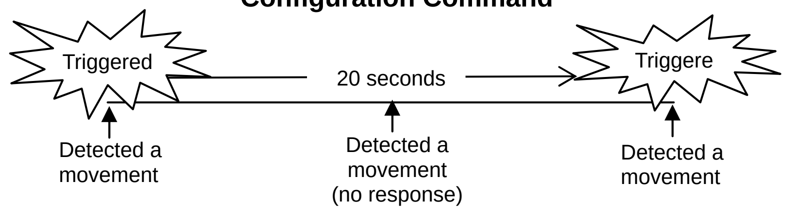


FIGURE 7

Note: The orange LED is on for one second when the detector detects a trigger.

2-5 Lux Level

The user can set a detecting percentage of lux level which determines when the light sensor will be activated. If percentage of lux level of ambient illumination falls below this percentage, and a person moves across or within the protected area, the detector will emit Z-Wave ON Command (i.e. Basic Set Command (Value = Basic Set Level)) to controller and activate connected modules and lighting. Percentage can be set between 1% to 100%.

Function	Parameter Number	Size	Range	Default
Lux Level	5	1	1~100 %	10

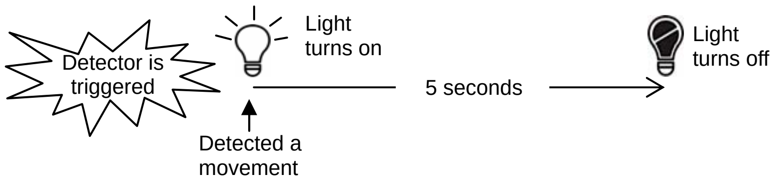
Configuration Command

2-6 On-Off Duration

The function of on-off duration setting will be useful if the detector is connected with a module or lighting. The duration determines how long the module/lighting should stay ON. For instance, Lamp Module turns off 100 secs after it has been turned on. This parameter can be configured with the value of 5 through 3600, where 5 means 5 second delay and 3600 means 3600 seconds of delay.

Function	Parameter Number	Size	Range	Default
On-Off Duration	6	1 or 2 (if value > 127)	5~3600 (sec)	5

Configuration Command



Note: The green LED will stay on for 1 second after 15 seconds of interval.

3. Advanced Programming

3-1 Battery Check Command

The users can also enquire the battery status of the Detector by sending BATTERY_ GET command via Z-Wave Controller. Once the unit receives the command, it will return BATTERY_REPORT command. If the unit is in low battery status, a Battery_Level = 255 (0xFF) command will be sent to the Z-Wave Controller.

BATTERY_REPORT Command
[Command Class Battery, Battery Report, Battery Level = 20%-100%]

3-2 Wakeup Command Class

The detector stays in sleep status for the majority of time in order to conserve battery life. However, it can be woken up by either triggers of movement or by setting WAKE_UP_INTERVAL_SET command via Z-Wave Controller. After the unit wakes up, it will send Wakeup Notification Command to the node ID that requires to be reported. The minimum and maximum wakeup interval is 60 seconds and 194 days respectively. Allowable interval among each wakeup interval is 1 second, such as 60, 61, 62 ….

Note: The default value is 1 hour, which implies that the detector awakes and sends the Wakeup Notification Command to the set node every hour.

Command Classes

The Motion Detector supports Command Classes including…

* COMMAND_CLASS_BASIC
* COMMAND_CLASS_BATTERY
* COMMAND_CLASS_VERISON
* COMMAND_CLASS_WAKE_UP_V2
* COMMAND_CLASS_ASSOCIATION_V2
* COMMAND_CLASS_CONFIGURATION
* COMMAND_CLASS_SENSOR_BINARY
* COMMAND_CLASS_MANUFACTURER_SPECIFIC

Troubleshooting

Symptom	Possible Cause	Recommendation
LED cannot be displayed	Run out of battery power	Replace a new battery
	Check if reverse battery polarity	Refit the battery with correct polarity
The detector not working	Check if mounting location is proper	Reposition its mounting location
		Remove the source of interference
	Check if the detector is out of order	Do not open the detector; send it to the local retailer.

Symptom	Possible Cause	Recommendation
Two minutes warm up is completed, but cannot hear long beep sound (LED flashes on & off repeatedly at 2-second intervals)	Check if detector is first power up or the detector has executed exclusion or reset procedure	Please carry out inclusion procedure; make sure there are ID codes stored in the detector.
The detector does not stay awake for 10 minutes	Check if detector is out of order	Please make sure link key is pressed 3 times within 1.5 sec. If detector still fails to stay awake for 10 seconds, repeat this step until it is succeeded.

Specifications

Battery	1.5V AA size x 3
Operating Range	Up to 30 meters line of sight (indoor)
Warm Up Time	About 2 minutes
PIR Detection Coverage	**Wall-Mounted:** Up to 10m x 110° (at 1.8m mounting height & 25°C) **Ceiling-Mounted:** Up to 5m x 360° (at 2.8m mounting height & 25°C)
Operating Frequency	868.42 MHz (SP814-1) / 908.42 MHz (SP814-2)
ZDK Version	V5.02

Specifications are subject to change without notice A501111536R

Federal Communication Commission Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

WARNING:

Do not dispose of electrical appliances as unsorted municipal waste, use separate collection facilities.

Contact your local government for information regarding the collection systems available.

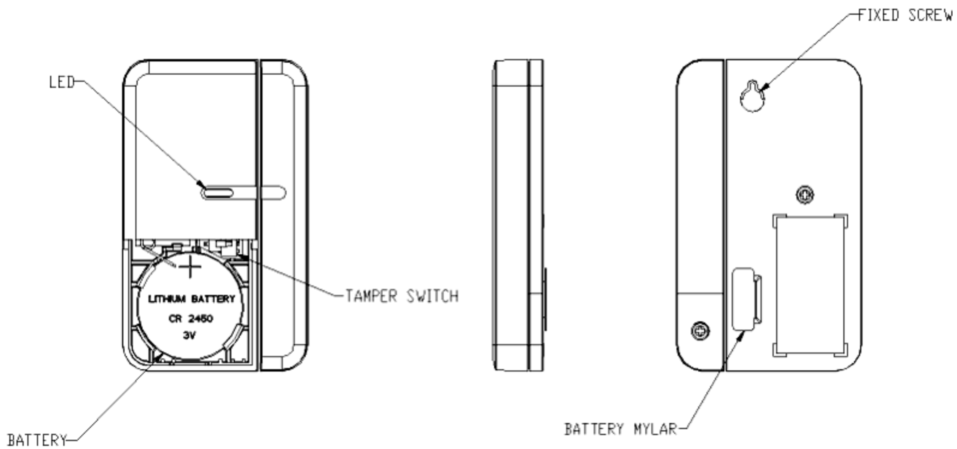
If electrical appliances are disposed of in landfills or dumps, hazardous substances can leak into the groundwater and get into the food chain, damaging your health and well-being.

When replacing old appliances with new once, the retailer is legally obligated to take back your old appliance for disposal at least for free of charge.

7.3. EVERSPRING HSM02 DATASHEET

HSM02 DOOR/WINDOW DETECTOR

The Door/Window Detector is a Z-Wave™ enabled device and is fully compatible with any Z-Wave™ enabled network. Z-Wave™ enabled devices displaying the Z-Wave™ logo can also be used with it regardless of the manufacturer, and ours can also be used in other manufacturer's Z-Wave™ enabled networks. Inclusion of this Door/Window Detector on other manufacturer's Wireless Controller menu allows remote turn-on of connected modules and their connected lighting when the Detector is triggered.



Include to or Exclude from a Z-Wave™ Network

In the front casing, there is a tamper switch which is used to carry out inclusion, exclusion or reset. When power is first applied, its LED flashes on and off alternately and repeatedly at 2-second intervals. It implies that it has not been assigned a node ID and cannot work with Z-Wave enabled devices. Please get familiar with the terms below before starting the operations.

Function	Description
Inclusion	Add a Z-Wave enabled device (e.g. Detector) to Z-Wave network.
Exclusion	Delete a Z-Wave enabled device (e.g. Detector) from the network.
Association	After inclusion, you have to define the relationship between devices. Trough association, device can be assigned as master/slave, and specify which slave is going to be controlled by which master.
Reset	Restore Detector to factory default.

The table below lists an operation summary of basic Z-Wave functions. Please refer to the instructions for your Z-Wave™ Certificated Primary Controller to access the setup function, and to include/exclude/associate devices.

Function	Description	LED Indication
No node ID	The Z-Wave Controller does not allocate a node ID to the unit.	2-second on, 2-second off
Inclusion	1. Have Z-Wave Controller entered inclusion mode.	
	2. Pressing tamper switch three times within 1.5 second will enter inclusion mode.	
Exclusion	1. Have Z-Wave Controller entered exclusion mode.	LED lights up once whenever tamper switch is pressed once.
	2. Pressing tamper switch three times within 1.5 second will enter exclusion mode.	
Reset	1. Press tamper switch three times within 1.5 second.	
	2. Within 1 second, press and hold the tamper switch until LED is off.	LED keeps on before reset function has been completed.
	3. IDs are excluded and all of preset value will be reset to factory default.	2-second on, 2-second off
Association	1. Have Z-Wave Controller entered association mode.	
	2. When pressing tamper switch three times within 1.5 second, the unit will emit the NIF which implies that the unit has entered association mode.	

※Including a node ID allocated by Z-Wave Controller means inclusion. Excluding a node ID allocated by Z-Wave Controller means exclusion.
※Failed or success in including/excluding the node ID can be viewed from the Z-Wave Controller.

Choosing A Mounting Location

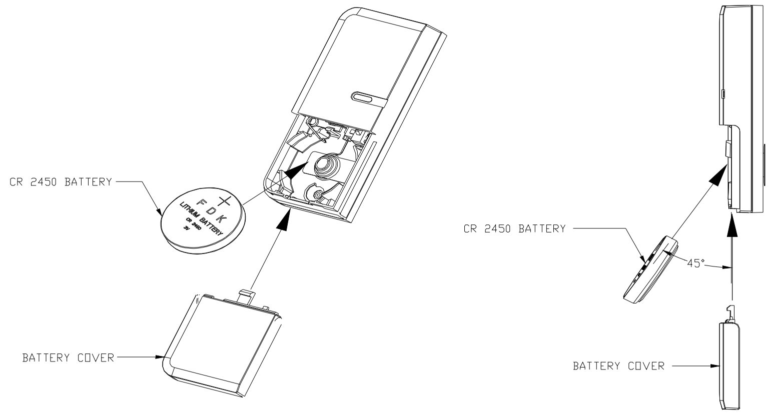
The Door/Window Detector is suitable for mounting in dry interior locations only.

Decide which doors/windows are to be protected by Door/Window Detectors, (usually the front and back doors as a minimum will have Door/Window Detectors fitted). Additional detectors may also be fitted where required to other vulnerable doors or windows, (e.g. garage, patio/conservatory doors etc).

Note: Take care when fixing the Detector to a metal frame, or mounting within 1m of metalwork (i.e. radiators, water pipes, etc) as this could affect the radio range of the device. If required, it may be necessary to space the magnet and detector away from the metal surface using a plastic or wooden spacer to achieve the necessary radio range.

Installation

1. Ensure that the system properly powered.
   Factory default built in a CR2450 battery inside the detector and uses a Mylar film to isolate battery from electric circuit of the detector. Remove the battery Mylar film when ready to let the detector work.
   If there is no battery inside the detector or need to replace a new battery please insert the battery in 45° angle as below figure



2. Using the adhesive tape to fit detector on the door or window.

3. Fit the magnet to the moving part of the door/window opposite the detector using the adhesive tape.

4. Ensure that the parallel gap between the magnet and detector is less than 20mm and that the matching line on the magnet is pointing towards and aligned with the line on the detector. An alarm condition will be occurred if the gap is greater than 35mm.

5. Remove the battery cover with the tamper switch not being pressed on the detector (test mode), detach or close the magnet from the Detector, the LED on the detector will illuminate.

6. After proper installation and test, put the battery cover back to the detector and the detector enters the normal mode.

Note: After removing batteries, wait for 5 seconds to refit batteries.

Operation

1. If first use of HSM02 with no node ID, LED will start twinkling for 30 sec. to lead the user for Inclusion. After HSM02 finishing Inclusion and enter sleeping mode, the unit will wake up by pressing Tamper and the user can see the LED start lighting up shortly every sec., currently the unit can receive set up for controller. After 30 sec., the unit will enter sleeping mode again, if set up is still needed, the user can press Tamper once more for HSM02 to be awake for another 30 sec.

2. Due to limited power output for CR2450, it can't continuously operate for a long time due to power consumption. Therefore, set up time for HSM02 should be minimized, and repeatedly press of Tamper should be avoided as well, in order to prevent unusual incident by a quick battery voltage drop down.

3. User can enter test mode by releasing or not pressing the Tamper SW, in the meantime if magnetic sensor is triggered then the LED will be illuminated. User can confirm whether the Tamper SW has been pressed properly by implementing this function. When Tamper SW is to be pressed and enter normal mode, LED will not be illuminated even if the magnetic sensor is

triggered, unless low battery is detected.

4. When the tamper switch is pressed, the unit enters normal mode and the red indicator LED on the Detector will not illuminate to conserve battery life when the detector is triggered, (unless the battery is low).

Programming

1. Z-Wave's Group (Association Command Class Version 1)

The unit supports two association groups. This has the effect that when the unit is triggered, all devices associated with the unit will receive the relevant reports.

Grouping 1 Support ALARM_REPORT, SENSOR_BINARY_REPORT
Grouping 2 Support BASIC_SET

Grouping 1 Application (Max. node=1)

Power Applied

Once the power has been applied, Alarm Report Command will be sent to Nodes in Grouping 1 to confirm the power applied status for HSM02.

Power Applied Notice:
[Command Class Alarm, Alarm Report, Alarm Level = 0x02, Alarm Type = 0x01]

1-1 Magnet status report: Binary Sensor Report

When the magnets of HSM02 are to be opened, Binary Sensor Report Command (Value = 0xFF) will be sent to Nodes in Grouping 1, and when the magnets are to be closed, Binary Sensor Report Command (Value = 0x00) will be sent as well.

Magnets to be opened:
[Command Class Sensor Binary, Sensor Binary Report, Value = 0xFF(255)]
Magnets to be closed:
[Command Class Sensor Binary, Sensor Binary Report, Value = 0x00(0)]

1-2 Low Battery Report

When HSM02 automatically wakes up, it will check the battery usage. When low battery is detected, Alarm Report Command will be sent to Nodes in Grouping 1, afterward, LED will light up for 1 sec to remind user when HSM02 is triggered due to open or close incidents.

Low Battery Report :
[Command Class Alarm, Alarm Report, Alarm Type = 0x01, Alarm Level = 0xFF]

1-3 Tamper Event Report (Alarm Report)

Press and hold the tamper switch more than 10 seconds then release, the unit will send ALARM REPORT command to the nodes of Grouping 1 to inform them there is a tamper event.

Tamper Event Report :
[Command Class Alarm, Alarm Report, Alarm Type = 0x01, Alarm Level = 0x11]

1-4 Control other Z-Wave Devices

When door/window is opened, the unit will send BASIC SET command which contains a value that is adjustable, to the nodes of Grouping 2. For instance, the brightness level of a lamp module can be fixed according to the set value.

Grouping 2 Application (Max. node=5)

However, the BASIC_SET command will be also sent to the nodes of Grouping 2. For instance, a lamp module will be turned off after receiving the BASIC_SET command.

Basic Set Command: Event Present: **[Command Class Basic, Basic Set, Value = 255 (0xFF)]**
Event Clear: **[Command Class Basic, Basic Set, Value = 0 (0x00)]**

2. Z-Wave's Configuration

2-1 Basic Set Level

When Basic Set Command is sent where contains a value, the receiver will take it for consideration; for instance, if a lamp module is received the Basic Set command of which value is decisive as to how bright of dim level of lamp module shall be.

Example:
1-99: ON (Binary Switch Device)
Dim Level (Multilevel Switch Device)

Function	**Parameter Number**	**Size**	**Range**	**Default**
Basic Set level	1	1	1~99	99

Configuration Command

2-2 Configuring the OFF Delay

The Configuration parameter that can be used to adjust the amount of delay before the OFF command is transmitted as Configuration Parameter #2. This parameter can be configured with the value of 0 through 127, where 0 means send OFF command immediately and 127 means 127 seconds of delay.

Function	**Parameter Number**	**Size**	**Range**	**Default**
Basic Set level	2	1	0~127	0s

Configuration Command

3. Advanced Programming

The following information is for someone that has some experience setting up a Z-Wave system or someone that has computer software running a Z-Wave controller.

3-1 Battery Check Command

The users can also enquire the battery status of the unit by sending BATTERY_GET command via Z-Wave Controller. Once the unit receives the command, it will return BATTERY_REPORT command. The unit will send Battery_Level = 255 (0xFF) command to the Z-Wave Controller to inform that the unit is in low battery status.

BATTERY REPORT Command: **[Command Class Battery, Battery Report, Battery Level = 20%-100%]**

3-2 Wakeup Command Class

The unit stays in sleep status for the majority of time in order to conserve battery power. However, it can be woken up at specified intervals by setting WAKE_UP_INTERVAL_SET command by Z-Wave Controller. After the unit wakes up, it will send Wakeup Notification Command to the node ID that requires to be reported and stay awake for 5 seconds if no WAKE_UP_NO_MORE_INFORMATION command is received. The minimum and maximum wakeup interval is 60 seconds and 194 days respectively. Allowable interval among each wakeup interval is 1 second, such as 60, 61, 62 ….

Note: The default value is 1 day, which implies that the detector awakes and sends the Wakeup Notification Command to the set node every day.

4. Factory Default Setting

Command	**Default setting**
Basic Set level	99
Period of Wake Up Notification	1 day

5. Command Classes

The Door/Window Detector supports Command Classes including…

*COMMAND_CLASS_ALARM
*COMMAND_CLASS_SENSOR_BINARY
*COMMAND_CLASS_CONFIGURATION
*COMMAND_CLASS_WAKE_UP
*COMMAND_CLASS_MANUFACTURER_SPECIFIC
*COMMAND_CLASS_VERSION
*COMMAND_CLASS_ASSOCIATION
*COMMAND_CLASS_BATTERY
---For Control Other Devices---
*COMMAND_CLASS_BASIC

Troubleshooting

Symptom	Possible Cause	Recommendation
Cannot carry out inclusion and association	Included a node ID allocated by other Z-Wave Controller.	Exclude a node ID then carry out inclusion and association with new Controller.
	Does not fit batteries or run out of battery power.	Check if batteries are fitted or replace a new battery.
LED not illuminating and not working	Does not fit batteries or run out of battery power.	Check if batteries are fitted or replace a new battery.
	Break down	Send it for repair and do not open up the unit.

Specifications

Battery	CR2450 3.0V 620mAh Lithium Battery
Range	Minimum 30 m line of sight
Frequency Range	908.42 MHz (US) / 868.42 MHz (EU)

Specifications are subject to change without notice

Federal Communication Commission Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

WARNING:

Do not dispose of electrical appliances as unsorted municipal waste, use separate collection facilities.

Contact your local government for information regarding the collection systems available.

If electrical appliances are disposed of in landfills or dumps, hazardous substances can leak into the groundwater and get into the food chain, damaging your health and well-being.

When replacing old appliances with new once, the retailer is legally obligated to take back your old appliance for disposal at least for free of charge.