| | |
|---|---|
| Project Number: | **297178** |
| Acronym: | **FATE** |
| Title: | **Fall Detector for the Elder** |
| Call (part) identifier: | CIP-ICT-PSP-2011-5 |
| Start date: | 01/03/2012 |
| Duration: | 36 months |

# D2.1 FATE complete system prototype

Nature[1]: P
Dissemination level[2]: PU
Due date: Month 10
Date of delivery: Month 10
Partners involved (leader in bold): **UPC**, FLOWLAB, MFKK, GEMA
Authors: Marc Arnal (FLOWLAB), Cristian Barrué (UPC), Tamás Csielka (MFKK), Juan José Guirao (UPC), J. Manuel Moreno (UPC), David Navarro (FLOWLAB), Carlos Pérez López (UPC), Jaume Romagosa (UPC), Israel Serrano (GEMA)

[1] R = Report, P = Prototype, D = Demonstrator, O = Other

[2] PU = Public, PP = Restricted to other programme participants (including the Commission Services), RE= Restricted to a group specified by the consortium (including the Commission Services), CO = Confidential, only for members of the consortium (including the Commission Services)

# Revision history

| Date | Partner | Description | Name |
|---|---|---|---|
| 16/11/2012 | UPC | Index proposal | J. Manuel Moreno |
| 10/12/2012 | UPC | UPC and MFKK contributions updated | J. Manuel Moreno |
| 12/12/2012 | UPC | UPC contribution on i-Walker updated | Cristian Barrué |
| 12/12/2012 | GEMA | GEMA contribution updated | Israel Serrano |
| 21/12/2012 | UPC | FATE use cases included | J. Manuel Moreno |
| 03/01/2013 | FLOWLAB | FLOWLAB contribution included | Marc Arnal |
| 07/01/2013 | UPC | Final review | J. Manuel Moreno |
| 16/01/2013 | UPC | Final approval | J. Cabestany |

# DISCLAIMER

The work associated with this report has been carried out in accordance with the highest technical standards and the FATE partners have endeavoured to achieve the degree of accuracy and reliability appropriate to the work in question. However since the partners have no control over the use to which the information contained within the report is to be put by any other party, any other such party shall be deemed to have satisfied itself as to the suitability and reliability of the information in relation to any particular use, purpose or application.

Under no circumstances will any of the partners, their servants, employees or agents accept any liability whatsoever arising out of any error or inaccuracy contained in this report (or any further consolidation, summary, publication or dissemination of the information contained within this report) and/or the connected work and disclaim all liability for any loss, damage, expenses, claims or infringement of third party rights.

# Glossary

**BAN:** Body Area Network
**SDP:** Service Discovery Protocol
**HDP:** Health Device Profile
**SPP:** Bluetooth Serial Port Profile
**TSN:** Transaction Sequence Number
**ZCL:** ZigBee Cluster Library
**GPS**: Global Position System
**SMS**: Short Message Service
**XML**: eXtensible Markup Language
**XSD**: XML Schema Definition
**USB**: Universal Serial Bus
**RF**: Radio Frequency
**GSM:** Global System for Mobile Communications
**3G**: 3rd generation of mobile telecommunications technology
**GPRS:** General Packet Radio Service

# List of figures

# List of tables

# Table of contents

# 1. Introduction

This document provides a detailed description of the software and hardware components that constitute the FATE system to be used during the pilots. The structure of the document is as follows: First the overall description of the FATE system will be provided, so as to have a clear view of the different components that constitute it. Then the functional details of these components will be explained. Thereafter the use cases of the system will be described, so that it should become clear which are the main goals of the services provided by the FATE system, as well as the corresponding user interaction. Finally, the typical use cases for the FATE system will be explained.

# 2. Description of the FATE system

The overall architecture of the FATE system when used at user's home is depicted in Figure 1.

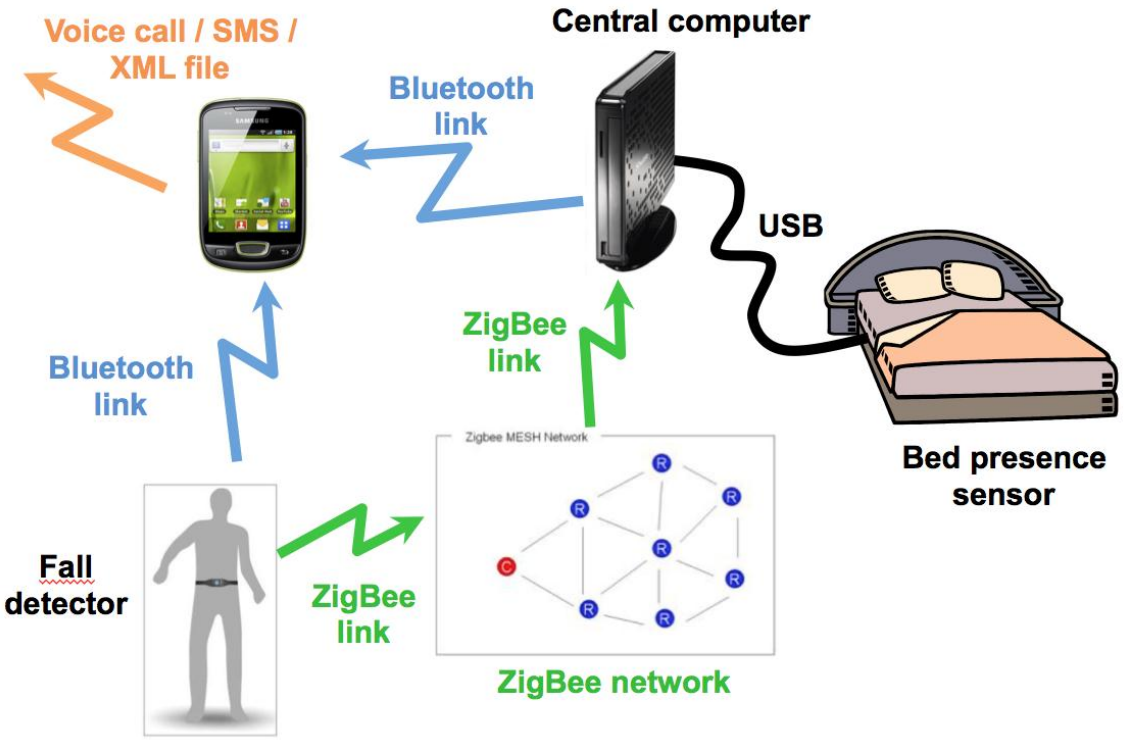



**Figure 1.** Overall architecture of the FATE system at user's home.

The core of the system is constituted by a highly sensitive fall detector that is built using accelerometers, a processing unit and a communications module. The processing unit of the fall detector runs continuously a dedicated on-line detection algorithm that is able to detect a user fall. When this event rises and the user is at home a specific alarm message is sent using a ZigBee communication link that arrives to a central computer through a dedicated ZigBee network.

The goal of the ZigBee network is to permit the detection of falls in any place at user's home. This network is coordinated by the central computer and is supported by a set of ZigBee wall routers that are distributed at specific locations in the home, so as to guarantee enough coverage.

Another component of the FATE system is the bed presence sensor. The goal of this sensor is to detect eventual falls during the user sleep period, when the fall detector is not worn. The sensor is sending continuously messages to the central computer using a USB link. These messages indicate if the user is present on bed or not. Once the user goes to bed, if he/she leaves it and does not come back for a specific time window then this situation may be interpreted as a possible fall.

The central computer receives the messages sent by the fall detector and the bed presence sensor and stores them locally for validation and analysis purposes. Additionally, it relays them to the user's mobile phone using a Bluetooth link. A specific application running in the mobile phone analyses these messages in real time and decides if a fall alarm has to be sent. There are three alarm formats in the FATE system, as requested by the call centres in each pilot site:

- An automatic voice call. This is the format to be used in the Italian pilot.
- An automatic voice call plus an XML file containing information about the user status. This is the format that will be used in the Spanish pilot.
- An SMS containing information about the user status. This is the format that will be used in the Irish and the Italian pilots.

The mobile phone can also send other types of messages indicating specific situations that may arise while using the FATE system (automatic or manual fall recovery or low battery status, among others). For a detailed explanation of the functionality of the application running in the mobile phone and the different types of messages that can be send, please refer to section 3.8 of this document.

When the user leaves home (a situation that is detected by the fall detector by the absence of a ZigBee network) the fall detector establishes a permanent link with the mobile phone using the Bluetooth protocol. This link contains the same information that is sent at home using the ZigBee network. In the case the user leaves home the mobile phone will include in the messages sent to the call center (SMS or XML file) information related to the position where a specific event has been detected.

For the FATE system installed at nursing homes the ZigBee and Bluetooth communication modules of the fall detector are replaced by an RFID tag. This tag communicates with the location solution provided by Gema Active Business Solutions, so that in the case a fall is detected the responsible personnel at the nursing home identify the place where it happened. In the same way, the bed presence sensor does send messages through a USB link, but through an RFID tag. The details corresponding to the FATE system to be used in nursing homes can be found in sections 3.2 and 3.6.7 of this document.

Additionally, the FATE system will contain another component for the pilots at nursing homes. It is the i-Walker, a robotic rollator developed by UPC that is based on a standard walker's frame and enhanced with sensors (6 force sensors, dual axis accelerometer and odometer), active motors and a processing unit. The i-Walker will be used as a technical aid to improve gait and balance, and therefore reduce the number of future falls. Section 3.7 of this document will provide the technical details corresponding to the i-Walker subsystem.

# 3. Components of the FATE system

This section provides a detailed functional description of the components that constitute the FATE system, as presented in section 2. First the hardware components of the system will be reviewed, and thereafter the software modules developed specifically for the FATE system will be analysed.

## 3.1. Fall detector to be used at home

The main directive in the sensor design process was to create a device with reasonable autonomy while maintaining a small physical size, allowing the user to wear the sensor with little difficulty. The algorithms implemented in the sensor have been developed assuming that it will be placed on the patient's waist on either hip to make it more comfortable to wear. The system components are restricted to comply with reasonable power consumption and a workable size; thus, the system is provided with the elements required for a device easy to wear during activities of daily life.

Figure 2 shows the overall organisation of the fall detector. It contains a microcontroller (PIC24F) that manages the components while processing on-line the data provided by the sensor (3D accelerometer). It is able to acquire data from the sensor and process it by means of the implemented algorithms while managing the energy supervision module, the user interface and the communication units.

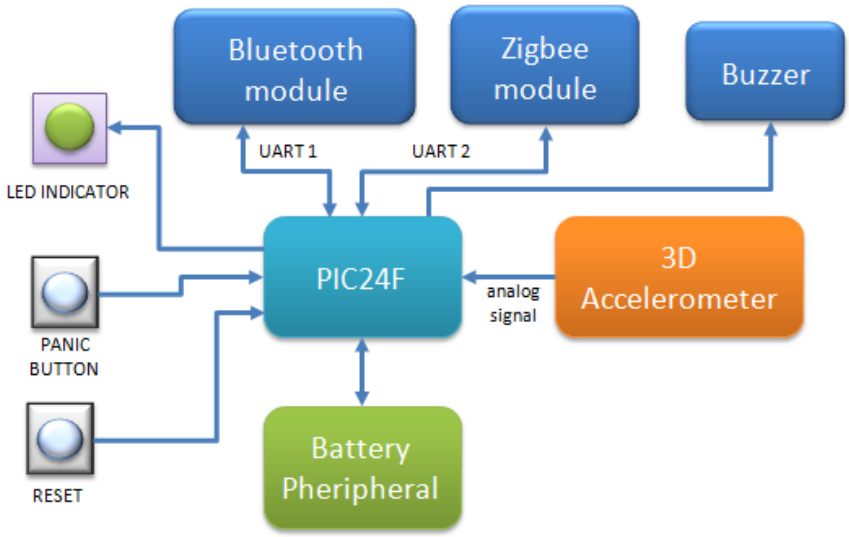



**Figure 2.** Internal organisation of the fall detector.

The system includes the classic elements of an Inertial Measurement Unit (IMU) and a control system dedicated to the battery and to optimise the energy consumption. The states of both the battery level and the main application process are shown to the user by using a very simple user interface constituted by a composite LED. A switch allows the user to interact with the device at any time, and a reset button to return to the initial state in case of system failure. In addition the system incorporates two communication modules, Bluetooth and ZigBee, and an alarm system activated by a buzzer.

The accelerometer used is the LIS344ALH from STMicroelectronics. It includes the sensing units in three axis in a single die, and therefore provides a single source for all the readings, thus simplifying the calibration of the measurements. It is configured so that the output is between ± 6g. The bandwidth for the sensor is 1.8 KHz, however, a low-pass, capacitor-based 100 Hz filter is inserted in order to eliminate the intrinsic noise of the sensor. The data provided by the accelerometer will permit the microprocessor to analyse the user movements at any time, so as to detect when a fall has occurred.

The inertial sensor is powered by a standard commercial (S EZPack Varta), Li-Polymer battery of 610 mAh. The inertial sensor has a connector to charge the battery specifically designed for battery EZPack S.

The system includes a module to manage flexibly and efficiently the battery, which leads to significant energy savings. Individual controllers were added, so that in this way it is possible to control the operation of each part of the system, eliminating unnecessary power consumption. The inertial sensor includes five regulators: the first, which is always active, feeds the main circuit. The other four regulators are directly controlled by the microprocessor, feeding the buzzer module, the analog section, the ZigBee unit and the Bluetooth unit. In this way and according to the state of the algorithm, the modules are enabled or disabled depending on the need of the system. Just the buzzer for example consumes more than the system microcontroller, analog section and communication units. It is therefore not advisable to be continually feeding this module.

The power management subsystem includes a module for controlling the charge level of the battery, thus the user can be informed when the device must be recharged. The power management algorithm shuts down the entire system when it detects a critical energy level, thus preventing data loss. Figure 3 shows the internal organisation of the power management module.

The Bluetooth communication unit is based on the embedded RF communication module Bluegiga WT12. It is an independent subsystem connected to the microprocessor via the UART serial communication controller. The module operates in the 2.4 GHz Industrial Scientific and Medical (ISM) frequency band. The Bluetooth module is configured in bridge mode, so that it sends all the data received through the serial port. Section 3.6.3 will present the Bluetooth protocol that has been specifically developed for the FATE system.

The ZigBee communication unit is based on the embedded ZigBee communication module Digi Xbee RF. It is an independent subsystem connected to the microprocessor via the UART serial communication controller. The module operates in the 2.4 GHz Industrial Scientific and Medical (ISM) frequency band and provides reliable data delivery with minimum power consumption.

The ZigBee module receives frames serially, extracts the data and builds the frame of RF according to the IEEE 802.15.4 standard. This new frame is sent by the ZigBee mesh network to the destination node. The same applies when data is received from the RF communications. The messages are taken from the ZigBee frames and sent to the microcontroller via serial. Section 3.6.4 will present the ZigBee protocol that has been specifically developed for the FATE system.

The user interface is constituted by a LED, an action button (panic button in Figure 2) and a reset button. The LED informs the user at any time about the status of the fall detector. The

action button permits the user to turn on the fall detector, to cancel an alarm or to generate an alarm at any time, even if a fall has not been detected. The reset button permits the user to turn off the fall detector. A detailed explanation of the user interface will be provided in section 4.

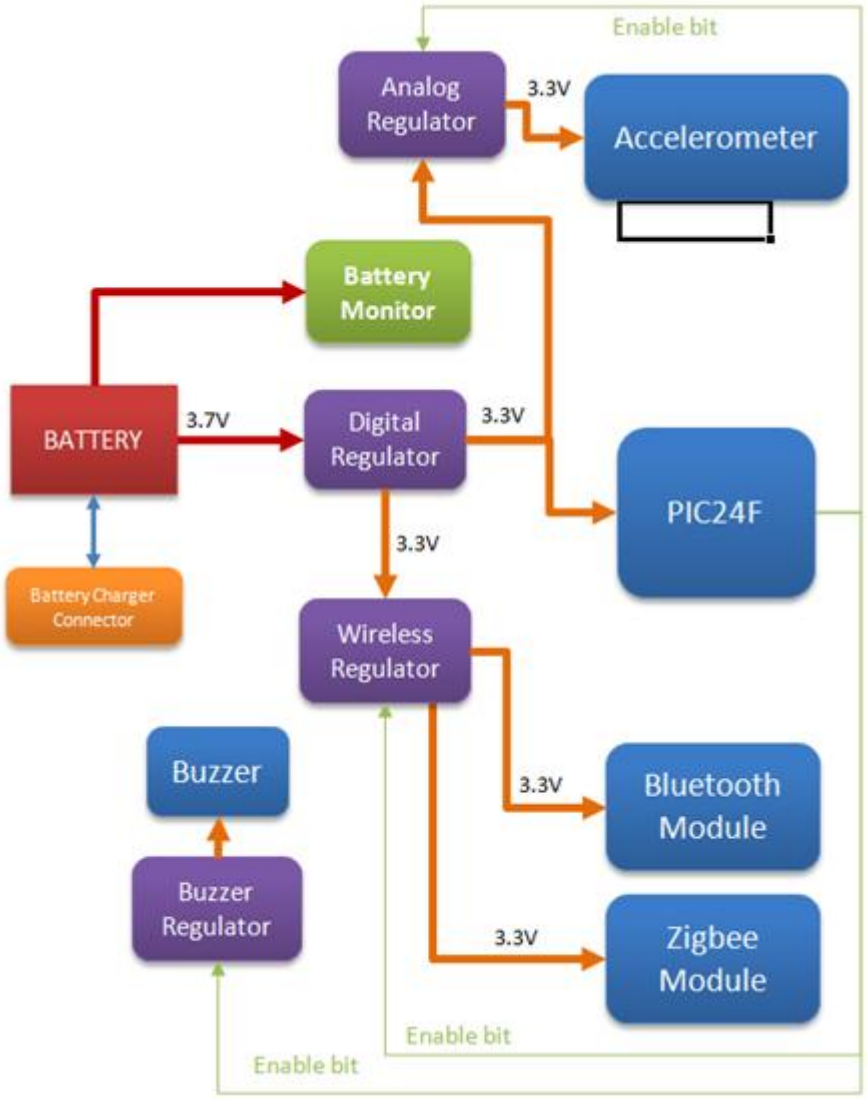



**Figure 3.** Organisation of the power management unit.

The fall detector case has been designed bearing in mind the following requirements:

- Male or female user.
- The detector must be worn all day long, excluding the night sleep interval.
- The user will take out the fall detector before going to sleep, and then it must be connected to the charger.
- The user must wear the fall detector on the waist, so as to guarantee a correct functional operation.
- The sensor must be placed so that the LED is in upright position, facing the user, thus guaranteeing that it is visible all the time.

Figure 4 shows the case developed for the fall detector.

**Figure 4.** Case of the fall detector.

The fall detector must be used with the neoprene belt provided with the system. The fall detector has to be placed inside the neoprene belt, which guarantees that the detector is in close contact with the user's skin. Figure 5 shows the placement of the fall detector in the belt and how the belt has to be worn.

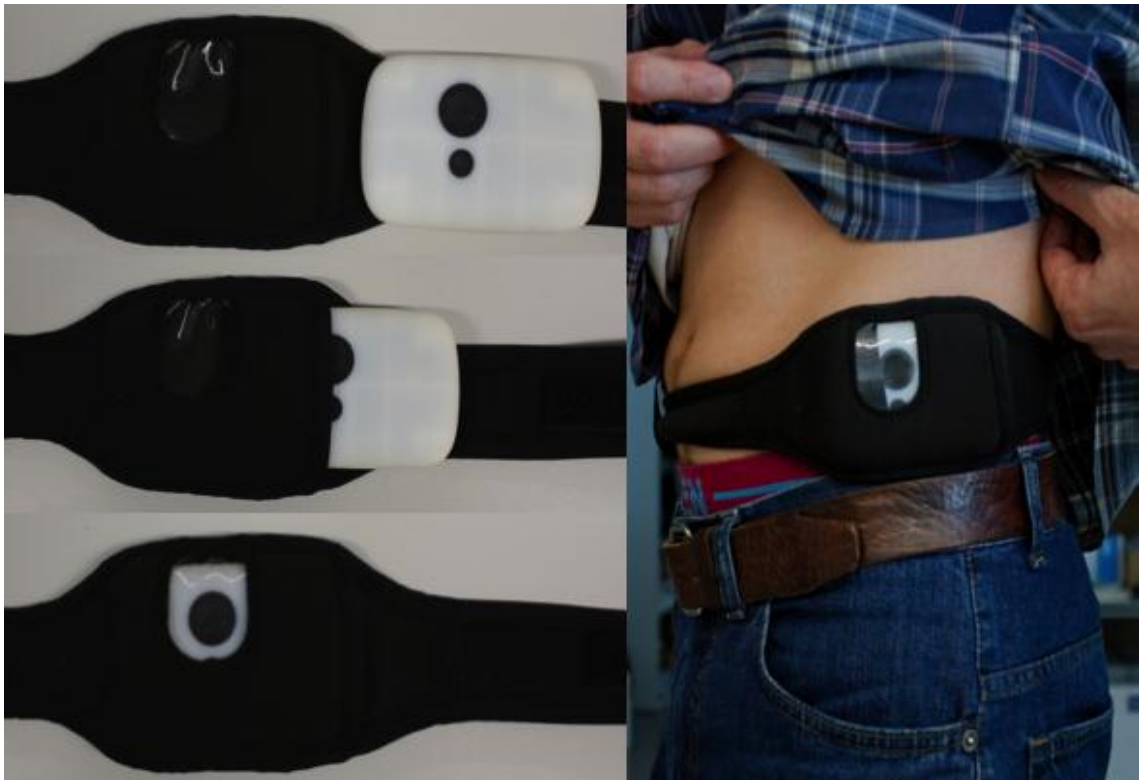


**Figure 5.** Fall detector placement and belt adjustment.

The characteristics of the neoprene material used to construct the belt are summarised in Figure 6.

SEDO
Chemicals Neoprene GmbH

# PRESS MOLDED BUNS
## Data Sheet

| | |
|---|---|
| Type | S |
| Polymer | Neoprene (CR) |
| Colour | black |
| ASTM D-1056-00 Classification | 2C1 / 2 |
| 25% Compression Deflection[1] [kPa] | 25 - 45 |
| 50% Compression Set[2] [ % ] | 25 max. |
| Density[3] [kg/m³] | 170 ± 30 |
| Durometer[4] [shore 00] | 35 - 43 |
| Impact Resilience[5] [%] | 70 min. |
| Shrinkage[6] [% linear] | 12 max. |
| Tensile Strength[7] [N/mm²] | 0,60 min. |
| Elongation at brake[8] [%] | 200 min. |
| Standard Sheet Size [cm] ±5% | 215 x 130 |
| Standard Bun Thickness [mm] ±5% | 30 |

| | | |
|---|---|---|
| 1) | 25% Compression Deflection | ASTM D-1056 |
| 2) | 50% Compression Set | ASTM D-1056 |
| 3) | Density | DIN EN ISO 845 |
| 4) | Durometer | 30 mm Thickness, skin both sides |
| 5) | Impact Resilience | DIN 53 512 |
| 6) | Shrinkage | 20 min. / 160°C, 30 min. |
| 7) | Tensile Strength | DIN 53 504, Normstab S 2, oc both sides |
| 8) | Elongation at brake | DIN 53 504, Normstab S 2, oc both sides |

Issued 01/05/09

**SEDO Chemicals Neoprene GmbH**
**Tränkeweg 18 a, D – 15517 Fürstenwalde**
**Phone: + 49 (3361) 59 65 20**
**www.sedochemicals.de**

**Figure 6.** Characteristics of the neoprene used to construct the belt.

## 3.2. Fall detector to be used at nursing homes

In the case of FATE project at nursing homes, the fall detection is integrated with location and identification solution provided by Gema Active Business Solutions. That means that are integrated in one device the fall detector and the RFID Tag. This Tag is the responsible to communicate the alarm generated when a fall is detected. In conclusion, all the aspects mentioned in the description of the fall detector for homes, applies in the case of nursing homes with the follows differences:

- The ZigBee and Bluetooth communication modules of the fall detector aren't needed in this device.
- The fall detector is connected to the Tag and integrated in one device.

Figure 7 shows the overall organisation of the fall detector in this case.

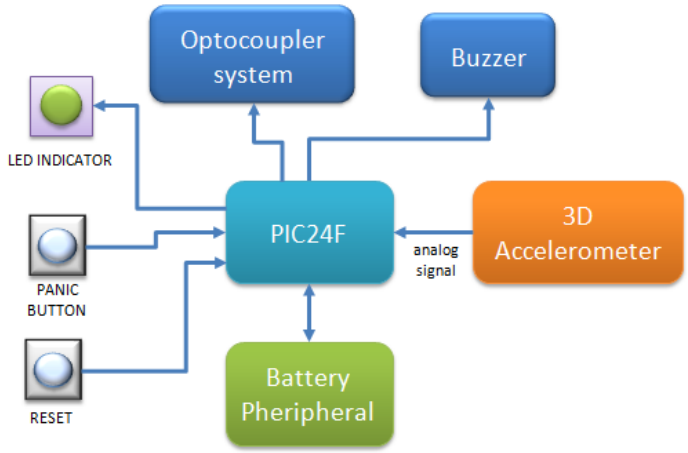



**Figure 7.** Internal organisation of the fall detector used in nursing homes.

The system incorporates an optocoupler system for communication between the fall detector (UPC board) and the RFID Tag (Visonic board). The mean reason to use this communication system is to obtain the maximum electrical isolation between the fall detector and the Tag. Figure 8 shows the general scheme of the integration.

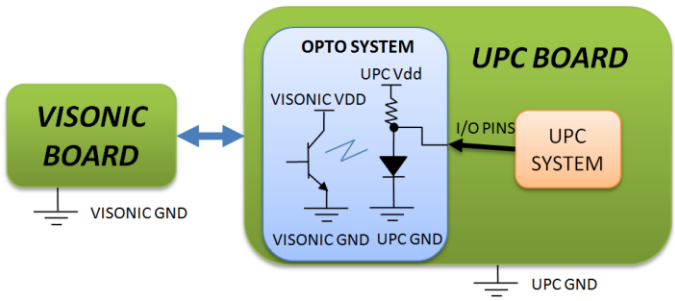



**Figure 8.** Overall scheme of fall detector and RFID tag integration.

Figure 9 shows the internal organization of the power management module for the fall detector in this case.

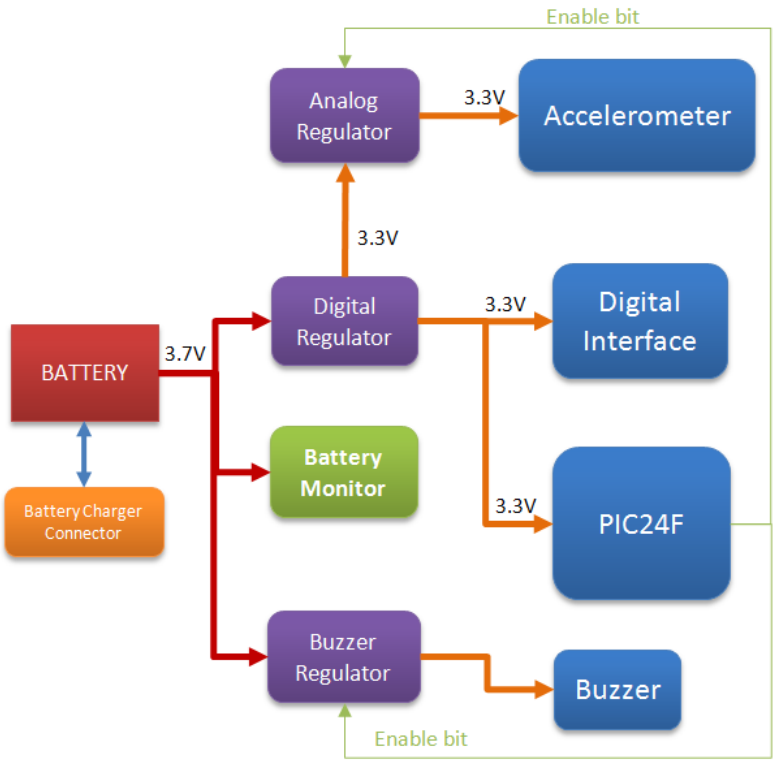



**Figure 9.** Organisation of the power management unit for the fall detector with RFID tag.

In the case of the Tag, the power source is a standard commercial replaceable on-board lithium battery (type CR2430) of 3V, 270mAH and an estimated duration about five years.

## 3.3. Bed presence sensor

As a bed presence sensor, an Ibernex product (NX0310 and N0321) has been chosen. See Figure 10.

The bed presence sensor has been designed to automatically inform of risk situations for users during sleeping/resting hours. The sensor detects if a user has not been in bed or has not got out of bed for a certain time, ensuring security and tranquillity for system users and for their caregivers.

This product is comprised of:

- **NX0310 Bed Presence Sensor**: It detects the user's presence or absence in the bed/chair. It uses very sensitive piezoelectric sensors that detect slight movements of the body on the bed during sleep (breathing, heart rate, etc.).

- **NX0321 Bed Sensor Concentrator**: 1 concentrator per bed; it admits up to two sensors for beds wider than 90 cm. It is connected by radiofrequency (RF), USB and wired signal, respectively.

These devices work together, avoiding risk situations for users, informing if they get out of bed, wander around or fall.

The bed presence sensor is placed underneath the user's mattress (on the slats of the bedframe)

or underneath the seat of the chair, and is completely undetectable. It monitors the presence or absence of a person and is connected to the concentrator.

In the event of risk situations for the user, the sensor concentrator generates a warning that it immediately sends to the remote relay, to the USB serial port or to the wired signal. The solution generates an alarm status if the user gets out of bed and does not return to it in a certain time interval, in order to identify problems as quickly as possible, thus helping care-givers manage risk situations quickly.

**Figure 10.** Bed presence sensor (left) and concentrator (right).

### 3.3.1. Technical specifications of sensor NX0310

- High sensitivity piezoelectric sensor.
- Detection of falls and presence/absence in bed.
- Monitors the vibrations produced even by involuntary movements (breathing, heartbeat etc.).
- Connection to NX0320 module by 3m cable.
- Undetectable by the user: very fine sensor that is placed under the mattress, on the base (slats or rigid).
- Independent of the type of mattress or base.
- Only one sensor can be used in single beds, situated in the centre of the bed, at chest height.
- Two sensors must be used in double beds, each one situated in the centre of the area of each user, at chest height.
- Secured to the base or slats of the flat frame with double-sided adhesive tape.
- The method used to secure it must not add pressure to the sensor surface.
- Supplied from the concentrator module (3.3 Vdc voltage).

### 3.3.2. Technical specifications of concentrator NX0321

- It can work with 1 or 2 NX0310 bed sensors; if 2 sensors are installed, they must both be in the same bed.

- Power supply (power source included)
  - 12Vdc nominal supply, 3.5Vdc minimum, 15.5Vdc maximum.
  - 9mA typical consumption, 45mA maximum.
  - DC power type connector, 5.5 mm outer diameter, 2.1 mm inner diameter.

- Two-coloured illuminated pushbutton to indicate status, transmit message to alarm system and temporary disablement.

- Configurable pushbutton luminosity.

- Buzzer for pulsation feedback and to notify abandonment of bed. It can be set to work in silent mode. 8 micro switches on the interior for configuring parameters (times, sounds, pushbutton luminosity level, etc.).

- USB communication: see section 3.3.4.

- Disconnected sensor detection.

- Press button for temporary disablement, to lift the user or make the bed without triggering the alarm. The sensor continues to monitor presence or absence but does not generate any action.

- Configurable arming time to avoid false alarms.

- Configurable courtesy time to permit temporary absence without triggering alarm.

- Message transmitted to room terminal (or to remote relay) by extra-long press of the button, to facilitate installation and, optionally, to trigger alarm.

- Dimensions: 90x55x25mm.

- Box with a removable base with screws on the cover that are hidden after installation. The base can be easily secured to the wall, headboard or bed structure with screws, flanges or adhesive.

### 3.3.3. Operation mode

The NX0310 sensor contains several piezoelectric elements that convert vibrations into electrical signals and the necessary electronics to amplify and filter these signals. The sensor does not detect weight or pressure, but the vibrations generated by movements, with sufficient sensitivity so as to be activated with the heartbeat and the breathing of the user through the mattress.

During normal operation, the pushbutton launches red coloured flashes if the user is not detected in the bed, and green if the user is detected. It is normal for a few seconds to elapse before the sensor detects the absence or presence of the user. These delays are essential to

prevent false alarms from being triggered. Certain actions may cause vibrations in the area of the bed, such as making the bed, moving it, moving the guard rails, leaving objects on the bed, etc. that may cause a false detection of presence. The use of arming times and temporary disablement permits these actions without generating false alarms.

If a presence is detected for less time than the "arming time", the sensor considers that the user is not in bed. The user's presence must be maintained during the "arming time" (configurable) so that the subsequent detection of absence will cause the triggering of warnings of bed abandonment. Once armed, the absence detection can cause the immediate transmission of the bed abandonment warning, setting the "courtesy time" to 0 seconds. The bed abandonment warning transmission can trigger an alarm in the system (depending on the configuration of the alarm system).

If the "courtesy time" is not nil, the warning transmission is postponed for the set time. This is the time that the user has to be able to abandon the bed without triggering warnings. To prevent a warning from being triggered, during the "courtesy time" it is possible to return to the bed to change to presence status (with the system already armed), or press the button of the concentrator to change to the temporary disablement mode. When the temporary disablement status is exited, it will change to absence or presence status, but without triggering a warning.

The bed concentrator button can be pressed at any time to activate and deactivate the temporary disablement status. Even though the button is not pressed again, the system always exits the disablement status after a previously established time interval. The sensor continues to work during the temporary disablement, but the changes in status are not notified to the terminal and warnings are not triggered This status can be used to carry out actions that could trigger false alarms, such as handling the bed in any way (making the bed) or controlled lifting of the user.

In those cases where it is advisable, the configuration with micro switches can be used to disable the acoustic notification (only during the courtesy period or for all sounds). The luminosity of the light notifications can also be reduced to avoid discomfort in the dark. This reduction does not affect the warning signals or the feedback of the use of the pushbutton, which are maintained in normal luminosity.

The "sensitivity reduction" means that the sensor will detect the absence or presence of the user more quickly, but the triggering of false alarms is more probable. This option is recommended only for demonstrations or test situations and not for normal operation.

### 3.3.4. USB communication

Characteristics of USB Communication between Bed Presence Sensor and Central Computer:

- USB Serial port (FTDI) 9600/8-N-1 (exactly 9532 baud, error 0.70%). Only alphanumerical characters are used and the "new line" ("\n") character as sent message terminator. The numerical values are sent in hexadecimal.

- Accepted commands (each command with one single character, the other characters are ignored):
    - "R": reset. The concentrator resets.
    - "S": status consultation. The concentrator responds with the status message "S".

- Commands sent:

- "R"+xx+"\n": reset. Sent when the concentrator is reset.
- "S"+xxyzw+"\n": current status. Sent as response to "S" command.
- "P"+xxyzw+"\n": long pulsation (3s) of the concentrator button.
- "a"+xxyzw+"\n": Abandonment. Sent when the courtesy period ends (alarm trigger).
- "s"+xxyzw+"\n": status change. Sent when a status change occurs (other than abandonment).

- The parameters of the previous message are the hexadecimal representation of the following values:
  - xx: value of the configuration micro switches during the last reset. Each bit is a switch.
  - y: sensor status (low level). It can take these values:
    - 1: disconnected sensor (or error)
    - 2: unknown. It only appears during the first few seconds after a reset.
    - 4: absence: No presence is detected.
    - 8: presence: Presence is detected. - z: concentrator status (high level). May indicate the following:
      - 0: standby. The user is not in bed.
      - 1: arming. The user has just gone to bed. During this time, if the user abandons the bed, it returns to standby status.
      - 2: armed. The user has been lying down for some time. If the user abandons the bed, it changes to courtesy status.
      - 3: courtesy: The user has abandoned the bed and time is given (this can be set to 0) for the user to return. If the courtesy time ends and the user has not returned to bed, the abandonment event (alarm) is triggered.
      - 4: Disabled. The concentrator is temporarily disabled because the button has been pressed. In this situation, the user can abandon the bed without triggering the alarm.
      - 5: disconnected. No sensor connected to the concentrator is detected.

### 3.3.5. Setup for the FATE system

The FATE system is setup to make the alert management from Central Computer via USB cable, disabling device alarms themselves, as they are managed by the mobile application, notified to the user and / or sent to the emergency contact if necessary.

The bed sensor has a set of switches (see Figure 11) for configuring the operation of the bed sensor (see Table 1).

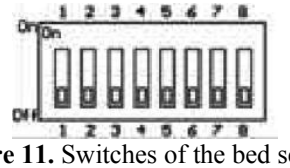



**Figure 11.** Switches of the bed sensor.

**Table 1.** Configuration options for the bed sensor through switches.

| Variable | Switch status | Setup |
|---|---|---|
| Silent Pulsation | 1 – OFF | Sounds are used as pushbutton pulsation feedback and to notify sensor disconnection. |
| | 1 – ON | These sounds are not generated |
| Arming Time | 2 – OFF | 60 s |
| | 2 – ON | 5 s |
| Courtesy Time | 3 – OFF 4 – OFF | 10 s |
| | 3 – OFF 4 – ON | 0 s |
| | 3 – ON 4 – OFF | 5 min |
| | 3 – ON 4 – ON | 15 min |
| Acoustic notification during courtesy time | 5 – OFF | The courtesy period is notified visually (red flickering of the pushbutton) |
| | 5 – ON | The courtesy period is notified visually and acoustically. |
| Temporal disablement time | 6 – OFF | 2 min |
| | 6 – ON | 10 min |
| Reduce luminosity | 7 – OFF | Normal pushbutton luminosity |
| | 7 – ON | Reduced pushbutton luminosity |
| Sensitivity reduction | 8 – OFF | Normal sensitivity |
| | 8 – ON | Sensitivity of the reduced sensor |

Table 2 shows the configuration used for the FATE system.

**Table 2.** Switch settings for the bed sensor in the FATE system.

| Switch Status | Description |
|---|---|
| 1 – ON | Turns off sounds of feedback to the user. Alarms and warnings are controlled by the mobile application. |
| 2 – ON | With five seconds of arming time is sufficient. |
| 3 – OFF 4 – OFF | Courtesy time before generating messages to the central computer is set at 10 seconds.<br>It is a reasonable time to avoid false positives, but sending relevant information to the Central Computer. |
| 5 – OFF | Turns off visual and audible notifications, allowing it to be the FATE system that decides when to notify, using mobile or emergency communication systems. |
| 6 – OFF | The deactivation management period is set at least 2 minutes.<br>The configuration and control sleep schedule alarms are controlled from the FATE system itself. |
| 7 – OFF | N/A |
| 8 – OFF | Is set to normal sensitivity to avoid false alarms. |

### 3.3.6. Bed presence sensor at nursing homes

In the case of FATE project at nursing homes, the bed presence sensor is integrated with location and identification solution provided by Gema Active Business Solutions. That means that are integrated in one device the bed presence sensor and the RFID Tag. This Tag is the responsible to communicate the alarm generated when, under the conditions configured, an

elder leaves the bed. In conclusion, all the aspects mentioned in the description of the bed presence sensor for homes, applies in the case of nursing homes with the follows difference:

- The USB link between bed presence sensor and PC is not needed in this device.

Figure 12 shows the general scheme of the integration.

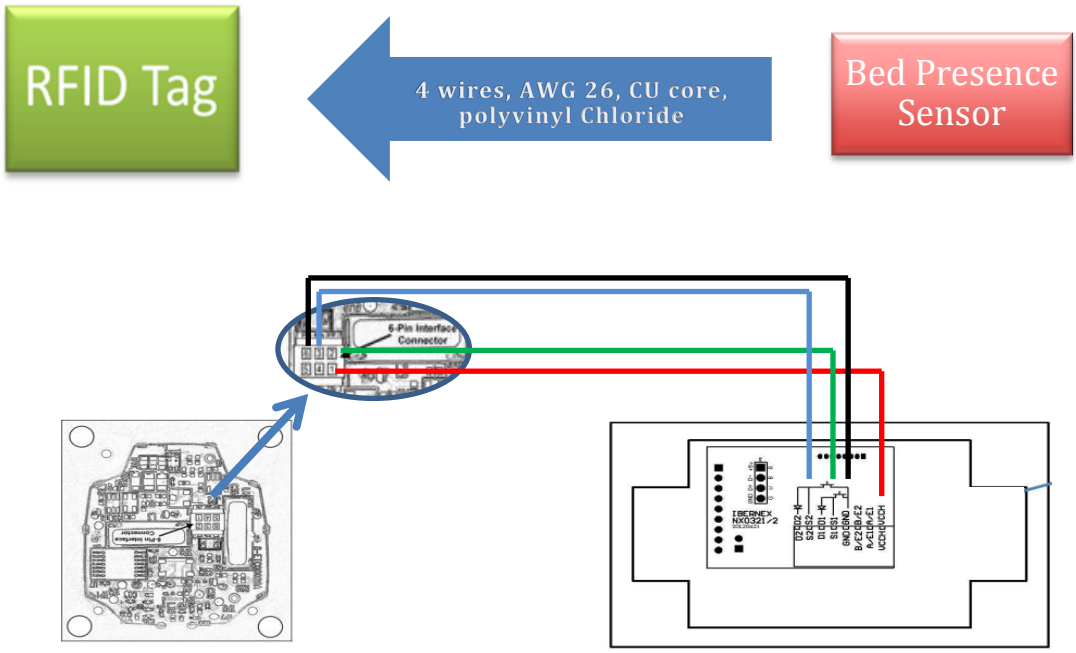



**Figure 12.** Overall scheme of the bed presence sensor and RFID tag integration.

## 3.4. Mobile phone

The mobile FATE system is responsible for making the management of messages received by the sensors of falls and bed sensor and alarm notification necessary.

The minimum requirements for a mobile device to be used in the FATE system are:

- **Compatible with Bluetooth 2.1** to communicate with the fall sensor and central computer.
- **Provide GPS** to indicate the position of the mobile user in the alarm generation time.
- **Operating System Android 2.3.4 or higher** to install the application FATE.
- **Compatible with 3G/GPRS** to communicate data over Ethernet when necessary, such as XML messages to emergency management center.

The selected mobile phone is the Samsung Galaxy Mini S5570. Its specifications are included on Table 3.

**Table 3.** Specifications of Samsung Galaxy Mini.

| | | |
|---|---|---|
| GENERAL | 2G Network | GSM 850 / 900 / 1800 / 1900 |
| | 3G Network | HSDPA 900 / 2100 |
| | SIM | Mini-SIM |
| | Announced | 2011, January |
| | Status | Available. Released 2011, February |
| BODY | Dimensions | 110.4 x 60.8 x 12.1 mm (4.35 x 2.39 x 0.48 in) |
| | Weight | 105 g (3.70 oz) |
| DISPLAY | Type | TFT capacitive touchscreen, 256K colors |
| | Size | 240 x 320 pixels, 3.14 inches (~127 ppi pixel density) |
| | | - TouchWiz v3.0 UI |
| SOUND | Alert types | Vibration; MP3, WAV ringtones |
| | Loudspeaker | Yes |
| | 3.5mm jack | Yes |
| | | - DNSe sound enhancement |
| MEMORY | Card slot | microSD, up to 32GB, 2GB included |
| | Internal | 160 MB storage, 384MB RAM |
| DATA | GPRS | Class 12 (4+1/3+2/2+3/1+4 slots), 32 - 48 kbps |
| | EDGE | Yes |
| | Speed | HSDPA, 7.2 Mbps |
| | WLAN | Wi-Fi 802.11 b/g/n, Wi-Fi hotspot |
| | Bluetooth | Yes, v2.1 with A2DP |
| | USB | Yes, microUSB v2.0 |
| CAMERA | Primary | 3.15 MP, 2048x1536 pixels, check quality |
| | Features | Geo-tagging |
| | Video | Yes, QVGA@15fps |
| | Secondary | No |
| FEATURES | OS | Android OS, v2.2 (Froyo), upgradable to v2.3 |
| | Chipset | Qualcomm MSM7227 |
| | CPU | 600 MHz ARMv6 |
| | GPU | Adreno 200 |
| | Sensors | Accelerometer, proximity, compass |
| | Messaging | SMS(threaded view), MMS, Email, Push Email, IM |
| | Browser | HTML |
| | Radio | Stereo FM radio with RDS |
| | GPS | Yes, with A-GPS support |
| | Java | Yes, via Java MIDP emulator |
| | Colors | Black |
| | | - SNS integration<br>- MP4/H.264/H.263 player<br>- MP3/WAV/eAAC+ player<br>- Organizer<br>- Document viewer/editor<br>- Image/video editor<br>- Google Search, Maps, Gmail, YouTube, Calendar, Google Talk, Picasa integration<br>- Voice memo/dial<br>- Predictive text input (Swype) |
| BATTERY | | Standard battery, Li-Ion 1200 mAh |
| | Stand-by | Up to 570 h |
| | Talk time | Up to 9 h 30 min |

The role of the mobile phone is to listen to Bluetooth communications performed by both the fall detector and the central computer and see if it is necessary to perform an emergency communication either by phone call, sending an XML web service or SMS.

As an emergency system, the application also requires:

- **Bluetooth always active** in order to make communication with the sensor or the central computer.
- **Disable AIRPLANE MODE** in case you need to make the call for emergencies.
- **Enable GPS** when outside house to indicate the position where it is in case of emergency.

For the philosophy of the Android operating system, any mobile user can control the requirements explained. That is why the FATE mobile application is checking these states all the time to activate/deactivate them automatically when needed.

## 3.5. Central computer

The FATE system uses the model Shuttle XS35GT V2 (see Figure 3) as central computer to manage communications between the ZigBee network, the bed presence sensor and cell phone (Bluetooth).

The most relevant characteristics for selecting this component were:

- Fanless Design.
- Slim PC.
- 24/7 nonstop operation (approved for permanent operation).
- Operating Systems: Linux distributions.
- VESA Mount.
- USB Ports for wireless components.

**Figure 13.** Central computer. Shuttle XS35GT V2.

Table 4 contains the specifications of the chosen central computer.

## 3.6. Wireless components

This section is devoted to the description of the components that support the wireless communication infrastructure of the FATE system. The specific Bluetooth and ZigBee communication protocols developed for the FATE system will also be explained.

### 3.6.1. Bluetooth module

As indicated in section 3.1, the Bluetooth link in the fall detector is supported by the embedded RF communication module Bluegiga WT12. This will allow for the direct communication between the mobile phone and the fall detector when the user is outside home. In order to permit the central computer to communicate with the mobile phone via a Bluetooth link it is equipped with a USB to Bluetooth adaptor supporting Bluetooth 2.1.

### 3.6.2. ZigBee module

As stated in section 3.1, the ZigBee link in the fall detector is supported by the embedded ZigBee communication module Digi Xbee RF. In order to permit the central computer to coordinate the ZigBee network at home it is equipped with a XU-Z11 USB to ZigBee adaptor from Digi. The ZigBee mesh network at home is supported by a set of XR-Z14-CW1P2 wall routers from Digi. These routers have been chosen so as to facilitate the system installation, since they just need a free electricity plug in order to operate.

**Table 4.** Specifications of Shuttle XS35GT V2.

| | |
|---|---|
| **FORM FACTOR** | 1.5 L slim form factor |
| **PROCESSOR** | Intel Atom D525 dual core processor |
| **CHIPSET** | Intel® NM10 Express Chipset |
| **MEMORY** | 1 x DDR3 SO-DIMM slot supports up to 4GB DDR3 800MHz |
| **VGA** | Built-in Nvidia next generation ION<br>Support 1080P and Blue-ray playback<br>Support D-sub + HDMI dual display |
| **AUDIO** | IDT92HD81<br>2.1 channel High Definition Audio |
| **ETHERNET** | JMC251<br>Supports 10/100/1000 Mb/Sec.<br>802.11b/g/n WLAN support |
| **STORAGE INTERFACE** | Support 1 x 2.5" SATA HDD, 5400 / 7200RPM<br>Support 1 x SD Card reader |
| **FRONT PANEL** | (1) Power button<br>(1) Power LED<br>(1) HDD LED<br>(1) USB 2.0 port<br>(1) SD card reader<br>(1) ODD bay |
| **BACK PANEL** | (1) RJ45 LAN port<br>(1) D-sub VGA port<br>(4) USB 2.0 ports<br>(1) Line out port<br>(1) MIC in port<br>(1) HDMI<br>(1) Kensington lock<br>(1) DC-in |
| **DIMENSIONS** | 252 x 154 x 33 mm |
| **POWER** | Automatics Voltage adjustment between 100 and 240VAC 50/60Hz, 40 Watts, 2 Pins |
| **ACCESSORIES** | " Optional " PV01 (VESA Mount)<br>" Optional " PHD2 (Second HDD bracket) |

### 3.6.3. Bluetooth communication protocol

The scenario covered by this protocol is the following:

- Setting up virtual serial ports (or equivalent) on two devices (e.g. PCs) and connecting these with Bluetooth, to emulate a serial cable between the two devices. Any legacy application may be run on either device, using the virtual serial port as if there were a real serial cable connecting the two devices (with RS232 control signalling).

This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates is optional. Only one connection at a time is dealt with in this profile. Consequently, only point-to-point configurations are considered. However, this should not be interpreted as imposing any limitation on concurrence; multiple executions of this profile should be able to run concurrently in the same device. This also includes taking on the two different roles (as DevA and DevB) concurrently.

Fall detector and central computer play the role of Device A in FATE project, while mobile phone does it as Device B. Device A takes initiative to form a connection to Device B, while Device B waits for Device A to take initiative to connect.

The communication is illustrated in Figure 14.

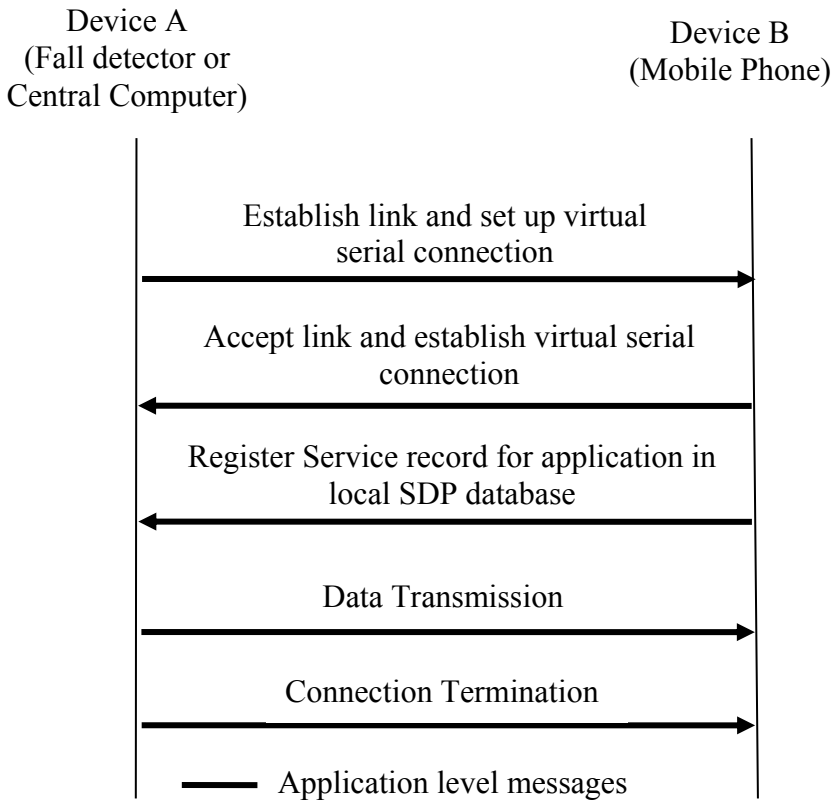



**Figure 14.** Bluetooth communication scheme in the FATE system.

In FATE system Device A sends directly an alarm order to the Device B.

The key to communicating between devices on a Bluetooth network is agreement on a profile. An example of a profile would be the Health Device Profile**.** The profile defines common actions between Bluetooth devices: wireless networks rely on the ability for autonomous

devices to join a network and discover other devices and services on devices within the network.

A Bluetooth profile describes how devices communicating over Bluetooth interact, by specifying the configuration of the channel and the sequence of data exchange needed to establish the channel. It specifies the dependencies on other protocols and profiles, and the manner in which connection is established and configured.

The Bluetooth SIG recently released the Health Device Profile (HDP) in an effort to standardize health device communication using Bluetooth technology. Since HDP is a new profile, it does not allow full interoperability between all medical sensors and collecting devices from different manufacturers. Manufacturer specific device descriptions may not be added to any public profile so any such device descriptions (like for example the fall detector) shall use a private profile ID. For this reason, the implementation has been based on Bluetooth Serial Port Profile (SPP) because it also allows the transmission of the same traffic that HDP does. As SPP is not configured for health device communication a manufacturer specific proprietary protocol will be used. The protocol defined in this document provides a common data exchange protocol and definition of device data formats.

Source refers to a *Source* of data defined by the Data Exchange Specifications, and a Sink is a receiver of that data. The *Source* may generate that data from sensors, or may relay data actually collected by some other device. The *Sink* may be a display unit, a store-and-forward intermediary, or any other consumer of Data Exchange Specification data.

Initiator is the Bluetooth device initiating an action to another Bluetooth device. The device receiving the action is called the acceptor. The initiator is typically part of an established link.

Acceptor is the Bluetooth device receiving an action from another Bluetooth device. The device sending the action is called the initiator. The acceptor is typically part of an established link.

Devices shall transfer data in standard network order (big-endian), which defines more significant (high-order) bytes being transferred before less significant (low-order) bytes, and bit ordering following the same pattern.

As an example, the decimal value 23456 (hex 5BA0) would be encoded (listing from first to last bit transmitted) as 0101101110100000 (i.e. byte 5B followed by byte A0).

To ensure secure data transfer, all connections shall always be on authenticated links. The Bluetooth 2.1 + EDR specification mandates the use of Secure Simple Pairing. This is very useful for health devices, as it allows an easier and more secure exchange of authentication and encryption credentials.

Many of the services offered over Bluetooth can expose private data or allow the connecting party to control the Bluetooth device. For security reasons it is necessary to be able to recognize specific devices and thus enable control over which devices are allowed to connect to a given Bluetooth device. At the same time, it is useful for Bluetooth devices to be able to establish a connection without user intervention.

To resolve this conflict, Bluetooth uses a process called *pairing*. During the pairing process, the two devices involved establish a relationship by creating a shared secret known as a *link key*. If a link key is stored by both devices they are said to be *paired*. Once pairing successfully

completes, a link will have been formed between the two devices, enabling those two devices to connect to each other in the future without requiring the pairing process in order to confirm the identity of the devices.

Device descriptions specified in FATE profile are inspired from the public Health Device Profile and summarized in
The contents contain data about a specific interface and store the current state of a given device. This set of contents has been defined just for the FATE Project with the purpose of transmitting and requesting information between the Mobile Terminal and the local devices.

Contents are identified by a 8-bit number and read access to these contents is required.

Table 6 lists the Content IDs representing the observed values, arranged into sets of device-related contents (fields are filled with examples).

**Table 6.** Fall detector and computer related contents.

| Content ID | Content Name | Content Description | Format |
|---|---|---|---|
| 0x00 | Vigilance Control Message | This message is used to verify the presence of the fall detector. | Unsigned 8-bit integer |
| 0x01 | Fall Down Alarm | This message is used to inform the detection of a fall down. | Unsigned 8-bit integer |
| 0x02 | Fall Down Automatic Recovery | This message is used to inform the automatic detection of a recovery after a fall down. | Unsigned 8-bit integer |
| 0x03 | Fall Down Manual Recovery | This message is used to inform the Manual (Panic Button) recovery after a fall down. | Unsigned 8-bit integer |
| 0x04 | Battery Charging | This message is used to inform that the device is shutting down because it is charging. | Unsigned 8-bit integer |
| 0x05 | Panic Activated | Panic button pressed when there is no fall down alarm | Unsigned 8-bit integer |
| 0x06 | Battery Low | This message is used to inform that the battery of the device is in critical level. | Unsigned 8-bit integer |

**Table 5.** Bluetooth device descriptions in the FATE system.

| Device Description | Device ID |
|---|---|
| Mobile phone | 0x01 |
| Fall detector | 0x02 |

The contents contain data about a specific interface and store the current state of a given device. This set of contents has been defined just for the FATE Project with the purpose of transmitting and requesting information between the Mobile Terminal and the local devices.

Contents are identified by a 8-bit number and read access to these contents is required.

Table 6 lists the Content IDs representing the observed values, arranged into sets of device-related contents (fields are filled with examples).

**Table 6.** Fall detector and computer related contents.

| Content ID | Content Name | Content Description | Format |
|---|---|---|---|
| 0x00 | Vigilance Control Message | This message is used to verify the presence of the fall detector. | Unsigned 8-bit integer |
| 0x01 | Fall Down Alarm | This message is used to inform the detection of a fall down. | Unsigned 8-bit integer |
| 0x02 | Fall Down Automatic Recovery | This message is used to inform the automatic detection of a recovery after a fall down. | Unsigned 8-bit integer |
| 0x03 | Fall Down Manual Recovery | This message is used to inform the Manual (Panic Button) recovery after a fall down. | Unsigned 8-bit integer |
| 0x04 | Battery Charging | This message is used to inform that the device is shutting down because it is charging. | Unsigned 8-bit integer |
| 0x05 | Panic Activated | Panic button pressed when there is no fall down alarm | Unsigned 8-bit integer |
| 0x06 | Battery Low | This message is used to inform that the battery of the device is in critical level. | Unsigned 8-bit integer |

Data Type ID indicates what type of contents the response command requires. Each data type is allocated an 8-bit data type ID, as indicated in Table 7.

The Body Area Network Communication Protocol requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The Protocol specifies how commands, commands responses and data transmissions are sent and received from the modules.

Protocol general structure satisfies the pattern depicted in Figure 15. The structure of the frame data field varies slightly depending on the type of the message. In addition, the following scheme specifies the byte order of the transmission.

**Table 7.** Data type IDs.

| ID | Data Type |
|---|---|
| 0x01 | Title |

| | |
|---|---|
| 0x02 | Long Octet String |
| 0x04 | Long Character String |
| 0x08 | RSS Feed |
| 0xE2 | Time |
| 0x20 | Unsigned 8-bit integer |
| 0x21 | Unsigned 16-bit integer |
| 0x22 | Unsigned 24-bit integer |
| 0x23 | Unsigned 32-bit integer |

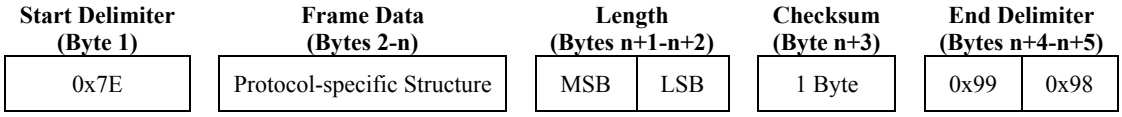



**Figure 15.** Structure of a frame.

The meaning of the different fields that constitute a frame is the following:

- **Start Delimiter:** the start delimiter field indicates the beginning of the frame. Any data received prior to the start delimiter is silently discarded. Fixed to 0x7E.
- **Frame Data:** data payload of the frame data forms a Protocol-specific structure as depicted in Figure 16.

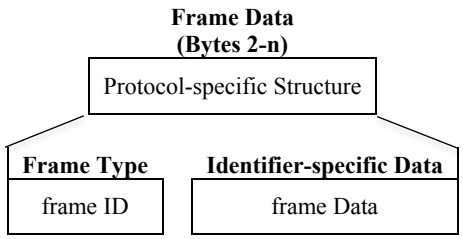



**Figure 16.** Structure of the Frame Data field.

The frame ID field (Frame Type) indicates which message will be contained in the frame Data field (Identifier-specific Data).

The message identifiers of most common messages are shown in Table 8.

**Table 8.** List of message identifiers.

| Message Names | Frame ID |
|---|---|

| | |
|---|---|
| Push Information | 0x8A |
| Push Information Response | 0x8B |
| Request Information | 0x8C |
| Request Information Response | 0x8D |
| Write Attributes | 0x8E |
| Write Attributes Response | 0x8F |

- **Length:** the length field has a two-byte value that specifies the number of bytes that will be contained in the frame data field. It does not include the start delimiter field, the checksum field and the end delimiter field.

- **Checksum:** to test data integrity, a checksum is calculated and verified.
    - To calculate: Not including frame delimiters and length add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.
    - To verify: Add all bytes (include checksum, but not the delimiter and length). It the checksum is correct, the sum will equal 0xFF.

- **End Delimiter:** the end delimiter has a two-byte field and indicates the ending of the frame. Any data received after the end delimiter is silently discarded. Fixed to 0x99 and 0x98.

Data exchange between the Mobile Terminal and the local devices (Fall detector and computer) is obtained using the Push Information message. In the default case data values are transferred directly from local device to the Mobile Terminal. In the next section we describe messages to be used in this case.

The local device sends a spontaneous event report to the Mobile Terminal with measurement observations. The Mobile Terminal confirms receipt of the local device's event report by means of Push Information Response message, as depicted in Figure 17.

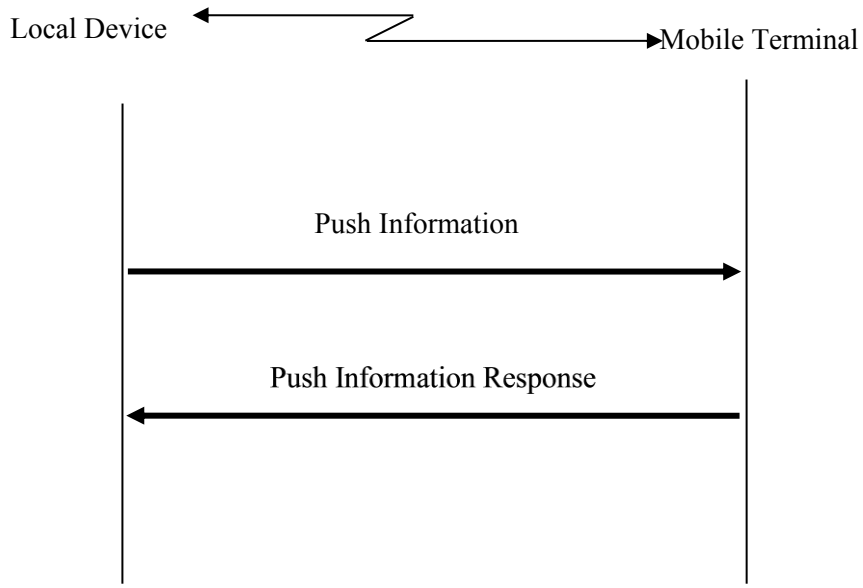



**Figure 17.** Push information message sequence in the Bluetooth protocol.

The structure of a push information message is presented in Table 9.

**Table 9.** Structure of a push information message in the FATE Bluetooth protocol.

| Field | Length (Bytes) | Description |
|---|---|---|
| Frame Type | 1 | Fixed Value = 0x8A |
| Transaction Sequence Number | 1 | Counter |
| Device ID | 1 | Type of device |
| Content ID | 1 | Type of value measured |
| Data Type ID | 1 | Type of encoding |
| Data Length | 2 | Length of measured value in Bytes |
| Data | var | Measured value |

The structure of a push information response message is depicted in Table 10.

**Table 10.** Structure of a push information response message in the Bluetooth protocol.

| Field | Length (Bytes) | Description |
|---|---|---|
| Frame Type | 1 | Fixed Value = 0x8B |
| Transaction Sequence Number | 1 | Counter |
| Device ID | 1 | Type of device |
| Content ID | 1 | Type of value measured |
| Response Code | 1 | Section 4.2.2 |

The following list provides an overview of the possible requests and responses that are possible in the FATE system:

- When a device sends any message it shall wait for the corresponding response before sending any other message.
- When a device receives any message, it shall send a response before sending any other message on the Control Channel.
- If the Control Channel is closed before a response is received, then the device that had been expecting the response shall behave as if it received an Unspecified Error response.
- If a message is received when a response is expected (i.e. out of sequence operation), then it is assumed that the two devices have sent simultaneous messages. When this occurs the Initiator (of the Control Channel) shall "win" as indicated by:
    - The message received by the Acceptor shall be processed as normal.
    - The message received by the Initiator shall be ignored.

    The next message on the Control Channel shall be the response sent from the Acceptor to the Initiator.
- If a response is received when no request is outstanding, then it shall be ignored.
- It there is no response within 800ms, the device will send the same message until the response is received.
- All responses shall include a "Response" value (where zero (0x00) indicates the request was successful).
- If an invalid message is received, the Response Packet shall set the Status Feedback to "Invalid Message ID".
- If a device does not support messages, but receives a message, that device shall respond with a "Request Not Supported".

The Response Packet shall contain the appropriate Response Code. Response descriptions when using Standard Op Codes are defined in Table 11.

**Table 11.** Response code descriptions.

| Response Code | Response Name | Response Code Description |
|---|---|---|
| 0x00 | Success | The corresponding message was received and processed successfully |
| 0x01 | Invalid Message ID | The message received is not valid |
| 0x02 | Invalid Parameter Value | One or more of the values received is invalid. This shall be used when:<br>• The request length is invalid<br>• Some of the parameters have invalid values |
| 0x08 | Resource Unavailable | The device is temporarily unable to complete the request. This is intended for reasons relating to the physical sensor (e.g. hardware fault, low battery), or when processing resources are temporarily committed to other processes. |
| 0x09 | Unspecified Error | An internal error other than those listed in this table was encountered while processing the request |

The Transaction Sequence Number (TSN) field is 8-bits in length and specifies an identification number for the transaction so that a response-style command frame can be related to a request-style command frame. The Transaction Sequence Numbers should be the same for the requests and responses; the default responses should use also the same transaction sequence numbers of the related commands in order to match the correspondent packets.

Transaction Sequence Number should be incremented each time a device sends a command. When a value of 0xFF is reached, the next command shall re-start the counter with a value of 0x00.

When a device sends the same message because there is no response, TSN field won't be incremented.

### 3.6.4. ZigBee communication protocol

The Body Area Network architecture proposed is a classical star topology network in which the central computer (CC) plays the double role of coordinator and gateway while the Fall Detector acts as end device. To simplify the communication, it is assumed that the ZigBee radio interface of the CC is always on while the local device is in sleep mode when not transmitting: for this reason, communication is always initiated by the local device, which also decides at which frequency to contact the Gateway. Figure 18 provides an overview of the ZigBee communication scheme in the FATE system.

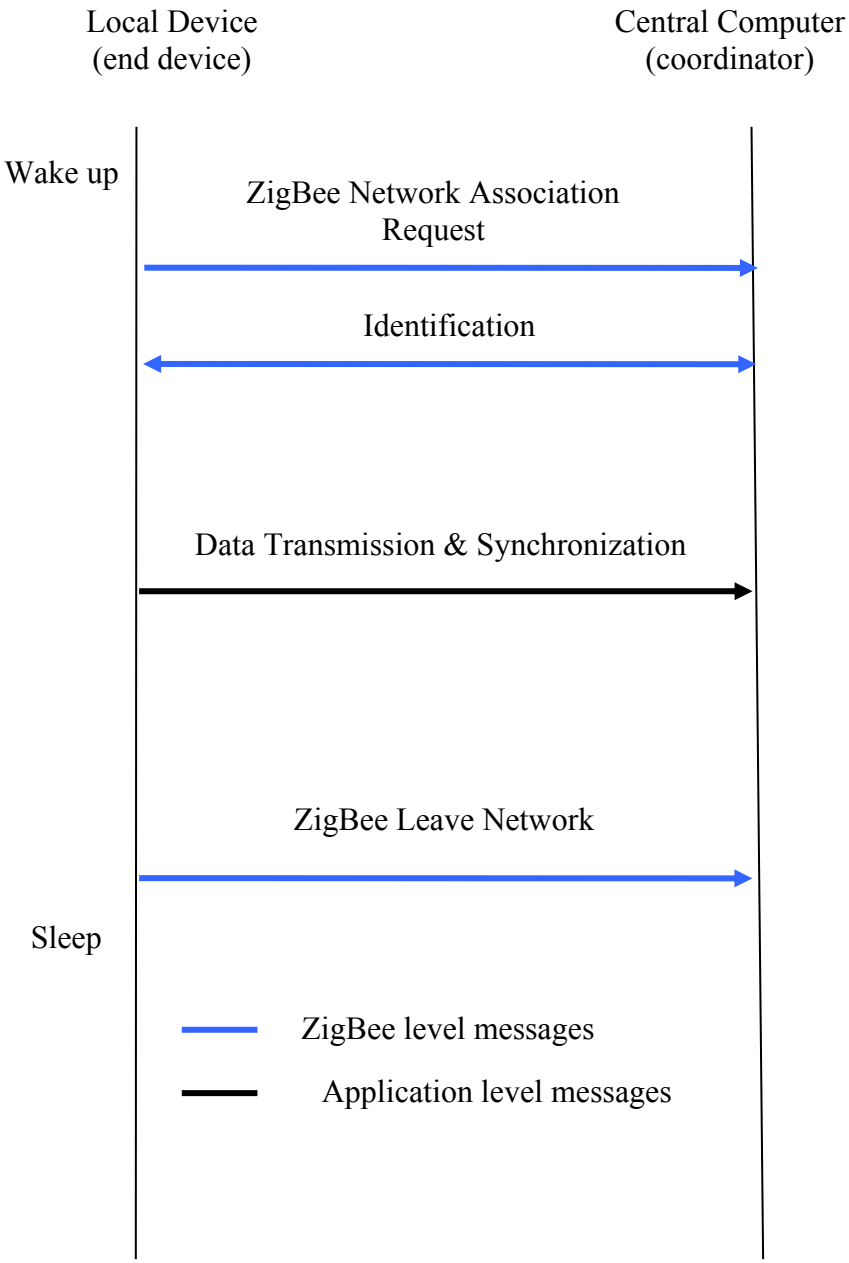



**Figure 18.** ZigBee communication scheme in the FATE system.

Even if the communication scheme proposed is the same for every local device, its behaviour could result different since devices have different needs.

On the other hand, when data expected from the local devices are not received, the Central Computer shall send an alert to the Call Centre by means of the Mobile Phone.

The key to communicating between devices on a ZigBee network is agreement on a profile. An example of a profile would be the Telecom Application Profile (0x0107)**.** The profile defines common actions between ZigBee devices: wireless networks rely on the ability for autonomous devices to join a network and discover other devices and services on devices within the network. Device and service discovery are features supported within the device profile.

ZigBee defines profiles in two separate classes: manufacturer-specific and public. Manufacturer specific device descriptions may not be added to any public profile so any such device descriptions shall use a private profile ID.

Once the profile identifier is defined, that profile identifier permits the profile designer to define the following:

- **Device descriptions**
- **Cluster identifiers**

For each profile identifier, there exists a pool of device descriptions described by a 16-bit value (meaning there are 65,536 possible device descriptions within each profile) and a pool of cluster identifiers described by a 16-bit value (meaning there are 65,536 possible cluster identifiers within each profile). Each cluster identifier also supports a pool of attributes described by a 16-bit value. As such, each profile identifier has up to 65,536 cluster identifiers and each of those cluster identifiers contains up to 65,536 attributes.

For public profile identifiers defined within the ZigBee Alliance, a cluster library has been created which provides a common definition and enumeration of clusters and their attributes. The cluster library is designed to sponsor re-use of cluster and attribute definitions across application profiles. By convention, when public profiles employ the ZigBee Cluster Library (ZCL), they will share a common enumeration and definition of cluster and attribute identifiers. The attributes that a particular cluster is capable of reporting are listed in the ZCL specification for each cluster.

A single ZigBee device may contain support for many profiles, provide for subsets of various cluster identifiers defined within those profiles, and may support multiple device descriptions. This capability is defined using a hierarchy of addressing within the device as follows:

- **Device:** the entire device is supported by a single radio with a unique IEEE and NWK address.
- **Endpoints:** this is an 8-bit field that describes different applications that are supported by a single radio. Endpoint 0x00 is used to address the device profile, which each ZigBee device must employ; endpoint 0xff is used to address all active endpoints (the broadcast endpoint). Consequently, a single physical ZigBee radio can support up to 254 applications on endpoints 0x01- 0xfe. Endpoints 0xf1-0xfe can only be used for ZigBee Alliance approved applications.

Once a device is created to support specific profiles and made consistent with cluster identifier usage for device descriptions within those profiles, the applications can be deployed. To do this, each application is assigned to individual endpoints.

Prior to any messages being directed to a device, it is assumed by the ZigBee protocol that service discovery has been employed to determine profile support on devices and endpoints (service discovery is made on the basis of profile identifier, input cluster identifier list, and output cluster identifier list). Table 12 shows the device descriptions available in the FATE system.

**Table 12.** ZigBee device descriptions in the FATE system.

| Device Description | Device ID |
|---|---|
| Central Computer | 0x0000 |
| Fall Detector | 0x0001 |

Each device used in this profile (at least the Central Computer) should be able to indicate to the user:

- That it is the coordinator of a network
- That it has successfully joined a network.
- That another device has successfully joined its network.
- That it has failed to join a network.
- That it is in the process of searching for or joining a network.

These indications may be implemented in a number of ways including indicator light(s) or an audible indicator.

The clusters proposed to be used in FATE profile, are listed in Table 13.The clusters are listed according to the functional domain they belong to in the ZCL.

**Table 13.** ZigBee clusters in the FATE profile.

| Functional Domain | Cluster Name | Cluster IDs | Description |
|---|---|---|---|
| Telecommunication | Information | 0x0900 | Attributes and commands for providing Information service to a ZigBee device. |

The information cluster will be used to exchange data packets between the Central Computer and the local devices. In the following table provides the Information Cluster Attribute Set.

Data Type ID indicates what type of contents the response command requires. Each data type is allocated an 8-bit data type ID. In this protocol Long Octet String (0x02) and Time (0xE2) Data types will be used respectively in the Information Cluster and in the Time cluster while unsigned integer Data types will be used in the Operational Commands cluster. Long Octet Strings are coded in Big Endian mode while Time fields are coded in Little Endian Mode. Table 15 lists the Data Type IDs to be used in the FATE profile.

**Table 14.** Information cluster attribute set.

| Attribute identifier | Attribute Name | Description | Format | Access | Qualifier |
|---|---|---|---|---|---|
| 0x00 0x00 | Node Description | String that defines the type of device | Character String | ReadOnly | Mandatory |
| 0x00 0x01 | Delivery Enable | Indicates whether the cluster provides information delivery service on pull-basis | 0x00 = No<br>0x01 = Yes | ReadOnly | Mandatory |
| 0x00 0x02 | Push Information timer | Indicates whether the cluster is able to send Push Information command and the time between those commands | Unsigned 32-bit integer | ReadOnly | Optional |
| 0x00 0x03 | Enable Secure Configuration | Indicates whether an application layer 15 security is required in order to process the configuration commands | 0x00 = No<br>0x01 = Yes | ReadOnly | Mandatory |

Figure 19 presents with more detail the ZigBee Network Creation procedure: the CC and the local device (LD) must share the same radio channel and the same PAN ID, next identification is made on the base of Profile ID and Cluster IDs.

The Simple Descriptor Request message can also be used by the CC to verify if network communication with LD is still alive.

**Table 15.** ZigBee data type IDs.

| ID | Data Type |
|---|---|
| 0x01 | Title |
| 0x02 | Long Octet String |
| 0x04 | Long Character String |
| 0x08 | RSS Feed |
| 0xE2 | Time |
| 0x20 | Unsigned 8-bit integer |
| 0x21 | Unsigned 16-bit integer |
| 0x22 | Unsigned 24-bit integer |
| 0x23 | Unsigned 32-bit integer |

The join parameters for the Local Devices (LD) are listed in Table 16.

**Table 16.** Join parameters for the Local Devices.

| Parameter | Description | Fall Detector |
|---|---|---|
| ScanAttempts | At boot time or when instructed to join a network, the device should complete up to n scan attempts to find its original PAN to join. | 3 |
| TimeBetweenScans | Determines the number of seconds between each unsuccessful scan attempt. | 1 sec |
| RejoinInterval | How quickly a device will attempt to rejoin the network if it finds itself disconnected. | 30 sec |
| MaxRejoinInterval | This parameter is intended to throttle how often a device will scan to find its network in case the network is no longer present and therefore a scan attempt by the device would always fail. | 30 sec |

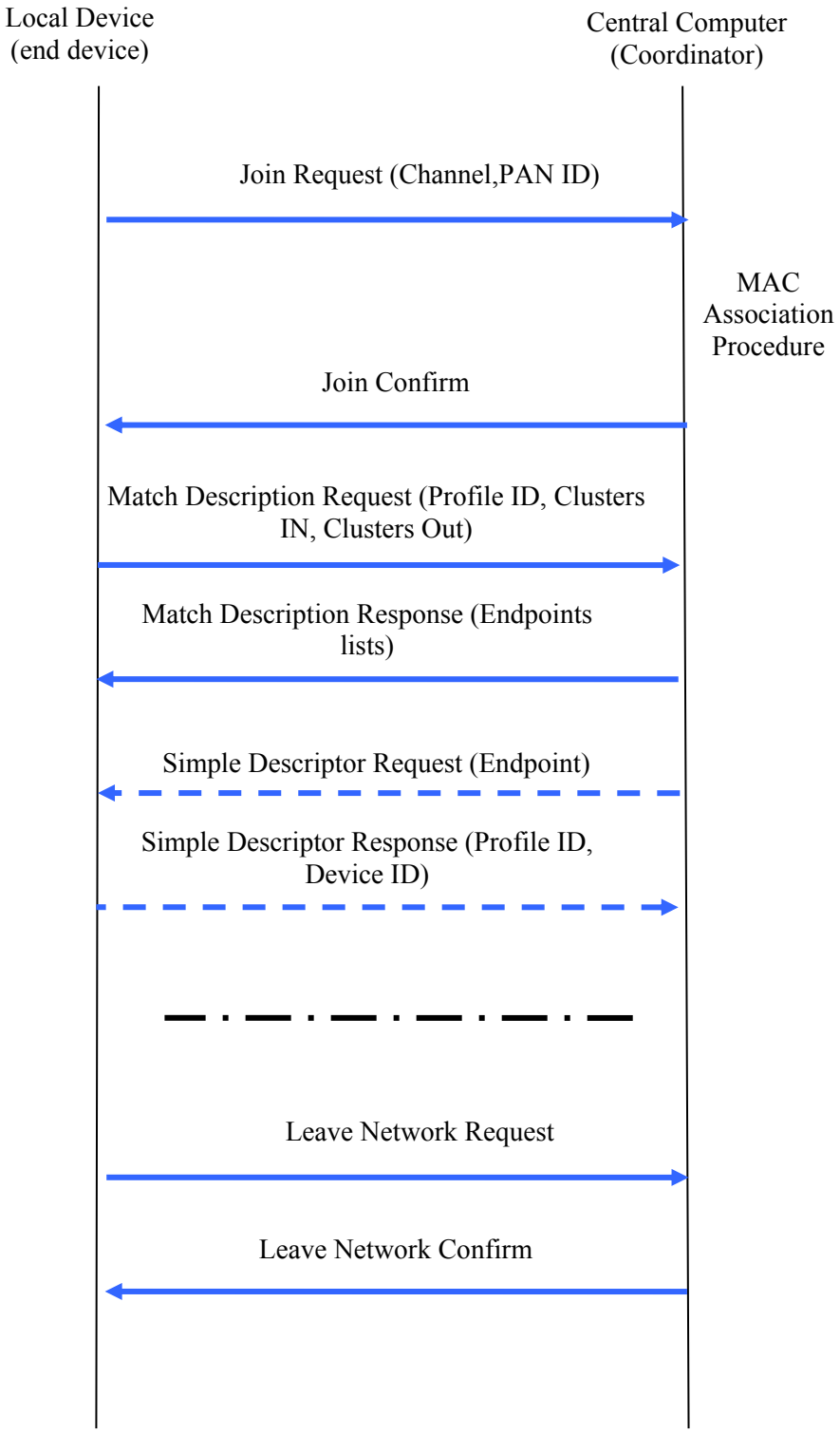



**Figure 19.** ZigBee network creation procedure.

Data exchange between the Central Computer and the local devices is obtained using the Information Cluster described previously and identified by ID 0x0900. In the default case data values are transferred directly from local device to the Central Computer.

In the case of an unsolicited measurement data transmission the LD sends a spontaneous event report to the Central Computer with measurement observations, as depicted in Figure 20.

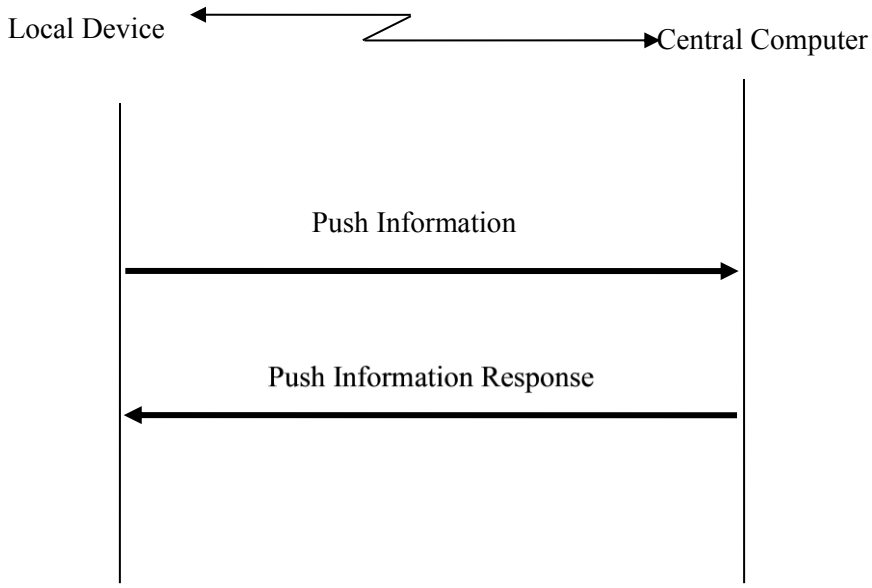



**Figure 20.** Unsolicited measurement data transmission scheme.

The contents of the different fields of the push information message are presented in Table 17, while the contents of the fields that constitute the push information response message are listed in Table 18.

Finally, Table 19 shows the observed values for the fall detector (content set identifier 0x01).

**Table 17.** Fields in the push information message of the ZigBee protocol.

| Field | Length (Bytes) | Meaning |
|---|---|---|
| Frame Control | 1 | Fixed Value = 0x19 |
| Transaction Sequence Number | 1 | Counter |
| Command ID | 1 | Fixed Value = 0x01 |
| Number | 1 | Number of contents transmitted |
| Content ID 1 | 2 | Type of value measured (Little Endian) |
| Data Type ID | 1 | Type of encoding |
| Data Length | 2 | Length of measured value in Bytes (Little Endian) |
| Data | var | Measured value (Big Endian) |
| Children 1 | 1 | Fixed Value = 0x00 |
| Content ID 2 | 2 | Type of value measured (Little Endian) |
| Data Type ID | 1 | Type of encoding |
| Data Length | 2 | Length of measured value in Bytes (Little Endian) |
| Data | var | Measured value (Big Endian) |
| Children 2 | 1 | Fixed Value = 0x00 |
| …….. | …… | …… |
| Content ID n | 2 | Type of value measured (Little Endian) |
| Data Type ID | 1 | Type of encoding |
| Data Length | 2 | Length of measured value in Bytes (Little Endian) |
| Data | var | Measured value (Big Endian) |
| Children n | 1 | Fixed Value = 0x00 |

**Table 18.** Fields in the push information response message in the ZigBee protocol.

| Field | Length (Bytes) | Meaning |
|---|---|---|
| Frame Control | 1 | Fixed Value = 0x11 |
| Transaction Sequence Number | 1 | Counter |
| Command ID | 1 | Fixed Value = 0x01 |
| Content ID 1 | 2 | Type of value measured (Little Endian) |
| Status Feedback | 1 | 0x00 = Ok<br>0x01 = Failure<br>0x02 = System not ready |
| Content ID 2 | 2 | Type of value measured (Little Endian) |
| Status Feedback | 1 | 0x00 = Ok<br>0x01 = Failure |
| …….. | …… | …… |
| Content ID n | 2 | Type of value measured (Little Endian) |
| Status Feedback | 1 | 0x00 = Ok<br>0x01 = Failure |

**Table 19.** Content ID for the values provided by the fall detector.

| Content ID | Content name | Content Description | Format |
|---|---|---|---|
| 0x01 0x00 | Vigilance Control Message | This message is used to verify the presence of the fall detector. | Unsigned 8-bit integer |
| 0x01 0x01 | Fall Down Alarm | This message is used to inform the detection of a fall down. | Unsigned 8-bit integer |
| 0x01 0x02 | Fall Down Automatic Recovery | This message is used to inform the automatic detection of a recovery after a fall down. | Unsigned 8-bit integer |
| 0x01 0x03 | Fall Down Manual Recovery | This message is used to inform the Manual (Panic Button) recovery after a fall down. | Unsigned 8-bit integer |
| 0x01 0x04 | Battery Charging | This message is used to inform that the device is shutting down because it is charging. | Unsigned 8-bit integer |
| 0x01 0x05 | Panic Activated | Panic button pressed when there is no fall down alarm | Unsigned 8-bit integer |
| 0x01 0x06 | Battery Low | This message is used to inform that the battery of the device is in critical level. | Unsigned 8-bit integer |

### 3.6.5. ZigBee network coverage analysis for installation

#### 3.6.5.1. The installation process

After the installation, the ZigBee network has to be able to serve the FATE system as a wireless communication channel between the fall detector, and the central computer at the home of elderlies. For this, the installation has to be performed in a way that ensures that the parts of the network can communicate to each other trustworthily, without any problem. The installation will be documented to record the used materials, as well as the results of the network qualification. The installation report will be the reference for each location in case of technical issues during the pilot period.

To reach the goal the main topic during the installation is to find the optimal places of the routers in the homes. The placement procedure will be defined later in a dedicated section. The houses will be different in all cases, therefore, the ideal router topology is can not be determined in advance, especially that the floor-plan of the houses is not known due to legal reasons. The positions of the routers can be defined by an algorithm, but due to the very various forms of the homes, easily can be that the router position will not have been selected based on the given algorithm. The experience of the installer person helps the process, however the described method aims to give an aid for that.

### 3.6.5.2. Qualifying the network

After installation, the network has to be checked to be sure, that the installation process was successful. This will be done during the network installation test procedure. The result of the test will be added to the installation report that contains all the relevant data regarding to the investigated network, and the results of the measurement.

To qualify a wireless link, three parameters will be used:

- **LQI (Link Quality Index):** It shows how easily the received signal can be demodulated. Lower values indicates better link than higher ones.

- **RSSI (Received Signal Strength Indicator):** It gives information about the power of the received signal. Higher value shows bigger received signal power, thus better link quality. If these two parameters measured to qualify the link, the goal is to have high RSSI, and low LQI. In this case the received signal has enough power to differ from the noise, and the received signal has not affected by other wireless devices.

- **PER (Packet Error Rate):** During a link test, measurements of this parameter is the method that clearly measures the rate of the correctly and the incorrectly received packets. During PER test, the transmitter sends a defined amount of packet to the receiver, and the receiver counts the packets that have arrived without any disturbances. The packet error rate (PER) is the number of incorrectly received data packets divided by the total number of received packets. The value of the PER is measured in percentage. As the result is closer to 0%, as better the link is. As a general rule, the PER has to be maximum 1%.

### 3.6.5.3. Device considerations

The installation process will be done in parallel with a continuous network check, to see how the parameters are changing together with the location.

To test exactly the same network that will be used later, the test should have been performed with exactly the same devices, and environment as during the pilot. This is possible only if the fall sensor could be used for the tests, (worn by the pilot participant, as the ZigBee antenna will be close the human body thus, the field of the antenna is affected by the user). But it is not realistic that the elderly walks around the rooms, to map the network. Therefore the worst case will be tested with other devices, without the need to involve pilot participants in technical details.

| |
|---|
| In the fall detector, the XBee Pro module (XBP24BZ7PIT-004J) is used.<br>The Transmitter Power: +18dBm<br>The receiver sensitivity: -102dBm @ 1% PER. |

| |
|---|
| Compared to the XStick (XU-Z11) used at the PC of the FATE system:<br>The Transmitter Power: +0dBm<br>The receiver sensitivity: -90dBm @ 1% PER. |

If during the installation the XStick is used to test the network, this 10dBm difference in the receiver sensitivity can be used as a safety margin to ensure the proper signal transmission at

any cases during the pilot period. The signal propagation depends on many parameters of the transmission channel, such as humidity, antenna orientation, blocking objects –even the user's body. So, this 10dBm margin is must to be kept to ensure the flawless operation of the network. Also the technical incidents have to be avoided, and the system has to be operating without any technical aid. The key point of the FATE system is, that the user acceptance through the invisible technical operation should be high enough.

However this approach of the network installation is a bit over-secured which can led to excessive usage of the routers. Additional tests had been performed to ensure that this method is not wasting routers during the installation.

During the installation, in all the cases commercial products will be used to make a secure and comparable way for the installation of the networks parallel in the different pilots. Furthermore standardized equipment can be used by wider range of people to install the system. This will be important when the system will have to be marketed. As hardware, two XStick-s will be used. With these devices, the safety margin can be maintained as mentioned above. One plugged into the Central Computer, and another with a laptop to check the field strength at a particular location. On the laptop, the Digi's X-CTU software will be used to make the range test as an aid for the router placement.

### 3.6.5.4. Range test method

To define the placement of the routers, the field strength has to be checked around the walking area of the user's home. The range test has to be performed with a built-in test function of the X-CTU.

As a ZigBee module receives a packet, the RSSI value can be read out from the device. This gives information about the field strength in a certain point. With the built-in test, the X-CTU can be used for a PER test. The software indicates the properly received packets in percentage. That means that the PER is:

PER=100-"Percent" [%]

Using the X-CTU, in the easiest way, the link can be tested from the coordinator side. During the installation, the following constraints have to be kept in mind:

1. The PER rate has to be readable at the actual position, not at the central computer.

2. The RSSI value indicates the received signal strength of the last hop in the path. This has to be checked at the moving node as well.

3. For the PER measurement, the data transmission has to be continuous.

Due to these reasons, the function of the nodes will be swapped. The signal propagation is reciprocal, the measured values do not depend on the device role in the network. In this case, the coordinator will be moving in the house, and the target node will be plugged into the central computer. The target node will be configured as a simple Router. The reason for this is that by definition a ZigBee standard, the End Device must have a sleep time, therefore the PER test would display the errors caused by the node in sleep mode for some time. The router can be addressed directly, therefore the data transmission between the two nodes can be continuous.

Figure 21 shows the components that permit to carry out the range test.

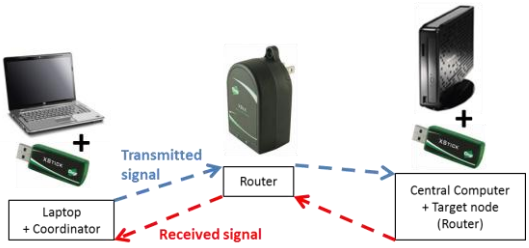



**Figure 21.** Components of the range test.

Before the range test, the used XSticks has to be configured with the X-CTU to be able to communicate to each other. The configuration, and the link test of the two XStick with the same Laptop will be the first stem to check that the modules that they can be used for the link test. The X-CTU software can be downloaded from the Digi's website[3].

In order to test the modem the following steps have to be completed:

- Connect the XStick to the laptop via the USB port.
- Then start the X-CTU program. It's important, that the module has to be connected before the software is started.
- On the PC settings tab, the module can be tested as it shown in Figure 22.

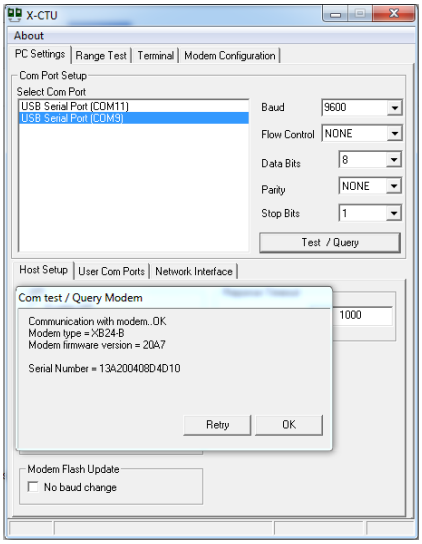



**Figure 22.** Modem test with X-CTU.

The two modules used during the installation have to be configured as a coordinator and as a router, respectively. To ensure the communication between the modules, the following rules have to be followed:

- The PAN_ID and the Stack Profile have to be identical for the two devices
- For the Router, the Destination address is:0x0000
- For the coordinator, the Destination address is the serial number of the Router.

[3] http://www.digi.com/support/productdetail?pid=3352&osvid=57&type=utilities

The summary of the setup is shown in the Figure4. There, the two configuration tabs of the modules are next to each other for better understanding.

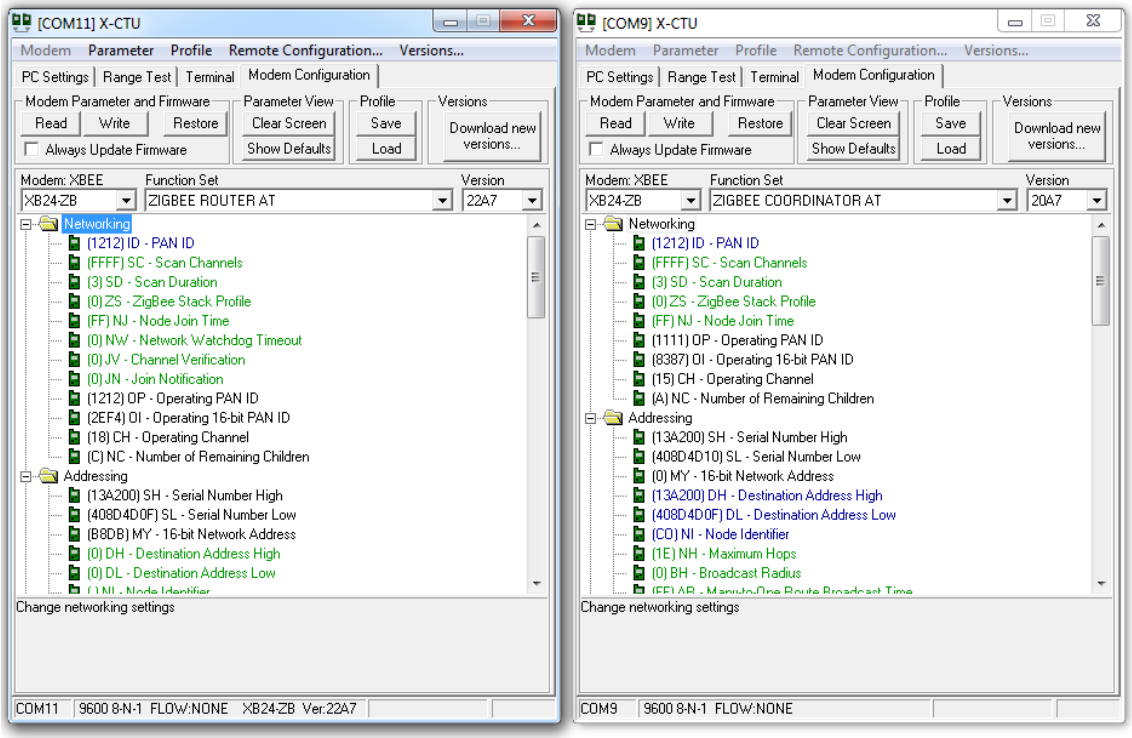



**Figure 23.** Module configurations.

After the proper setup, the coordinator will detect the router after the "Node Discover" command. The serial number of the router can be read in the response message. (Figure 24 left). For the details of the message, please refer to the AT command reference tables[4].

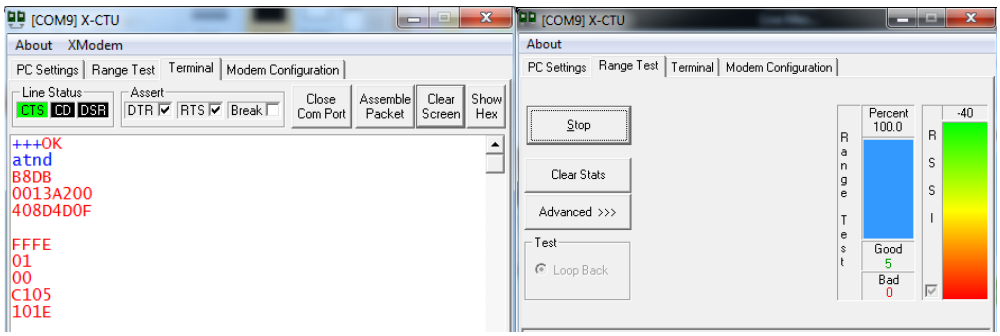



**Figure 24.** Connected router and Coordinator.

Then the link test between the two modules can be performed. The results will show excellent signal condition. (Figure 24 right). The method of the link test is described in details in the reference [5]of DIGI.

[4] ftp://ftp1.digi.com/support/documentation/90000976_C.pdf
[5] ftp://ftp1.digi.com/support/documentation/90001067_a.pdf

### 3.6.5.5. Process flow

This section describes the ZigBee installation process. This process helps to install the network parts in the homes of the pilots, but in all cases the installation has to be checked for the proper operation. Figure 25 illustrates this process.

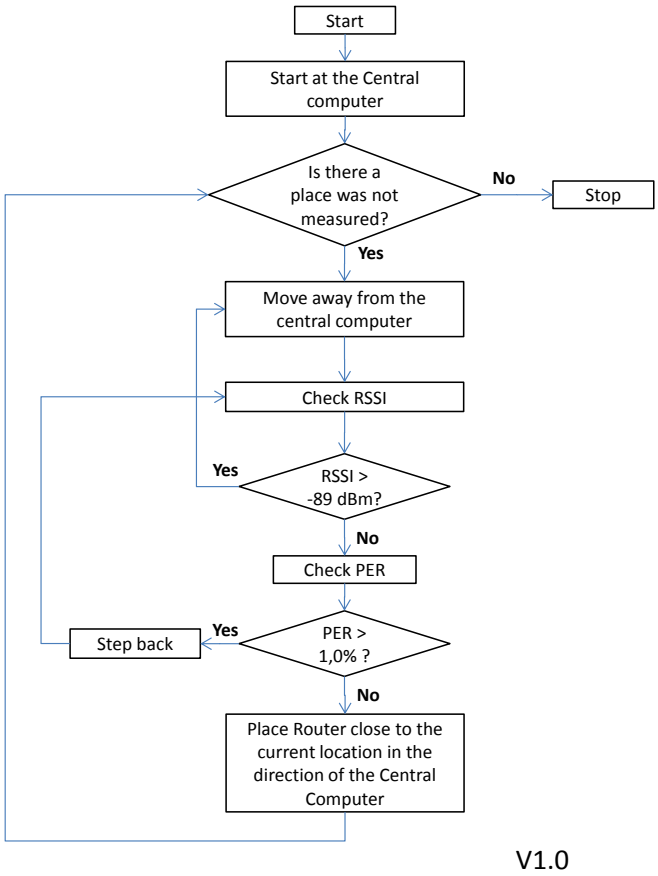



**Figure 25.** ZigBee installation process.

The following aspects have to be kept in mind:

- The installation starts at the Central computer.
- The routers have to be installed off the floor.
- Routers have to be placed on the inner walls of the house –not to waste the router's range for the outdoor area. (The house has to be walked around, to gain information about the rough its floorplan, as it won't be provided due to legal reasons).
- ***Important:*** **the nature of the house can be used for better placement, in most cases a location has to be find for the router where form the biggest possible area has a direct line of sight.**

### 3.6.5.6. Range test results

The ZigBee network is tested in the office of MFKK. The link was tested in dedicated points in the office. First the signal propagation was tested to check the usage range of the XStick USB adapter. Then a router was added to the network to extend its range. The measurement area was 130m2, where the link was tested in 35 different positions. Then the results were placed in an 80x80 matrix and all the elements of the matrix were calculated based on iterative average

calculation. This gives us good visualization of the results, and the changes in the network in case of different modifications. Figure 26 displays these results.

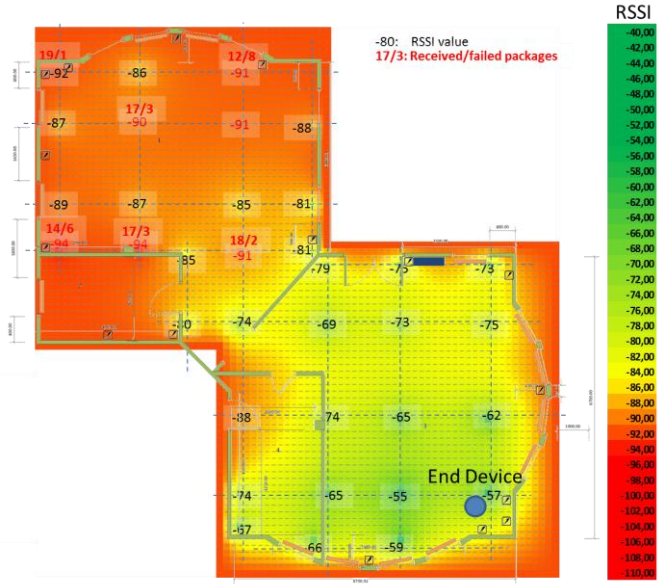



**Figure 26.** ZigBee network measurements with the XStick USB adapter.

With the XStick, just one half of the floor could be covered without packet errors. The wall between the two halves is made of plasterboard. This caused less attenuation than a brick or concrete wall.

The router was installed in the same room as the end device, but too far from the critical area in the upper room. As the result, the received signal strength increased in the area of the crossing point between the two rooms, but there is also an uncovered area left.

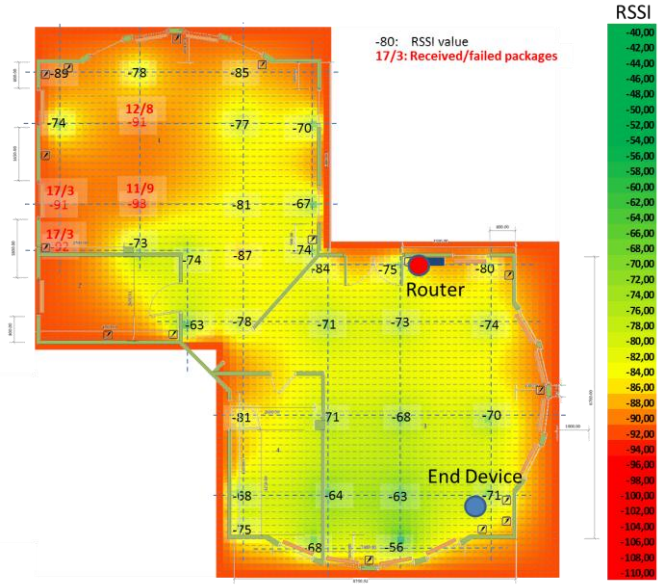



**Figure 27.** Measurements when the router is installed close to the end device.

In the next step the router is moved further from the computer to make better coverage. The selected point is the crossing point between the two rooms. The limitations that can occur during the installation are clear: the power outlet positions are fixed, and have big influence on forming the ZigBee network. Figure 28 displays this situation.

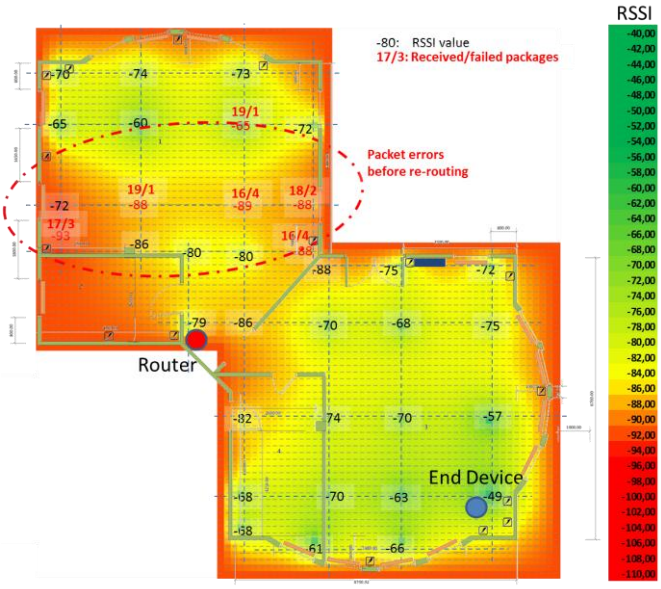



**Figure 28.** Packet errors before re-routing.

Moving the router away from the computer solved the coverage problem of the upper room. At bigger distances, some packet errors had been detected before the coordinator chosen another path through the router. After approximately 3-5 seconds with packer errors, the coordinator found the router as better transmission point. On Figure 29, the coverage after re-routing can be seen.

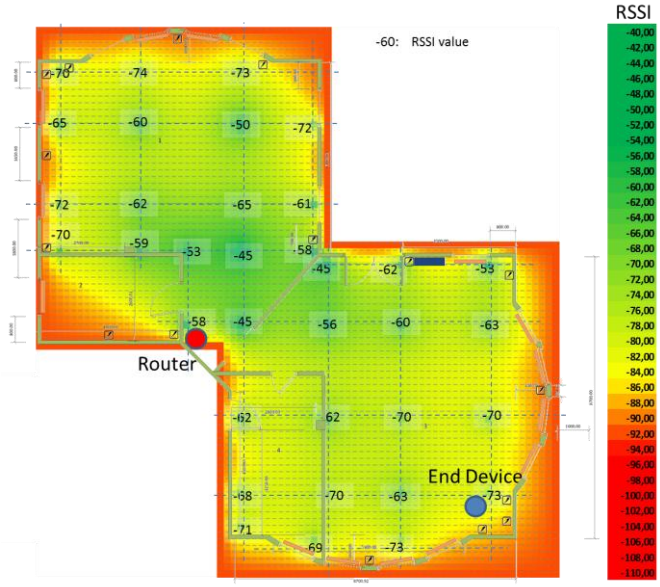



**Figure 29.** Floor covered with one router.

After the router is found, the coverage of the floor was successful. Compared to Figure 28, on Figure 29 lower RSSI values appeared around the end device as well. The reason for that is the indirect packet transmission between the Coordinator and the end device.

### 3.6.5.7. Further tests

In the following example, the network coverage tests were continued in the basement of the MFKK's office. During this installation, the detected weaknesses of the ZigBee network coverage could be improved with only one router. The example is showed due to its nature. Having a look on the floorplan (Figure 30), the area has to be covered shows its characteristic: The space is divided into two parts with a wall. This wall is an obstacle for the radio waves (Figure 30 top). Following the router installation procedure, the router should be placed in front of or behind the wall. But by placing the router to a point where the most of the space has a direct line of sight, the coverage can be solved easier (Figure 30 bottom).

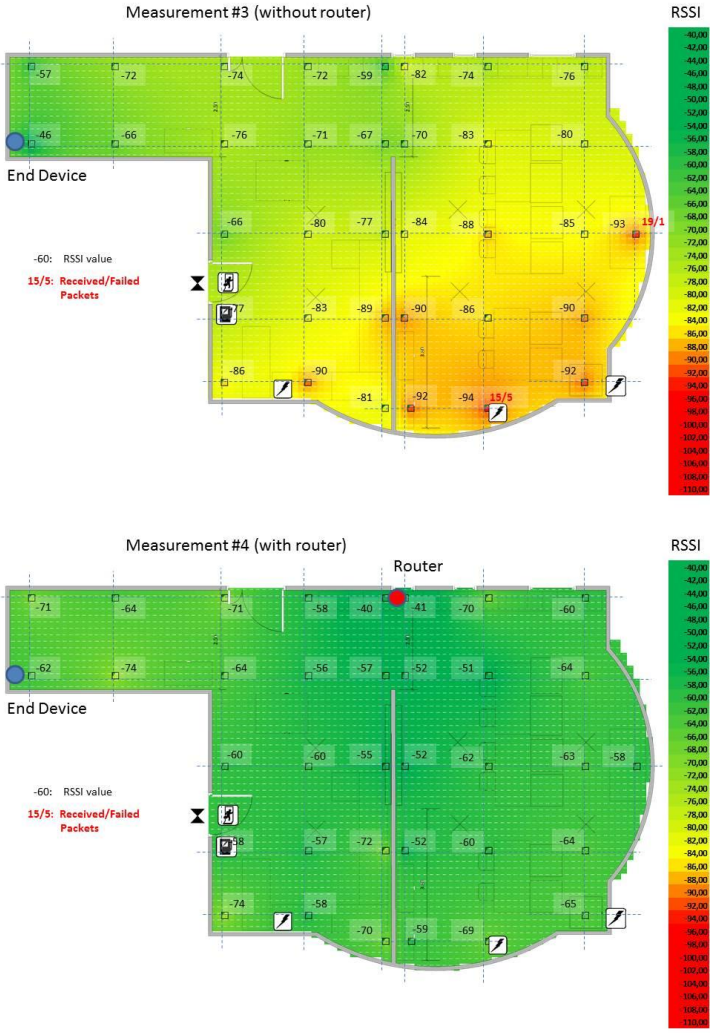



**Figure 30.** Router placed in a location with direct sight to the rooms.

During the pilots the receiver will not be the XStick, but the XBee Pro module. The receiver sensitivity difference is used during the installation to maintain a safety margin between the network check and the real use of the network during the pilot period. This difference has been

measured as the XBee Pro module (XBP24BZ7PIT-004J) module is used as a coordinator during the network check. The results can be compared on Figure 31.

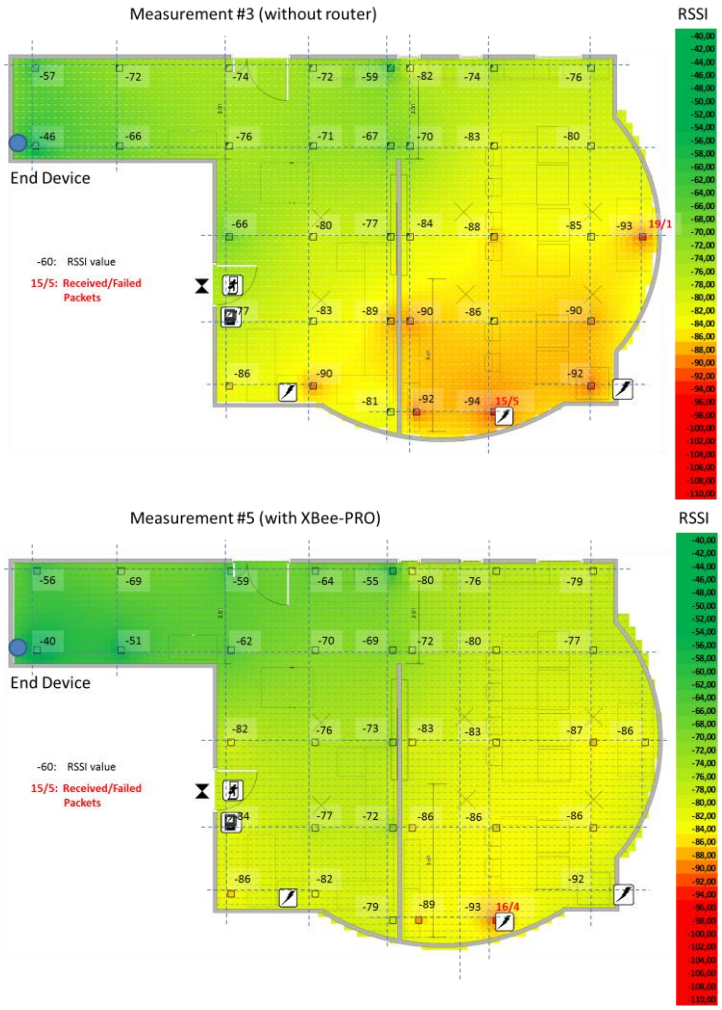



**Figure 31.** Measurements with the XStick (top) and the XBee Pro module (bottom).

The difference between the RSSI values of the two measurement data is plotted in Figure 32.

Even if it is planned to use the XStick for network check during the installation, the tests showed, that the RSSI values of the XBee-pro are not in all the cases higher than the results of the XStick. During the measurements, one location of the 35 was detected as critical from this point of view, but in the most cases the measurement proved that the use of the Xbee Pro adds a margin to the planned coverage, makes the system more reliable.

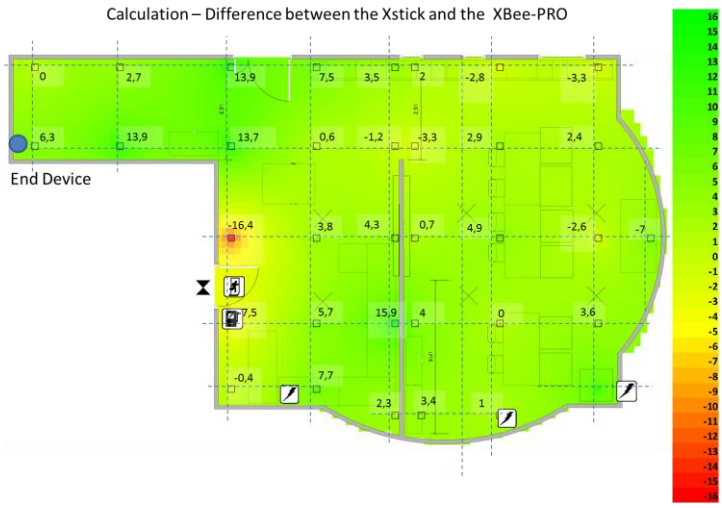



**Figure 32.** RSSI difference between the XStick and the XBee Pro.

### 3.6.6. Wireless infrastructure at nursing homes

In the FATE system at senior living facilities, the responsible to transmit, receive and manage the alarms generated for the fall detector and the bed sensor will be a real time locating, identifying and monitoring RFID (Radio Frequency Identification) technology-based solution. This section is devoted to the description of the components that support this solution.

Figure 33 shows the general architecture of the FATE system at nursing homes.

#### 3.6.6.1. Component description

The FATE system infraestructure at residences corresponding to the real time locating and identifying RFID (Radio Frequency Identification) technology-based solution is composed by the following elements:

- Eiris Server (locating and identifying software platform)
- Readers RF, LF Exciter and IR
- EDP (ElPas Display Panel)
- Infraesterucutre accessories (Junction Box, mounting bracket , etc..).

Figures 31 to 38 provide a detailed description of these components.

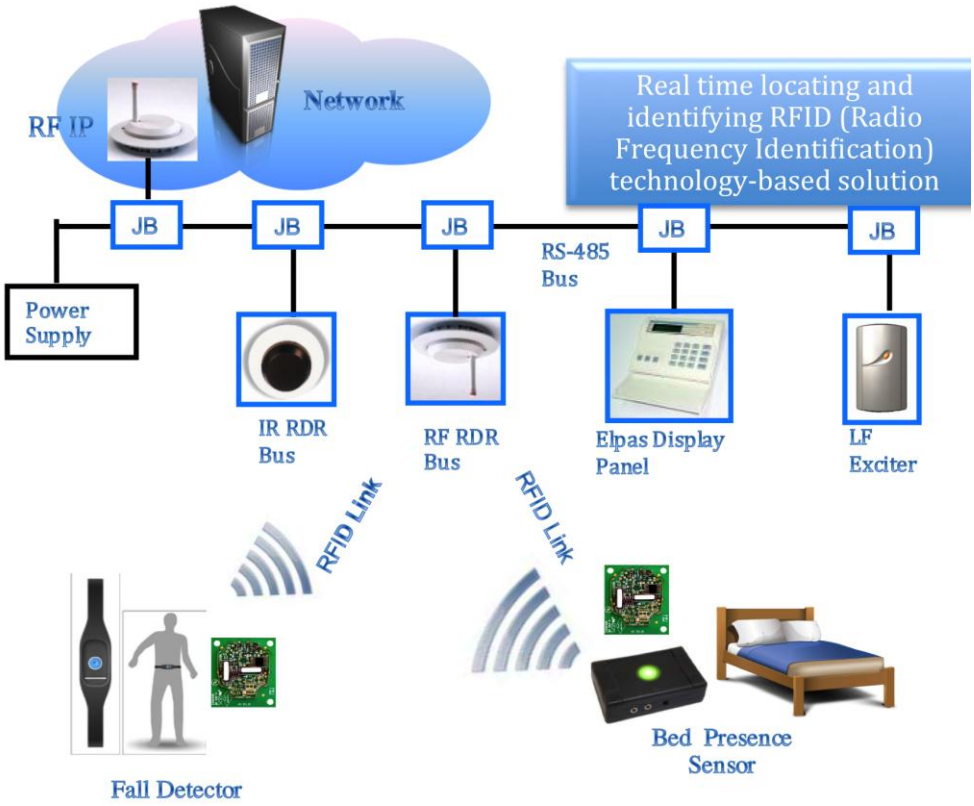



**Figure 33.** Overall architecture of the FATE system at nursing homes.


## EIRIS – For Security Monitoring & Control

EIRIS Visibility & Security Management Software from Visonic Technologies is an affordable, end-to-end solution; providing real-time monitoring, command and control in a single unified system for today's most demanding security and safety applications.

EIRIS is ideally suited for today's organizations that are seeking to effectively and reliably self-monitor and manage their security/safety operations. Out-of-the-box, EIRIS enables seamless integration of access control, intrusion detection, digital video surveillance, patient protection, duress alarm monitoring and asset tracking into one, easy to use, enterprise-class security platform.

### Function Summary

- Robust Client/Server architecture
- Intuitive Tools for alert Configuration
- Intuitive alarm handling screens
- Interactive, multi-layered graphical map displays
- Dynamic graphic monitoring & control
- Live CCTV alarm/event integration
- Automatic event/alarm recording
- Detailed event and tracking reports
- Integration with third-party applications
- Web and PDA client support
- Windows & XML based Web Services APIs

### Standard Features

**Network Configuration Tools**
EIRIS' intuitive configuration tools enable commissioning and database enrollment of all wired and wireless system devices without the need for programming experience.

**Automated Security Processes**
Enables user definable, rule-based events and alarms, (triggerable by location and status), for automating and enhancing monitoring and response processes.

**Dynamic graphic monitoring & control**
Supports real-time interactive, multi-layered graphical map displays for monitoring and controlling device status, system events and all security/safety alarms.

**Bundled Localization Editor**
EIRIS' bundled language localization editor customizes user interfaces and menus for different locales Supports Asian languages and right-to-left scripts (Hebrew and Arabic).

**Scalable System Expansion**
Modular, multi-site, client/server architecture supports easy system expansion without interruption of service as needs grow.

**Open APIs Extend Functionality**
Open Application Program Interfaces (APIs) offer real-time, bi-directional alarm/event processing and control with legacy security, life safety and back office systems.

**Integrated Incident Reporting**
EIRIS' integrated report generator creates historical reports of alarms, tag movements, device status and event times for enterprise-wide information delivery.

### EIRIS License Options

A complete line of licensable software configurations are available to meet the security requirements of any size organization. Whether needing to secure a small office or a large multi-site enterprise, organizations can start small and then upgrade capacity or system options as security or safety needs change

**Figure 34.** Basic Eiris software specification.

**Elpas**
*Active RFID/RTLS Solutions*

## RF IP Ceiling Reader

### Description

The Elpas RF IP Ceiling Reader is a supervised; 433MHz fixed receiving device. The reader is designed to detect, and relay real-time 'Location' and 'State' data from Elpas Active RFID Asset, Personnel or Infant Protection Tags to host applications.

The RF IP Ceiling Reader supports standard IT network communications and is easily integrated onto wired or wireless Ethernet/Wi-Fi networks to enable indoor, facility-wide monitoring and tracking of assets or personnel in real-time.

Architecturally attractive, the RF IP Ceiling Reader is easily surface mounted onto solid ceilings or flush mounted into dropped (false) ceilings. The reader supports large tag populations at read-distances up to 20m/65ft (360° coverage area) in open office environments and is remotely configurable for customized applications. On-board I/O ports enable the monitoring of one general purpose analogue input and control of two open-collector digital switched outputs.

The RF IP Ceiling Reader supports XML messaging technology for integration with external control and monitoring applications plus full-duplex data transmission with up to 15 Elpas RS-485 BUS devices.

Elpas RF IP Ceiling Reader

*Elpas RF IP Ceiling Reader – Confidential Technical and Product Specifications*

| Operational Specifications | |
|---|---|
| **Operating Frequency** | 432.92 MHz (868 MHz upon special request) |
| **Ethernet** | 10/100Base –TX (auto-sensing) |
| **Ethernet Specification** | Version 2.0 / IEEE 802.3, Ethernet II frame type, UDP protocol |
| **RS-485 BUS** | 230Kbit/sec |
| **Read-Range (Note 1)** | Installation grid: 20m/65ft radius |
| **Sensitivity** | -102dbm |
| **Tag Density** | Up to 125 tag messages/second |
| **Message Length** | 4-31 byte messages (encapsulated for messages > 4 bytes) |
| **Buzzer Indicator** | Upon power-up: Remotely configurable… ….Device Malfunction: Beeps continuously |
| **Green LED Indicator** | Upon power-up: Lites continuously |
| **Red LED Indicator** | Corrupted firmware- lights continously<br>Unregistered in EIRIS: Toggles on/off every second<br>Tag/Badge Detection: Flashes once per message |
| **Service Pin** | Generates service message |
| **Encoding** | Factory programmed ID |
| **Input/Outputs** | 1 dry contact analogue input 2 open collector digital outputs (up to 100mA) |
| **Power Requirements** | 16-28Vdc, 80mA at 24Vdc=2W |
| **General Specifications** | |
| **Mounting** | Ceiling flush mount |
| **Construction** | White polycarbonate plastic |
| **Dimensions** | 17cm x 4 cm (6.6 inches x 1.6 inches) |
| **Weight** | 200grams/7.054 ounces approximate |
| **Tamper Protection** | Open 'State' spring-loaded switch button |
| **Device Interfaces** | RF antenna: Female RP SMA connector<br>Ethernet: Female RJ-45 (8P8C) connector<br>RS-485 Bus & Power: Female RJ-11 (4P4C) or Four–position removable terminal block<br>Analogue Input: Two–position fixed terminal block Digital Outputs: Three–position fixed terminal block |
| **Operating Environment** | Temp: -10°C to 70°C (14°F to 159°F); Humidity: 20% to 80% non-condensing |
| **Storage Temperature** | -40° to 70°C (-40°to 159°F) |
| **Remote Configuration and Supervision** | EIRIS 4.6.3 (or higher) software |
| **Compliance Standards** | FCC: FCC PART 15, Sub-part B, Class B CE: EN60950-1, CAN/CSA-CEI/ICE CISPR 22 IC: ICES-003 |
| **Warranty** | 1 year limited warranty |

**Figure 35.** Basic reader RF IP specification.

Elpas
Active RFID/RTLS Solutions

## LF Exciter

### Description

The Elpas Low Frequency (LF) Exciter from Visonic Technologies is a supervised, short-range, UHF emitter that adds pin-point detection functionality to any Elpas safety, security or monitoring installation.

The LF Exciter is engineered to emit spherical, low-power electromagnetic fields (125 KHz) of up to 3 m (10 ft) in radius. These harmless LF fields are user-adjustable, so that they can be tuned to precisely cover most indoor doorways or entrance/exit areas. So, whenever an individual or asset bearing an Elpas active RFID tag enters the electromagnetic field generated by the exciter, the mobile tag is prompted to transmit its pre-programmed data messages (including the exciter's Neuron ID). The messages are instantly detected by the Elpas RF reader infrastructure and are locally processed and/or relayed over the LonTalk network to the EIRIS server machine in support of the configured Elpas application.

Elpas LF Exciter

*Elpas LF Exciter – Confidential Technical and Product Specifications*

| | |
|---|---|
| **LF Specifications** | |
| **Technology** | Low frequency electromagnetic field (125 KHz) |
| **Effective Range** | Up to 3 m (10 ft) radius (spherical field) |
| **Transmission Rate** | Continuous bursts of LF transmissions (each about 12 ms in duration) |
| **Output and Format** | 3-byte messages (preamble, exciter ID and CRC) |
| **Output Power** | Less than 60 dbμv at 30 m (100 ft); adjustable, using the onboard trim control potentiometer |
| **Output Bit Rate** | 2,000 bit per second |
| **RF Specifications** | |
| **Technology** | UHF RF (433.92 MHz) |
| **Effective Range** | 20 m (65.5 ft) radius (360° coverage area) |
| **Transmission Rate** | 1 RF transmission (about 2 ms in duration), 10 seconds apart |
| **Electrical Specifications** | |
| **Status Inidcators** | Power On: Red LED blinks 5 times upon power on, then lights constantly<br>Invalid ID Code: Red LED blinks continuously; buzzer beeps repetitively |
| **ID Address** | Set by an onboard, 8-position DIP switch |
| **Power Requirements** | 24 V DC nominal ± 30%; 200 mA |
| **Power Consumption** | Approximately 2W; power consumption is a function of address (according to the DIP switch settings, where FF is the maximum and 00 is the minimum) |
| **General Specifications** | |
| **Construction** | Polycarbonate plastic |
| **Dimensions (H x W x D)** | 17 x 4 cm (6.6 x 1.6 inches) |
| **Weight** | Approximately 200 grams (7.054 ounces) |
| **Device Interfaces** | RJ-11 (6P6C) power source connector<br>RJ-45 (8P8C) master-slave connector |
| **Operating Environment** | Temperature: -10°C to 70°C (14°F to 159°F)<br>Humidity: 20% to 80%, non-condensing |
| **Remote Management** | EIRIS 4.5 (or higher) enterprise software |
| **Compliance with Standards** | FCC: FCC ID: 04X5-RLE00125, FCC PART 15, Sub-part B, Class B, Sub-part C;<br>CE: EN300220-1, EN300220-2, EN300330-1, EN300330-2, EN301489-1, EN301489-3, EN60601-1-2, CISPR 11, EN60950-1, IEC60601-1, IEC 61000-4-2, IEC 61000-4-3, IEC 61000-4-4, IEC 61000-4-6 |
| **Warranty** | 1-year limited warranty (excluding battery) |

**Figure 36.** Basic reader LF specification.

Elpas
Active RFID/RTLS Solutions

## IR BUS Ceiling Reader

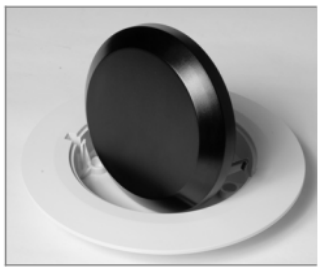
### Description

The Elpas High-Resolution IR BUS Ceiling Reader from Visonic Technologies is a supervised, fixed infrared locating device. The reader is designed to detect and relay sub-room 'Location' and 'State' data in real-time from Elpas Active RFID IR-Enabled Asset, Personnel or Infant Protection Tags to host applications.

The reader can be easily integrated anywhere onto wired or wireless Ethernet/Wi-Fi networks (using an Elpas RF IP Reader as a RS-485 BUS master) to enable indoor facility-wide sub-room monitoring and tracking of assets or personnel in real-time.

Architecturally attractive, the reader is easily surface mounted onto solid ceilings or flush mounted into dropped (false) ceilings. The Elpas High-Resolution IR BUS Ceiling Reader supports large tag populations at read-distances up to 15m/50ft (360° coverage) and is remotely configurable for customized applications. Onboard I/O ports enable the monitoring of one general purpose analog input and control of one open-collector digital switched output.

The Elpas High-Resolution IR BUS Ceiling Reader also supports XML messaging technology (via the Elpas RF IP Reader) for integration with external systems plus full-duplex data transmission with up to 15 Elpas RS-485 BUS devices.



Elpas IR BUS Ceiling Reader

*Elpas IR BUS Ceiling Reader – Confidential Technical and Product Specifications*

| Operational Specifications | |
|---|---|
| Operating Frequency | 455KHz |
| Read Range (Note 1) | Installation grid: 15 m / 50 ft radius |
| Tag Density | Up to 50 tag messages per second |
| Message Lenght | 4–31 byte message (encapsulated for messages > 4 bytes) |
| Buzzer Indicator | Upon power-up: beeps once<br>Device malfunction: beeps continuously |
| Visual Indicators | Corrupted firmware: red LED lights continuously<br>Unregistered in EIRIS: red LED blinks every second<br>Tag/Badge detection: flashes once per message |
| Service Pin | Generates service message |
| Encoding | Factory-programmed ID |
| RS-485 BUS | 230Kbit/sec |
| Input / Output | 1 analog input, 1 open collector digital outputs (up to 100 mA) |
| Power Requirements | 16–28 Vdc/50mA |
| **General Specifications** | |
| Mounting | Surface-mounted on solid walls and ceilings, flush-mounted into dropped (false) ceilings |
| Construction | Black poly carbonate plastic |
| Dimensions (H x W x D) | 17 x 4 cm (6.6 x 1.6 inches) |
| Weight | Approximately 200 grams / 7.0 ounces |
| Tamper Protection | Open 'State' spring loaded switch button |
| Device Interfaces | RS-485 Bus & Power: Female RJ-11 (4P4C) or Four–position removable terminal block<br>One Analog Input: Two-position fixed terminal block<br>One Digital Output: Two-position fixed terminal block |
| Operating Environment | Temp: -10°C to 70°C (14°F to 159°F); Humidity: 20% to 80% non-condensing |
| Storage Temperature | -40° to 70°C (-40°to 159°F) |
| Remote Management | EIRIS 4.6.3 (or higher) software |
| Compliance Standards | CE: EN300220-1, EN300220-2, EN60950-1, ICES-003, CAN/CSA-CEI/ICE CISPR 22<br>FCC: FCC PART 15, Sub-part B, Class B FCC ID: O4X5-IRB0088<br>IC: 1467G IRB00880 |
| Warranty | 1-year limited warranty |

**Figure 37.** Basic reader IR specification.

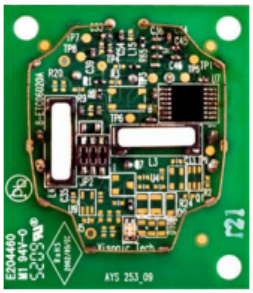
Elpas
Active RFID/RTLS Solutions

## Wireless Input OEM Module

### Description

The Elpas Wireless Input Module is an embedded PCB board OEM connectivity solution. The module provides a 433MHz, wireless interface to the Elpas RTLS fixed network infrastructure for the remote monitoring of single or multi-level staff/patient/resident call points, pull switches or any other ancillary facility monitoring points.

The module monitors and reports on 'State' changes of up to three analog inputs in real-time to a host Elpas RTLS application or to other third-party staff and patient care monitoring solutions. The LF-enabled version (P/N 5-ETC90010-1) also includes a low frequency LF receiver (125KHz) that enables real-time location detection that can be used to protect against the unauthorized removal of the call point from the area being monitored.



Elpas Wireless Input OEM Module
(actual PCB may vary in appearance)

| Specifications | |
|---|---|
| **Radio Technology** | RF (433.92 MHz) and LF (125KHz) |
| **Transmission Range** | 20 meters / 65 Ft (360° coverage area) |
| **Transmission Rate** | Every 60 seconds |
| **LF Sampling** | Every 150ms |
| **Device Inputs/Outputs** | **Inputs:** 3 analog contacts....**Output:** ACK signal |
| **Power Requirements** | 3.3VDC +/-30%; 285mAh |
| **Power Sources** | On-board lithium battery, (type CR2430 - supplied) or via host call/monitoring point |
| **Battery Life** | Approximately 5 years |
| **Module ID** | Unique factory programmed ID |
| **Construction** | PCB board (no enclosure) |
| **Dimensions (H x W x D)** | 39 x 35 x 10 mm / 1.5 x 1.4 x 0.64 inches |
| **Weight** | 9 grams/ 0.75 ounces (with battery) |
| **Device Interface** | 6-pin low profile connector for: input power, 3 dry contact inputs/ ACK output signal |
| **Operating Environment** | **Temp:** -10°C to 70°C (14°F to 159°F); **Humidity:** 20% to 80% non-condensing |
| **Compliance Stanards** | **FCC:** GSA-ETC00433 .....**IC:** 1467G-ETC00433.....**CE:** EN 300 220-1, EN 300 220-2, EN301 489-1, EN301 489-3, EN 60601-1-2, CISPR 11: 2004, IEC 61000-4-2, IEC 61000-4-3 |
| **Warranty** | 1 year limited warranty (excluding battery) |

**Specifications - Subject to change without notice**

### Ordering Information

| Part Number | Description |
|---|---|
| **5-ETC90010-1** | Wireless Input OEM, RF/LF, 433Mhz, (Pack of 10) |
| **5-ETC90010-2** | Wireless Input OEM, RF, 433Mhz, (Pack of 10) |
| **5-BC012430** | 3V/270mAh Lithium Battery, CR2430 (25 pcs) |

**Figure 38.** Wireless input OEM module overview.

Elpas
Active RFID/RTLS Solutions

## Elpas Display Panel

### Description

The Elpas Display Panel (EDP) is a wall mountable, remote management panel that enables users to view and clear individual system alerts and to manage the status of specific Elpas Active RFID tags without the need to access a host RTLS client station.

The EDP contains a four-line, twenty-character LCD for displaying real-time tag and alert status information and a keypad for managing open alerts or tag status. The device has an audible buzzer for indicating the receipt of a new alert or of device malfunction and supports one analog device input and the control of one open-collector digital output.

The BUS Version of the EDP (P/N 5-EDP00485) is powered and communicates via RS-485 while the IP Version of the device (P/N 5-EDP00485-1) supports Ethernet network communications for relaying data to and from a host RTLS application.

The Elpas Display Panel is remote configurable using Eiris Tracking & Management or ELC Programmer Software and supports data transmission with up to fifteen other Elpas BUS devices such as RF or IR Readers, I/O Modules, LF Exciters and/or Proximity Readers.

EDP – Closed Panel View

EDP – Open Panel View

*Elpas Display Panel (EDP) – Confidential Technical and Product Specifications*

| Specifications | |
|---|---|
| **Display** | 4 line-20 character illuminated LCD |
| **Ethernet – IP Version** | 10 Base –TX (auto-sensing) / UDP protocol |
| **RS-485 BUS** | 230Kbit/sec |
| **Green LED Indicator** | Lights continuously when powered |
| **Red LED Indicator** | **Corrupted Firmware:** Lights continuously **Unregistered in Software:** Toggles on/off every second |
| **Audible Indicator** | **Power-Up:** User configurable…….**Device Malfunction:** Beeps continuously<br>**Incoming Alerts:** User configurable |
| **Power Requirements** | 12–28 VDC **BUS Version**: 60mA……**IP Version:** 100mA |
| **Construction** | White polycarbonate plastic |
| **Dimensions (H x W x D)** | 11 x 16 x 2.5 cm (4.3 x 6.3 x 1.0 inches) |
| **Weight** | 260 grams (9.0 ounces) |
| **Keypad** | Slim-line landscape keypad |
| **Device Interfaces – IP Version** | **Ethernet:** Female RJ-45 (8P8C) connector<br>**Digital Input/Output:** Three-Position terminal block<br>**Power:** Two-Position terminal block |
| **Device Interfaces – BUS Version** | **RS-485 Data Bus & Power:** 2 Female RJ-11 (6P6C) connectors<br>**Digital Input/Output:** Three-Position terminal block<br>**Power:** Two-Position terminal block |
| **Encoding** | Factory programmed ID |
| **Input/Output** | 1 analog input 1 open collector digital output (up to 100mA) |
| **Operating Environment** | **Temperature:** -10°C to 70°C (14°F to 159°F) **Humidity:** 20% to 80%, non-condensing |
| **Storage Temperature** | -40° to 70°C (-40°to 159°F) |
| **Remote Management** | ELC Programmer Software (V2.0 or higher)….EIRIS 4.7 (or higher) Enterprise Software |
| **Compliance Standards** | CE, FCC, IC Compliant |
| **Warranty** | 1-year limited warranty |

**Figure 39.** EDP display panel basic specifications.

Elpas
Active RFID/RTLS Solutions

## RS-485 Junction Box

**Description**

The Elpas RS-485 Junction Box is a multi-drop, solid state fixed network accessory that provides installers with a convenient power/data interface for linking Elpas RS-485 BUS devices together.

Each box contains four female RJ-11 (6P6C) modular jacks that support Elpas Network Drop Cables, for the easy connection of Elpas BUS devices. The device has two loop-through 8 pin 110 punch-down block connectors to enable multiple RS-485 Junction Boxes to be connected in a daisy-chain wiring configuration via CAT-5 cabling.

For powering the RS-485 BUS, the junction box contains a two–position removable terminal block for connecting to an Elpas PS60 Power Supply or to other compatible third-party16-28VDC/2.5A power sources. The junction box also has a two–position terminal block for BUS end of-the-line termination and a green LED Power ON status indicator.

Constructed of white polycarbonate plastic, the RS-485 Junction Box can be surface mounted onto solid walls and ceilings or easily located above dropped (false) ceilings.


Elpas RS-485 Junction Box

*Elpas RS-485 Junction Box – Confidential Technical and Product Specifications*

| Specifications | |
|---|---|
| **Total Pass Through Current** | 2.5 Amps |
| **RS-485 BUS** | 230Kbit/sec |
| **RS-485 Backbone Interface** | Two eight–pin 110 punch down blocks |
| **RS-485 Bus Device Interfaces** | Four RJ-11 (6P6C) female network drop connectors (for RS-485 data & 16-28 VDC) |
| **Power Interface** | Two–position removable terminal block (for 16-28 VDC) |
| **End of-the-Line Termination** | Two–position terminal block |
| **Green LED Indicator** | Upon power-up, lights continuously |
| **Construction** | White polycarbonate plastic |
| **Dimensions (H x W x D)** | 110 x 63 x 26mm (4.3 x 2.5 x 1.0 inches) |
| **Weight** | 70 grams/ 2.5 ounces |
| **Operating Environment** | **Temp**: -10°C to 70°C (14°F to 159°F) **Humidity:** 20% to 80% non-condensing |
| **Warranty** | One year limited warranty |

**Figure 40.** RJ-485 junction box basic specifications.

Elpas
Active RFID Solutions

**Technical Specifications**
**Reader Mounting Bracket**

**Description**

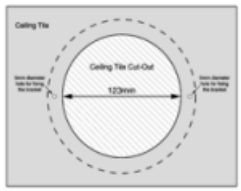
The Reader Mounting Bracket is used to flush mount Elpas LF Exciters or Location Readers in suspended ceiling tiles.

The bracket is constructed of coated steel and has a 15mm/0.59inch center hole to allow cables to pass through. The bracket also contains two 4mm/0.16inch diameter threaded holes; (83mm/3.26 inches apart) for clamping the base of the device into place, plus two 5mm/0.2 inch holes at each end of the bracket for fixing the bracket to the ceiling tile.

Two M4 x 35mm Phillips head screws are included per bracket.

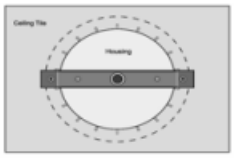
**Mounting Information**

1. Cut a 123mm (4.8 inch) diameter hole n the tile where the device is to be flush mounted.

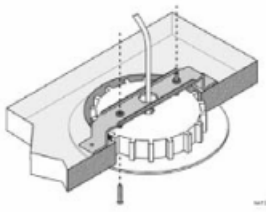



2. Pull the applicable cable(s) through the cable entry hole of the bracket and into the base of the reader or exciter.

3. Insert the bracket through the mounting hole placing it on top of the ceiling tile. Position it so that the bracket is firmly supported by the ceiling tile.




4. Insert the base of the reader/ exciter into the mounting hole. Next, screw the base to the mounting bracket place using the 2 supplied M4 x 35mm Philips head screws. Ensure that the screw heads are recessed inside the screw holes. If not the device will not close properly and damage may occur.



5. Connect the cable(s) to the reader/exciter; then insert the device back into its base.

**Figure 41.** Mounting bracket description and short mounting information.

## 3.7. i-Walker

### 3.7.1. i-Walker description

i-Walker is a robotic rollator that integrates sensors and actuators. It uses a standard walker frame modified for this purpose. Actuators are two hub motors integrated in the rear wheels and are used for breaking or helping the user. Sensors are arranged in the frame to detect forces, tilt and movement. Power is supplied by an integrated battery. Finally, a network of distributed microcontrollers drives the system and records and provides information to the therapists.

Four main services are provided by the i-Walker platform. Three are related to elder /impaired assistance. The fourth is used for data logging. All the assistance should be planned by a physiotherapist. Services provided are:

- Active motor assistance to compensate lack of muscle force on climbs.
- Active break assistance to compensate lack muscle force on descents.
- Active differential assistance to compensate unbalanced muscle force.
- Recording of sensor measurements and actuators activities for later evaluation (left and right hand forces, normal forces, tilt and odometry).

When in daily use, i-Walker needs periodic charge of the batteries. Additionally to software services, a luminous indicator shows the state of the batteries.

For every new patient, i-Walker assistance must be configured by physiotherapist. i-Walker can be configured by setting two main parameters $\upsilon$ and $\lambda$. The parameter $\upsilon$ is an offset that can be set to create a permanent resistance or pulling force to the i-Walker. The parameter $\lambda$ is a gain applied to the forces done by the user. The combination of both parameters allows the therapists to create a patient's tailored configuration.

### 3.7.2. Hardware architecture

i-Walker is based on a standard rollator frame improved with sensors and actuators. The following components are used in the i-Walker construction:

***Mechanical components:***

- A standard rollator frame sized 500mm (W) x 600mm (L) x 850mm (H).
- Two 150W hub motors, 100mm diameter, to be embedded on the rollator rear wheels.
- Two modified handlebars with brake handle and force measurement.

***Sensors/electronics:***

- 32 strain gauges mounted in 8 bridges to measure handlebar forces (X-Y-Z) and normal wheels forces (F).
- PGA signal conditioners for strain gauge measurements.
- Battery packs providing a minimum 4h autonomy (in continuous usage).
- DC-DC voltage converters to provide energy to the different modules.
- A circuit accelerometer for inclination measurement.
- A CAN bus communications network to connect all the microcontroller boards.

***Computers:***

- On-top Raspberry Pi computer based on an ARM-6 processor, with 512MB RAM, UART, 2 USB connectors, Ethernet, HDMI and microSD socket.
- 8 microcontroller boards based on Microchip DSPIC 30f4011 for i-Walker control: 2 motor controllers, 2 handlebar controllers, 2 PGA controllers, 1 battery controller and tilt sensor and 1 bridge for communications.
- 1 communications interface computer based on a Raspberry PI board enabling Ethernet /Bluetooth / WIFI interfaces for i-Walker.

Figure 42 shows i-Walker computer resources and used networks.

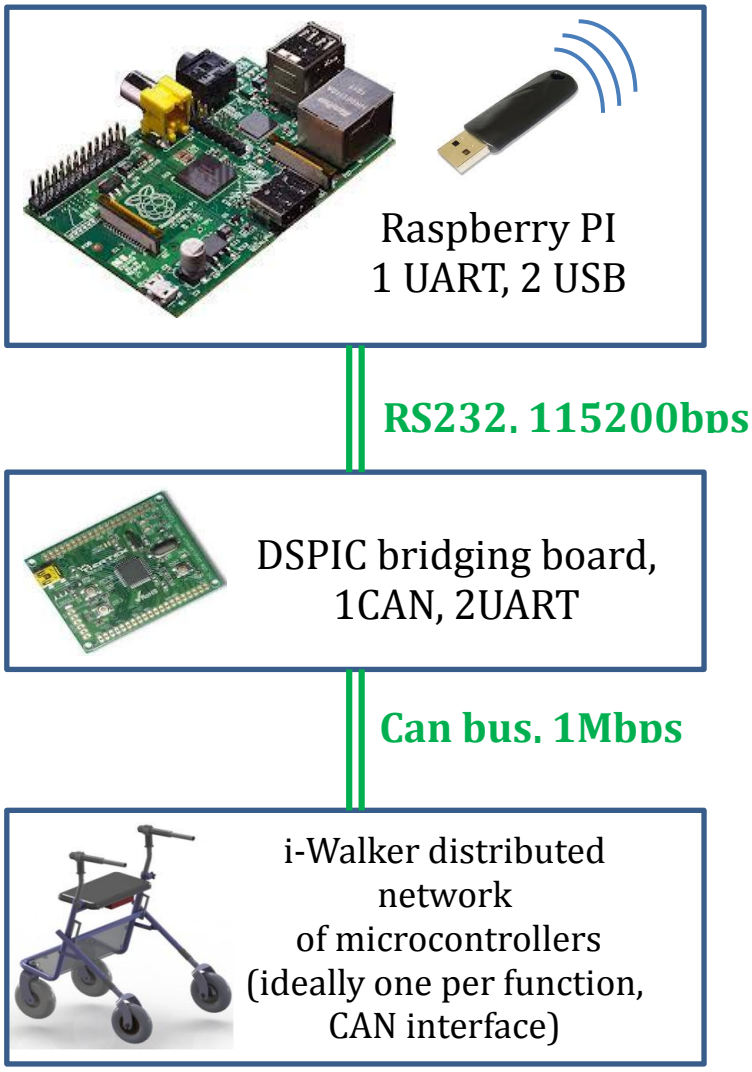



**Figure 42.** i-Walker computer architecture.

### 3.7.3. Software architecture

On-Top computer is an ARM-6 Raspberry Pi board, running a Debian based operating system, and executing communication scripts to bridge WIFI, Ethernet or Bluetooth interfaces.

The 8 Microchip microcontroller based boards are programmed in C Language using C18 compiler integrated on MPLAB environment.

Each board has specific functionalities: AD conversion, SPI interface, digital interfaces, PWM outputs, and a common bus CAN interface. Software routines for all these functionalities are implemented in a common C library resource and linked when necessary.

All the boards implement watchdog services to avoid hangout problems and answer to CAN bus status petitions.

The control strategies, user assistance and data recovery tasks are distributed along the network of microcontrollers. Figure 43 shows the CAN Network implemented on i-Walker.

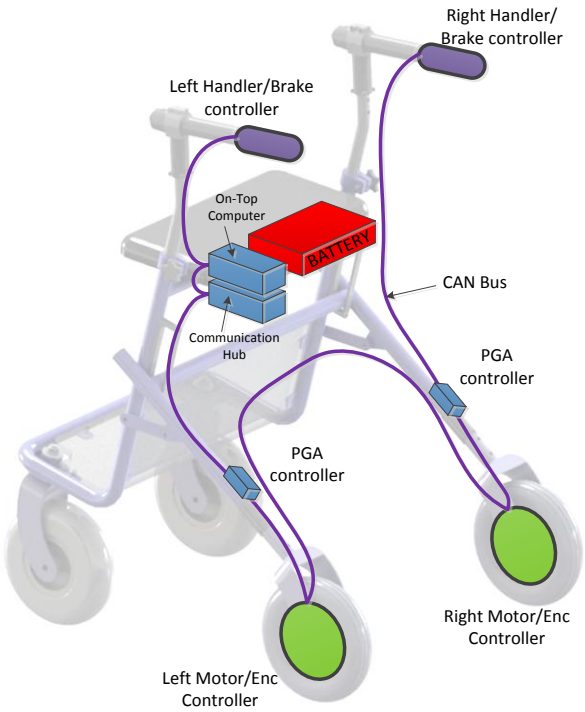



**Figure 43.** CAN network implemented on i-Walker.

### 3.7.4. Internal i-Walker communication: CAN bus

For our platforms, the Frame Identifiers are built taking into account the following design considerations:

1) Messages with variables are identified by means of a unique CAN Frame ID. In order to address a particular message, a specific Frame ID must be filled following this structure:

| Frame ID bits |
|---|

ttt nnn mmmmm

*ttt*: Node Type (motor controller, force sensors, etc.)
*nnn*: Node Identification Number (motor controller 1, motor controller 2, etc.)
*mmmmm*: Message Identifier (speed setpoint message, odometry message, tilt message, etc.)

2) In our Frame ID hierarchy, we have reserved a set of identifiers with maximum priority, just in case that some special functions need to be performed by the prototypes in future upgrades. These set of identifiers have Node Type '000' and Node ID '000'; this yields a total amount of 32 high priority frames.

Furthermore, a thorough specification of the actual messages sent by the nodes can be seen in Table 20.

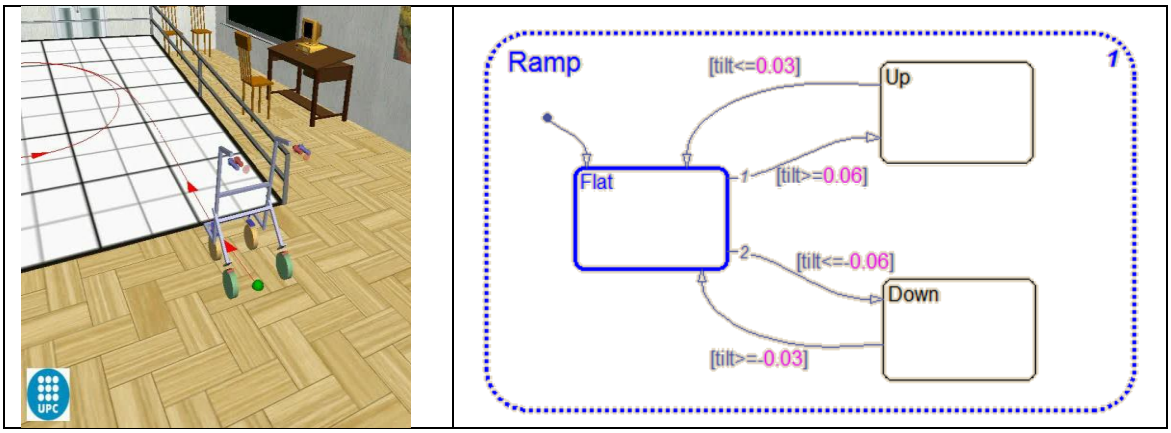
**Table 20**. CAN bus messages.

| Electronics Board | CAN Frame ID | | | Data | | | | | | | | Data Length (Bytes) | Period (ms) | Data Rate (Kbps) | BW usage (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Node Type | Node ID | Data ID | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | | | | |
| 4PGAs left | 3 | 0 | 10 | Force X High | Force X Low | Force Y High | Force Y Low | Force Z High | Force Z Low | Force Normal High | Force Normal Low | 8 | 2 | 54 | 5.4% |
| 4PGAs right | 3 | 0 | 11 | Force X High | Force X Low | Force Y High | Force Y Low | Force Z High | Force Z Low | Force Normal High | Force Normal Low | 8 | 2 | 54 | 5.4% |
| ACCEL | 3 | 0 | 15 | Acceleration 1 High | Acceleration 1 Low | Acceleration 2 High | Acceleration 2 Low | Angle 1 High | Angle 1 Low | Angle 2 High | Angle 2 Low | 8 | 1 | 108 | 10.8% |
| | 3 | 0 | 20 | Pose X1 High | Pose X1 Low | Pose Y1 High | Pose Y1 Low | Pose Z1 High | Pose Z1 Low | Pose A1 High | Pose A1 Low | 8 | 100 | 1.08 | 0.11% |
| | 3 | 0 | 21 | Pose E1 High | Pose E1 Low | Pose R1 High | Pose R1 Low | Pose X2 High | Pose X2 Low | Pose Y2 High | Pose Y2 Low | 8 | 100 | 1.08 | 0.11% |
| | 3 | 0 | 22 | Pose Z2 High | Pose Z2 L2w | Pose A2 High | Pose A2 Low | Pose E2 High | Pose E2 Low | Pose R2 High | Pose R2 Low | 8 | 100 | 1.08 | 0.11% |
| MIX-IT left | 3 | 0 | 0 | Speed High | Speed Low | Odometry Ticks MSB | Odometry Ticks | Odometry Ticks | Odometry Ticks LSB | Controller Temperature | Interface inputs | 8 | 2 | 54 | 5.4% |
| | 5 | 0 | 2 | Speed mrad/s$^2$ High | Speed mrad/s$^2$ Low | CPU Usage (%) | | | | | | 3 | 2 | 34 | 3.4% |
| MIX-IT right | 3 | 0 | 1 | Speed High | Speed Low | Odometry Ticks MSB | Odometry Ticks | Odometry Ticks | Odometry Ticks LSB | Controller Temperature | Interface inputs | 8 | 2 | 54 | 5.4% |
| | 5 | 1 | 2 | Speed mrad/s$^2$ High | Speed mrad/s$^2$ Low | CPU Usage (%) | | | | | | 3 | 2 | 34 | 3.4% |
| Gate-it | 5 | 0 | 1 | Left Motor Setpoint Type | Left Motor Speed Setpoint High | Left Motor Speed Setpoint Low | Left Motor Current Setpoint High | Left Motor Current Setpoint Low | Left Motor P. Braking Setpoint | | | 6 | 2 | 46 | 4.6% |
| | 5 | 1 | 1 | Right Motor Setpoint Type | Right Motor Speed Setpoint High | Right Motor Speed Setpoint Low | Right Motor Current Setpoint High | Right Motor Current Setpoint Low | Right Motor P. Braking Setpoint | | | 6 | 2 | 46 | 4.6% |
| | 3 | 0 | 3 | Left Wheel RPM High | Left Wheel RPM Low | Right Wheel RPM High | Right Wheel RPM Low | Distance Travelled (mm) High | Distance Travelled (mm) Low | CPU Usage (%) | Battery Level (%) | 8 | 2 | 54 | 5.4% |
| | 3 | 0 | 4 | Pose X High | Pose X Low | Pose Y High | Pose Y Low | Pose Z High | Pose Z Low | Pose PSI High | Pose PSI Low | 8 | 2 | 54 | 5.4% |
| | 3 | 0 | 5 | Left Wheel Speed High | Left Wheel Speed Low | Right Wheel Speed High | Right Wheel Speed Low | Left Wheel Theta High | Left Wheel Theta Low | Right Wheel Theta High | Right Wheel Theta Low | 8 | 2 | 54 | 5.4% |
| | 3 | 0 | 6 | Left Castor Speed High | Left Castor Speed Low | Right Castor Speed High | Right Castor Speed Low | Left Castor Theta High | Left Castor Theta Low | Right Castor Theta High | Right Castor Theta Low | 8 | 2 | 54 | 5.4% |
| | 3 | 0 | 7 | Left Castor PSI' High | Left Castor PSI' Low | Right Castor PSI' High | Right Castor PSI' Low | Left Castor PSI High | Left Castor PSI Low | Right Castor PSI High | Right Castor PSI Low | 8 | 2 | 54 | 5.4% |
| Reactive Control | 3 | 0 | 12 | Left Lambda (%) | Right Lambda (%) | Left Nu (dN) | Right Nu (dN) | | | | | 4 | 10 | 7.6 | 0.76% |
| | 3 | 0 | 13 | Motors Setpoint Type | Longitudinal Speed Setpoint High | Longitudinal Speed Setpoint Low | Rotational Speed Setpoint High | Rotational Speed Setpoint Low | Left Motor Passive Brake % Setpoint | Right Motor Passive Brake % Setpoint | Experiment Status | 8 | 10 | 10.8 | 1.08% |
| | 3 | 0 | 14 | Left Motor Current Setpoint High | Left Motor Current Setpoint Low | Right Motor Current Setpoint High | Right Motor Current Setpoint Low | Left Motor Active Brake Current Setpoint High | Left Motor Active Brake Current Setpoint Low | Right Motor Active Brake Current Setpoint High | Right Motor Active Brake Current Setpoint Low | 8 | 10 | 10.8 | 1.08% |

### 3.7.5. Computational methods

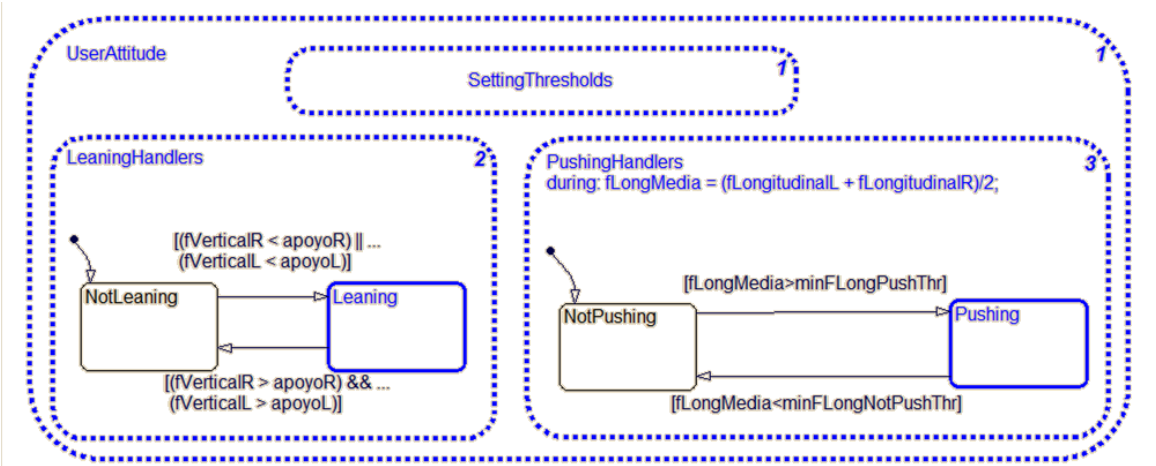
A number of algorithms are distributed between the microcontroller network. Exchanged data between CAN Bus nodes is used to make algorithm calculations.

***Ramp detection***



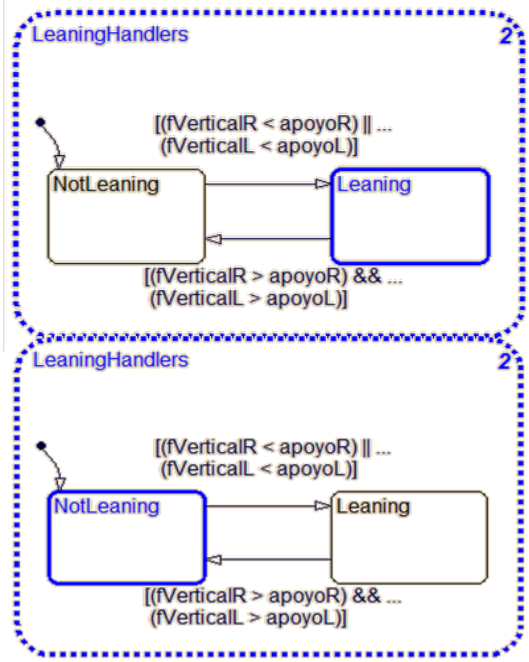

***User Attitude detection***

The following methods are used to detect if the user is interacting with i-Walker

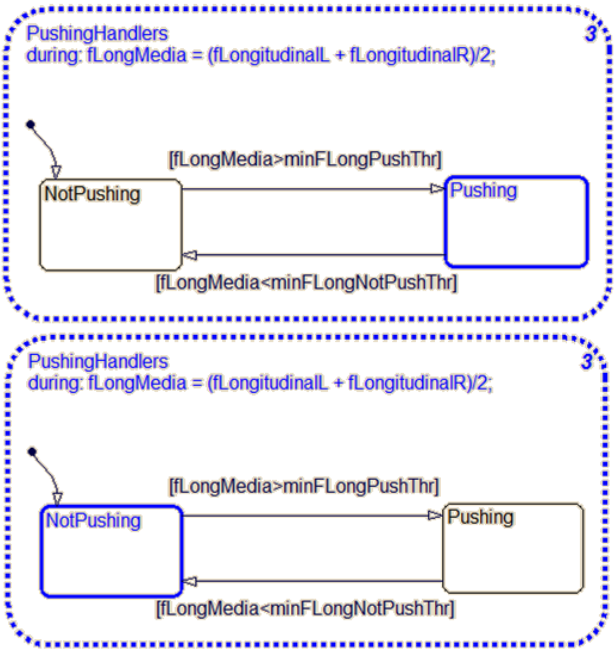


More in detail:

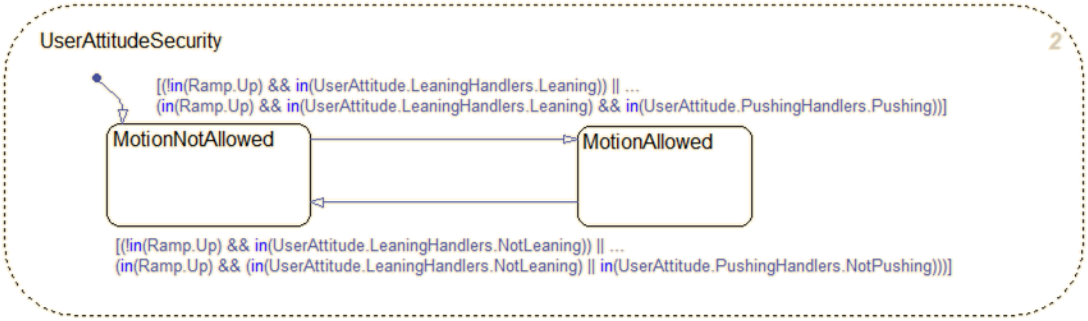
*Leaning detection*




*Pushing detection*

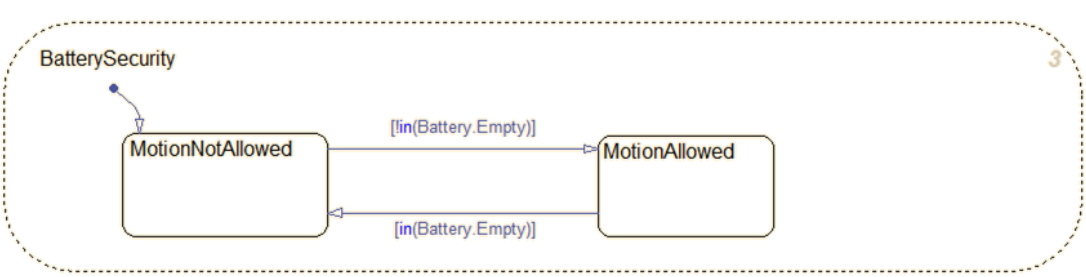


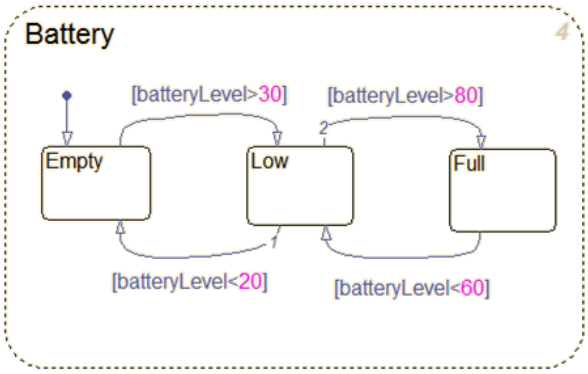

*Security brake*



***Battery level detection***

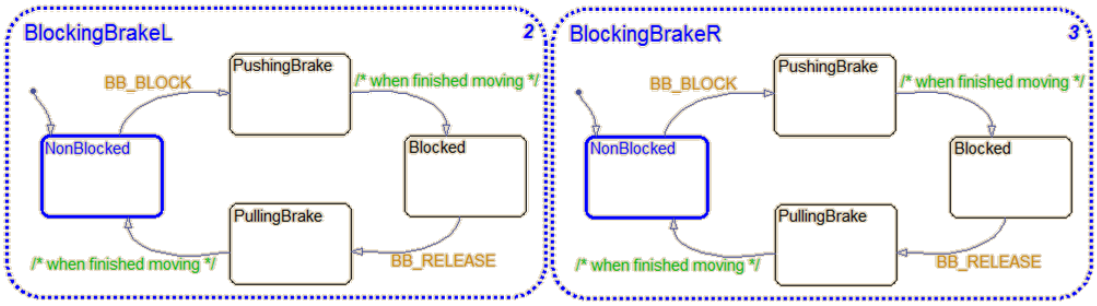
i-Walker will be inoperative when battery discharges under a security level.



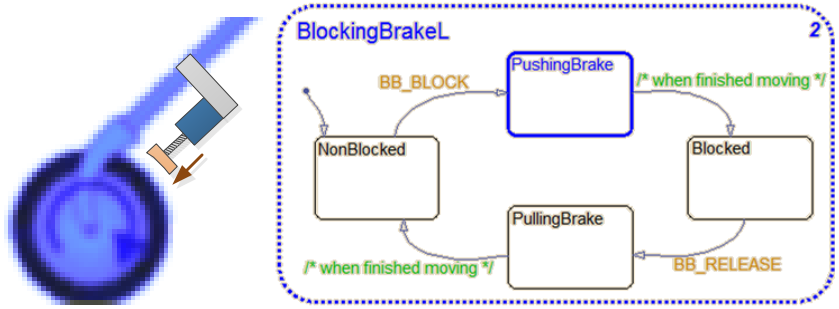




***Braking system***

Braking system will block or release i-Walker wheels.



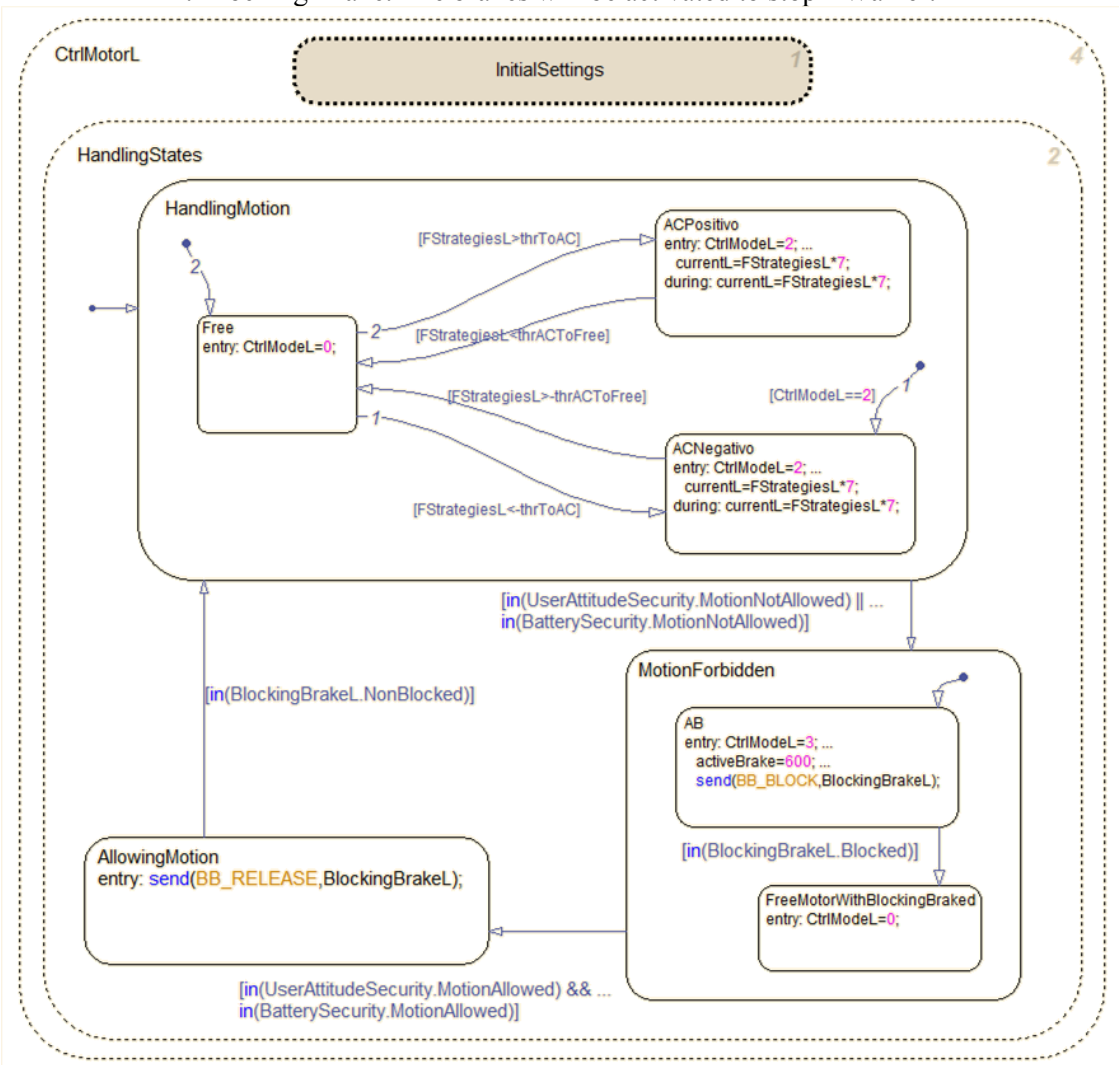

Braking activity example:



***Motor system***

Motor controllers (left motor controller is shown as an example in next figure) will set de motor activity level. Three behaviours can be determined:

- AC: Active Current. The motor helps user's walk.
- AB: Active Brake. Motors are used to slow user's walking speed.
- BB: Blocking Brake. The brakes will be activated to stop i-Walker.




## 3.7.6. Interface for customisation of the configuration

All i-Walker parameters and collected data are mapped in a *modbus* dataspace, as shown in Table 21:

**Table 21.** i-Walker parameter list.

| @ModBus hexa | @ModBus deci | Data type | R/W | Magnitude | Units | Range |
|---|---|---|---|---|---|---|
| A | 10 | Int16-U | R/W | RecordingExperiment | | [0,1] |
| B | 11 | Int16-U | R/W | Lambda Left | | |
| C | 12 | Int16-U | R/W | Lambda Right | | |
| D | 13 | Int16-U | R/W | Nu Left | | |
| E | 14 | Int16-U | R/W | Nu Right | | |
| 14 | 20 | Int16-S | R | Left Hand Force X | cN | [-20000...20000] |
| 15 | 21 | Int16-S | R | Left Hand Force Y | cN | [-20000...20000] |
| 16 | 22 | Int16-S | R | Left Hand Force Z | cN | [-20000...20000] |
| 17 | 23 | Int16-S | R | Right Hand Force X | cN | [-20000...20000] |
| 18 | 24 | Int16-S | R | Right Hand Force Y | cN | [-20000...20000] |
| 19 | 25 | Int16-S | R | Right Hand Force Z | cN | [-20000...20000] |
| 1A | 26 | Int16-S | R | Left Normal Force | cN | [-20000...20000] |
| 1B | 27 | Int16-S | R | Right Normal Force | cN | [-20000...20000] |
| 1E | 30 | Int16-U | R | Tilt | | |
| 1F | 31 | Int16-U | R | Roll | | |
| 20 | 32 | Int16-U | R | Handlebar LEDs Left | | 2 bits tipo blinking + 6 bits RGB |
| 21 | 33 | Int16-U | R | Handlebar LEDs Right | | 2 bits tipo blinking + 6 bits RGB |
| 22 | 34 | Int16-U | R | Handlebar Vibrator Left | | [0,1,2] |
| 23 | 35 | Int16-U | R | Handlebar Vibrator Right | | [0,1,2] |
| 24 | 36 | Int16-U | R | Hand Brake Left | | 0 Free, 1 Braking, 2 Braked |
| 25 | 37 | Int16-U | R | Hand Brake Right | | 0 Free, 1 Braking, 2 Braked |
| 26 | 38 | Int16-U | R | Wheel Blocking Status Left | | |
| 27 | 39 | Int16-U | R | Wheel Blocking Status Right | | |
| 28 | 40 | Int16-U | R | Current consumption | mA | [0...10000] |
| 29 | 41 | Int16-U | R | Battery Voltage | mV | [0...24000] |
| 2A | 42 | Int16-U | R | Motor ControlModeSetpoints | | Consigna independientes para Left&Right: 0f, 1as, 2ac, 3ab |
| 2B | 43 | Int16-S | R | External Request for LeftWheel Velocity | mm/s | -1000...1000 |
| 2C | 44 | Int16-S | R | External Request for RightWheel Velocity | mm/s | -1000...1000 |
| 2D | 45 | Int16-S | R | External Request for active current left | mA | -3000...3000 |
| 2E | 46 | Int16-S | R | External Request for active current right | mA | -3000...3000 |
| 32 | 50 | Int16-U | R/W | TestMode | | [0,1] |
| 33 | 51 | Int16-U | R/W | TimeOutMax | ms | |
| 34 | 52 | Int16-U | R/W | Motor ControlModeSetpoints | | Consigna independientes para Left&Right: 0f, 1as, 2ac, 3ab |
| 35 | 53 | Int 16-s | R/W | External Request for LeftWheel Velocity | mm/s | -1000...1000 |
| 36 | 54 | Int 16-s | R/W | External Request for RightWheel Velocity | mm/s | -1000...1000 |
| 37 | 55 | Int 16-s | R/W | External Request for active current left | mA | -3000...3000 |
| 38 | 56 | Int 16-s | R/W | External Request for active current right | mA | -3000...3000 |
| 39 | 57 | Int16-U | R/W | Test Handlebar LEDs Left | | 2 bits tipo blinking + 6 bits RGB |
| 3A | 58 | Int16-U | R/W | Test Handlebar LEDs Right | | 2 bits tipo blinking + 6 bits RGB |
| 3B | 59 | Int16-U | R/W | Test Handlebar Vibrator Left | | [0,1,2] |
| 3C | 60 | Int16-U | R/W | Test Handlebar Vibrator Right | | [0,1,2] |
| 3D | 61 | Int16-U | R/W | Test Wheel Blocking Left | | [0, 1] |
| 3E | 62 | Int16-U | R/W | Test Wheel Blocking Right | | [0, 1] |
| 46 | 70 | Int32-s | R/W | Estimated Pose X High | mm | -2^31...2^31-1 |
| 47 | 71 | Int32-s | R/W | Estimated Pose X Low | mm | -2^31...2^31-1 |
| 48 | 72 | Int32-s | R/W | Estimated Pose Y High | mm | -2^31...2^31-1 |
| 49 | 73 | Int32-s | R/W | Estimated Pose Y Low | mm | -2^31...2^31-1 |
| 4A | 74 | Int16-s | R/W | Estimated Pose Orientation | mrad/s | 0...2^16-1 |
| 4B | 75 | Int16-s | R | Left_SpeedMMpS | mm/s | -1000...1000 |
| 4C | 76 | Int16-s | R | Right_SpeedMMpS | mm/s | -1000...1000 |
| 4D | 77 | Int32-s | R | Left_EncoderHigh | Enc ticks | -2^31...2^31-1 |
| 4E | 78 | Int32-s | R | Left_EncoderLow | Enc ticks | -2^31...2^31-1 |
| 4F | 79 | Int32-s | R | Right_EncoderHigh | Enc ticks | -2^31...2^31-1 |
| 50 | 80 | Int32-s | R | Right_EncoderLow | Enc ticks | -2^31...2^31-1 |
| 5A | 90 | Int16-u | R | CPUUsage | % | 0...100 |

Parameters in range {A..E}, are used to set the helping level i-Walker has to perform. Two parameters, *lambda* and *nu,* are available to the physiotherapists to create a tailored behaviour of i-Walker.

i-Walker Modbus is mapped in Ethernet or WIFI Configurations by the On-top computer.

### 3.7.7. i-Walker on-project integration

Medical experts ask to record a list of i-Walker parameters for further analysis of patient gait, monitor rehabilitation process and study of falls. These 19 parameters are shown in Table 22. They are referring to forces, angles and speed during patient walk.

**Table 22**. i-Walker gait-related parameters.

| | | | | | | |
|---|---|---|---|---|---|---|
| 14 | 20 | Int16-S | R | Left Hand Force X | cN | [-20000...20000] |
| 15 | 21 | Int16-S | R | Left Hand Force Y | cN | [-20000...20000] |
| 16 | 22 | Int16-S | R | Left Hand Force Z | cN | [-20000...20000] |
| 17 | 23 | Int16-S | R | Right Hand Force X | cN | [-20000...20000] |
| 18 | 24 | Int16-S | R | Right Hand Force Y | cN | [-20000...20000] |
| 19 | 25 | Int16-S | R | Right Hand Force Z | cN | [-20000...20000] |
| 1A | 26 | Int16-S | R | Left Normal Force | cN | [-20000...20000] |
| 1B | 27 | Int16-S | R | Right Normal Force | cN | [-20000...20000] |
| 1E | 30 | Int16-U | R | Tilt | | |
| 1F | 31 | Int16-U | R | Roll | | |
| 24 | 36 | Int16-U | R | Hand Brake Left | | 0 Free, 1 Braking, 2 Braked |
| 25 | 37 | Int16-U | R | Hand Brake Right | | 0 Free, 1 Braking, 2 Braked |
| 46 | 70 | Int32-s | R/W | Estimated Pose X High | mm | -2^31...2^31-1 |
| 47 | 71 | Int32-s | R/W | Estimated Pose X Low | mm | -2^31...2^31-1 |
| 48 | 72 | Int32-s | R/W | Estimated Pose Y High | mm | -2^31...2^31-1 |
| 49 | 73 | Int32-s | R/W | Estimated Pose Y Low | mm | -2^31...2^31-1 |
| 4A | 74 | Int16-s | R/W | Estimated Pose Orientation | mrad/s | 0...2^16-1 |
| 4B | 75 | Int16-s | R | Left_SpeedMMpS | mm/s | -1000...1000 |
| 4C | 76 | Int16-s | R | Right_SpeedMMpS | mm/s | -1000...1000 |

These data (46 bytes) are packed with a time stamp by the Raspberry on-top computer. Time stamp will help in the analysis of data, allowing dynamic interpretation of them. Table 23 depicts the data structure use in the i-Walker communications.

**Table 23**. i-Walker communications data structure.

| Number (4 bytes) | i-Walker Data (46 bytes) | Time Stamp ( 4 bytes) | CRC (2 bytes) |
|---|---|---|---|

Every 100ms data is stored in Raspberry PI microSD card. At this rate, one hour of experimentation will fill 2MB of memory, and, if a Wireless connection is used, 10Kbps of bandwidth will be necessary. Figure 44 shows how i-Walker integrates to project platform by recording data on a microSD card or directly sharing it on a Wireless Network.

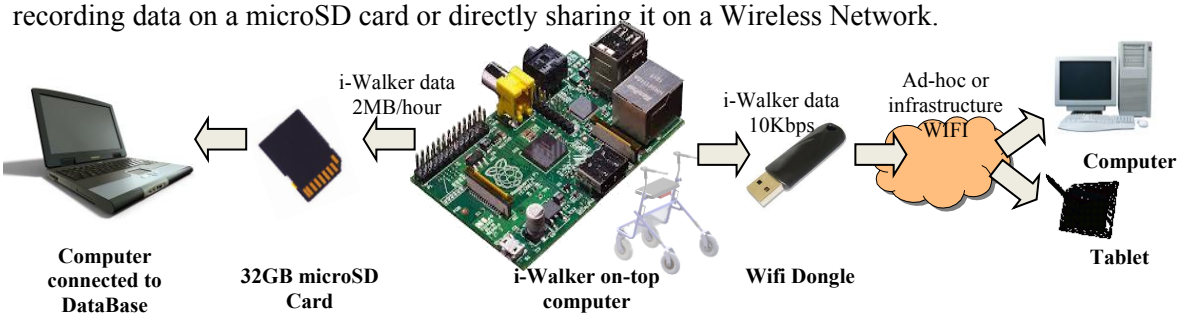



**Figure 44.** i-Walker integration and communications.

## 3.8. Software components

### 3.8.1. Software for the central computer

The software installed on the central computer is responsible for managing the entire ZigBee network and bed sensor and transmit the messages received to the mobile phone. It also controls the access to the mobile phone to inform the fall detector, when necessary, that the system is not directly accessible and try it directly to the mobile phone.

The following sections describe the software installed and operation.

#### 3.8.1.1. Installed software and configuration

Installed Software in Central Computer:

- Linux O.S.: Ubuntu Server 12.04
- Java 1.6
- Debian Packages (Bluetooth): bluez, bluez-utils, rfkill, bluetooth, libbluetooth-dev, sudo and all dependencies.
- Fate Application (java).

Configuration:

- Udev Rules for bed presence sensor and XStick Zigbee: fix serial ports.
- Daemon that launches Fate Application at PC starts.

#### 3.8.1.2. Operation of the central computer

When the FATE application is running, the first thing is to check all connections:

- First, check if the phone is paired and available in the system. If not, it makes visible by Bluetooth and try discovery to connect with it and still trying until to have connection.
- Next: constantly check the serial ports to ensure that XStick ZigBee and Bed Presence sensor are connected. If not, it repeats the process periodically.

The application has two processes that are constantly listening ZigBee and USB ports.

Zigbee accepted messages are described in Section 3.6.4. The central computer receives a ZigBee frame with push information and answers with another ZigBee frame with push information response.

As shown in Figure 45, when the central computer receives a ZigBee Frame, transforms the Push Information from ZigBee protocol to Bluetooth protocol (see section 3.6.3) and tries to communicate with the phone to send the message following the next steps:

1. Seek the mobile and sends the message.
2. Wait mobile response.

3. Respond to the ZigBee network depending on the response of the mobile phone.

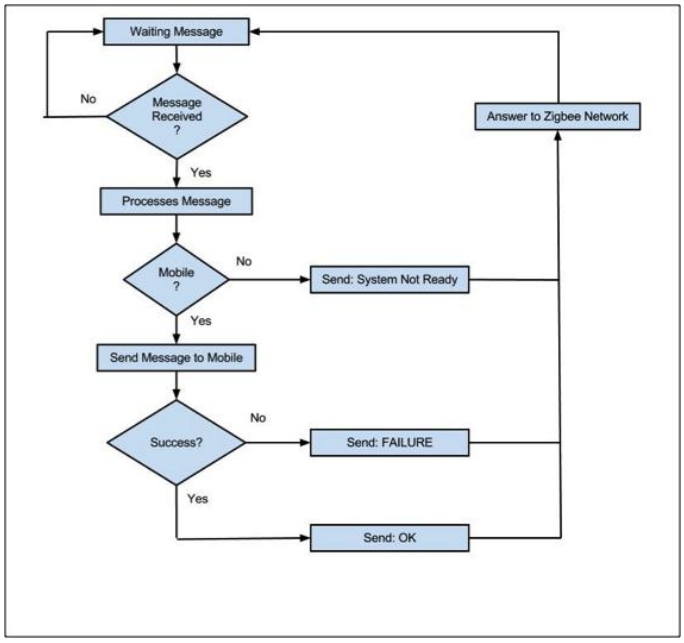



**Figure 45.** Messages management process for the fall detector through the ZigBee network.

Table 24 shows the functionality of the application when it receives a message via the ZigBee network.

**Table 24.** Operation of the central computer when receiving a message from the fall detector.

| **Fall Sensor sends message** | **Central Computer Action** | **Mobile Response** | **Central Computer Response to Fall Sensor** |
|---|---|---|---|
| Any Message | Transforms message and forwards it to the mobile | Unable to connect to the mobile phone or responds with an empty message. | Push Information Response SYSTEM NOT READY |
| | | SUCCESS Message | Push Information Response OK |
| | | Any Message other than SUCCESS | Push Information Response FAILURE |

The bed sensor protocol is described in the section 3.3.4. The FATE system ignores all messages received from bed presence sensor except ABSENCE and PRESENCE that are sent to the mobile phone.

For communication with the mobile phone, the central computer uses the same Bluetooth protocol described in section 3.6.3 by adding the messages of Table 25 to the content ID field.

**Table 25.** Content ID added to Bluetooth messages triggered by the bed sensor.

| Content ID | Content Name | Content Description | Format |
|---|---|---|---|
| 0x10 | Bed Presence | User is in bed | Unsigned 8-bit integer |
| 0x11 | Bed Absence | User have left the bed | Unsigned 8-bit integer |

As shown in Figure 46, the management process repeats the next steps:

1. Receive ABSENCE or PRESENCE message.
2. Create a message Bluetooth adding the corresponding Content ID to the Push Information.
3. Connect with mobile phone and sends the message.

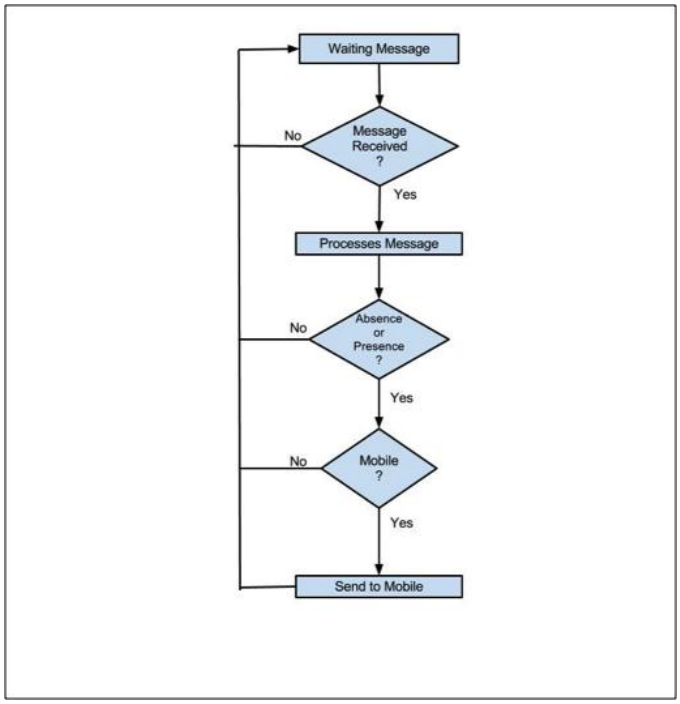



**Figure 46.** Message management process for the bed presence sensor by USB.

### 3.8.2. Software for the mobile phone

The main aim of the mobile application is to listen all events generated by both the fall detector and the bed presence sensor, notify the user to take action and send warnings to people or companies that need to manage a set of alert messages.

The software is contained in an android file (*.Apk) that is installed on the mobile. It supports Android versions 2.3.4 or higher. It uses Bluetooth communication from mobile phone and

GSM communication to make voice calls and send SMS. In some cases, it can use GPRS communication to send XML messages.

The application is set to be always active since the start of the mobile operating system.

The android application is structured in four activities:

- **Main Activity**: controls the application and execute the main tasks: fall sensor and central computer messages processing, notify user and send alerts.
- **Bluetooth Control Activity**: ensures that Bluetooth is always active on mobile.
- **Airplane Mode Control Activity**: it is detecting when the airplane mode is active, so as to indicate to the fall detector that the system is unavailable.
- **GPS Control Activity:** is responsible for activating the GPS when the user leaves home.

Figure 47 summarises the most important processes of the application, which are described in next sections.

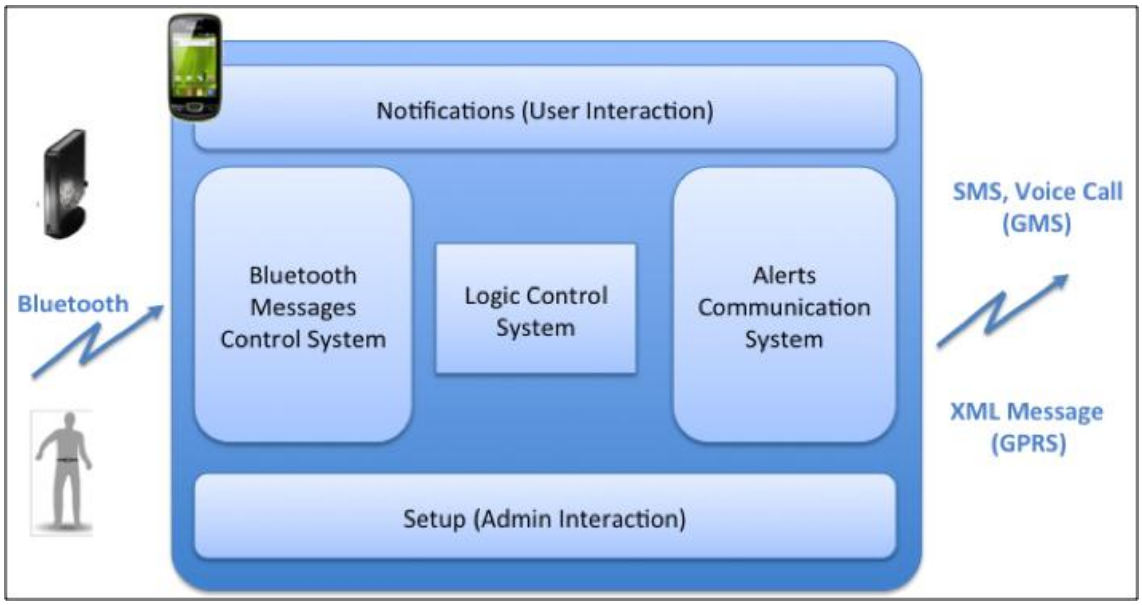



**Figure 47.** Functional scheme of the mobile application.

### 3.8.2.1. Bluetooth Messages Control System (BMCS)

The mobile application has a service system that is constantly listening Bluetooth connection attempts. These connections come from both the central computer (bed and fall sensor) or fall detector.

Through this connection, the system manages the various incoming messages to set different actions by managing timers.

Table 26 details the messages that it receives and how it manages the application.

**Table 26.** Operation of FATE mobile application when receiving Bluetooth messages.

| Input Messages | Action |
|---|---|
| VIGILANCE CONTROL MESSAGE | Reset Timer: Sensor Not Detected.<br>Start Timer: User Notification<br>**Description**: resets the timer. If the set value is exceeded, the NO SENSOR alert process starts<br>Previously, when the Notification timer exceeds the set value notifies the user via mobile. |
| FALL DOWN | Start Timer: Fall Down Alarm.<br>Start Timer: User Notification<br>Description: If the timer exceeds the set value, the DETECTION BY FALL SENSOR alert process starts.<br>Previously, when the Notification timer exceeds the set value notifies the user via mobile. |
| FALL DOWN AUTOMATIC RECOVERY | Stop Timer: Fall Down Alarm.<br>**Description**: Stop timer. If you have not reached the set value, the FALL DETECTION BY SENSOR alert process does not start.<br>Automatically starts the process FALL RECOVERY. |
| FALL DOWN MANUAL RECOVERY | Stop Timer: Fall Down Alarm.<br>**Description**: Stop timer. If you have not reached the set value, the FALL DETECTION BY SENSOR alert process does not start.<br>Automatically starts the process FALL RECOVERY. |
| PANIC ACTIVATED | Start Timer: Panic Alarm.<br>Start Timer: User Notification<br>**Description**: If the timer exceeds the set value, the USER PANIC alert process begins.<br>**Previously, when the Notification timer exceeds the set value notifies the user via mobile.** |
| BED ABSENCE | If it is night time (by setup):<br>Start Timer: Bed Absence Alarm<br>Start Timer: User Notification<br>**Description:.** If the timer exceeds the set value, the TO MUCH TIME OUT OF BED alert process starts.<br>**Previously, when the Notification timer exceeds the set value notifies the user via mobile.** |
| BED PRESENCE | Stop Timer: Bed Absence Alarm<br>**Description**: Stop timer. If you have not reached the set value, the TO MUCH TIME OUT OF BED alert process does not start. |
| BATTERY LOW | Notifies the user via mobile. |
| BATT. CHARGING | Removes the notification to the user of low battery. |

### 3.8.2.2. Logic Control System

This system is based on a set of variables and timers. The timers are activated or stopped by events received by the BMCS. When these timers reach the value determined by setup, they create user notifications and/or alerts by the Alert Control System (ACS).

The timers of the application are:

- **Sensor not Detected**: When it times out without receiving sensor messages warns ACS to communicate an alert.
- **User Notification**: When it times out, notifies the user by screen and sound. The time is set according to the type of message received by the BMCS.
- **Fall Down Alarm**: When it times out warns ACS to communicate an alert.
- **Panic Alarm**: When it times out warns ACS to communicate an alert.
- **Bed Absence Alarm**: When it times out warns ACS to communicate an alert.

### 3.8.2.3. Alert Communication System (ACS)

The alert communication system, generates 6 types of messages, as indicated in Table 27.

**Table 27.** Alert messages of the FATE system.

| Type Messages (alerts) | Description (when sent a message) |
|---|---|
| **FALL DETECTION** | When fall sensor detect a fall and it spend X minutes without cancelation on the mobile or receive fall recovery by sensor. |
| **FALL RECOVERY** | When receive a fall recovery after sent a message of FALL DETECTION BY SENSOR.<br>Informational message. No Alert. |
| **USER PANIC** | When user push the panic button on the sensor. |
| **NO SENSOR/LOW BATT** | When mobile and PC don't receive signals from fall sensor in X minutes, but the last messages received indicate low battery of the fall sensor.<br>Allow adequate time to charge the sensor before deciding to send the message.<br>It is possible, but unlikely, that the user has been fallen. |
| **NO SENSOR (FALL?)** | When mobile and PC don't receive signals from fall sensor in X minutes, but the last messages received indicate correct or high battery of the fall sensor.<br>Allow adequate time to charge the sensor before deciding to send the message. |
| **OUT OF BED (FALL?)** | When the user leaves the bed and no signal fall sensor (probably loading) for X period.<br>This alert occurs only during night time set by configuration. |

Messages are sent by different methods:

- **SMS**: An SMS is sent to the indicated number and type of message, if configured. The SMS format is explained in Table 28 and some examples are given in Table 29.

- **Voice Call**: It generates an automated call to indicated number and type of message, if configured. It checks that the user does not hang up until the called user takes the call.

- **XML Message (Web service)**: It generates an XML message that is sent by HTTP REST method if configured. This message follows the XML Schema (XSD) included in Table 30.

**Table 28.** SMS format for alert messages in the FATE system.

| LINE | FIELD | DESCRIPTION | VALUES |
|---|---|---|---|
| 1 | SERVICE | Used to differentiate from other SMS messages. It's a fixed text named "FATE" | FATE |
| 2 | USER | the phone number of the mobile phone.(setup on FATE Android app).<br>Max 15 characters.<br>Note: It's not possible access to the information of phone number on the mobile. | Examples:<br>+34607014479<br>+35316030200<br>+39071501031<br>607014479 |
| 3 | EVENT TYPE | differentiate the type of message. See the list of Type Messages.<br>It will be translated to local language. | values: FALL DETECTION, FALL RECOVERY, USER PANIC, NO SENSOR/LOW BATT, NO SENSOR (FALL?), OUT OF BED (FALL?) |
| 4 | EVENT TIME | Date and Time of Event (fall, recovery, panic, ...) | format: YYYY-MM-DD HH24:MI:SS(+XX)<br>Note: +XX is over GMT |
| 5 | EVENT LOCATION | Indicates if user is at home or outside. ("HOME", "OUTDOOR").<br>It will be translated to local language.<br>Note: "OUT" usually means outside home, but it really means that sensor don't have access to PC at Home (no power?) and connect directly to the mobile. | values: HOME, OUTDOOR |
| 6 | DEPARTURE TIME FROM HOME | Time difference between EVENT TIME and DEPARTURE FROM HOME (negative: before event).<br>Shows "none" if event location is "HOME" (user at home) | format:+HH24:MI:SS or "none" |
| 7 | GPS LOCATION | Location Obtained by mobile GPS.<br>Shows "none" if event location is "HOME" or if it's not possible to know the location on mobile. | |
| | LAST TIME LOCATION | Time difference between FALL TIME and LOCATION indicated on message (negative: before event). | format:<br>GPS:+HH24:MI:SS,+99,99999,+99,99999, XXXXX or "GPS:none"<br>4 values separated by commas:<br>- last time location: +HH24:MI:SS<br>- Latitude (decimals depends of mobile GPS)<br>-Longitude (decimals depends of mobile GPS)<br>- Margin or error: 0 to 999, ">1km" |
| | LATITUDE | latitude coordinate in decimal geographic format of location | |
| | LONGITUDE | longitude coordinate in decimal geographic format of location. | |
| | MARGIN OF ERROR | max meter error from indicated location.<br>Shows ">1km" if error meter is more than 1000 meters. | |
| 8 | GSM LOCATION | Location Obtained by Triangulation GSM.<br>Shows "none" if event location is "HOME" or if it's not possible to know the location on mobile. | |
| | LAST TIME LOCATION | Time difference between FALL TIME and LOCATION indicated on message (negative: before event). | format:<br>GSM:+HH24:MI:SS,+99,99999,+99,99999 ,XXXXX or "GSM:none"<br>4 values separated by commas:<br>- last time location: +HH24:MI:SS<br>- Latitude (decimals depends of mobile GPS)<br>-Longitude (decimals depends of mobile GPS)<br>- Margin or error: 0 to 999, ">1km" |
| | LATITUDE | latitude coordinate in decimal geographic format of location | |
| | LONGITUDE | longitude coordinate in decimal geographic format of location. | |
| | MARGIN OF ERROR | max meter error from indicated location.<br>Shows ">1km" if error meter is more than 1000 meters. | |

**Table 29.** Examples of SMS messages in the FATE system.

| CASE | DESCRIPTION | SMS MESSAGE |
|---|---|---|
| **FALL AT HOME** | | FATE<br>607014479<br>FALL DETECTION<br>HOME<br>2012-11-07 10:24:22(+01)<br>none<br>GPS:none<br>GSM:none |
| **FALL OUTSIDE HOME** | User fall 20 minutes after leave home.<br>Better location on mobile was 35 seconds since fall. | FATE<br>607014479<br>FALL DETECTION<br>OUTDOOR<br>2012-11-07 10:24:22 (+01)<br>-00:20:00<br>GPS:-00:00:35,+40,5280,-3,6493,10<br>GSM:-00:00:38,+40,5278,-3,6495,500 |
| **FALL OUTSIDE HOME WITHOUT LOCATION** | User fall 20 minutes after leave home.<br>Mobile couldn't find a location from leaving home until the message is sent. | FATE<br>607014479<br>FALL DETECTION<br>OUTDOOR<br>2012-11-07 10:24:22 (+01)<br>-00:20:00<br>GPS:none<br>GSM:none |
| **RECOVERY AT HOME** | 4 minutes and 26 seconds from fall at home (previous message), the sensor indicates a recovery. | FATE<br>607014479<br>FALL RECOVERY<br>HOME<br>2012-11-07 10:28:48(+01)<br>none<br>GPS:none<br>GSM:none |
| **RECOVERY OUTSIDE HOME** | 4 minutes and 26 seconds from fall outdoor (previous message), the sensor indicates a recovery.<br>Better location on mobile when recovery was 2 seconds before event. | FATE<br>607014479<br>FALL RECOVERY<br>OUTDOOR<br>2012-11-07 10:28:48(+01)<br>-00:24:26<br>GPS:-00:00:02,+40,52640,-3,64998,10<br>GSM:-00:00:45,+40,52635,-3,64999,>1km |

**Table 30.** XSD of XML messages in the FATE system.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:element name="missatge">
              <xsd:complexType >
                        <xsd:sequence>
                              <xsd:element name="basicData"   type="basicData"/>
                              <xsd:element name="gpsPosition"   type="position"/>
                              <xsd:element name="smsLocalization"   type="position"/>
                         </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
                    <xsd:complexType name="basicData" mixed="true">
                      <xsd:sequence>
                                        <xsd:element name="service"      type="xsd:string"/>
          <xsd:element name="user"      type="xsd:string"/>
          <xsd:element name="eventType"        type="eventType"/>
          <xsd:element name="eventLocation"                type="eventLocation"/>
          <xsd:element name="eventTime"   type="xsd:string"/>
          <xsd:element name="departureTimeFromHome"   type="xsd:string"/>
                      </xsd:sequence>
                    </xsd:complexType>
                    <xsd:complexType name="position" mixed="true">
                      <xsd:sequence>
                                        <xsd:element name="lastTimeLocation"   type="xsd:string"/>
          <xsd:element name="latitude"   type="xsd:string"/>
          <xsd:element name="longitude"   type="xsd:string"/>
          <xsd:element name="marginOfError"   type="marginOfError"/>
                      </xsd:sequence>
                    </xsd:complexType>
          <xsd:simpleType name="eventType">
               <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="FALL DETECTION BY SENSOR"/>
                  <xsd:enumeration value="FALL RECOVERY"/>
                  <xsd:enumeration value="USER PANIC"/>
                  <xsd:enumeration value="NO SENSOR &amp;amp; LOW BATTERY"/>
                  <xsd:enumeration value="TOO MUCH TIME OUT OF BED"/>
               </xsd:restriction>
          </xsd:simpleType>
          <xsd:simpleType name="eventLocation">
               <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="HOME"/>
                  <xsd:enumeration value="OUTDOOR"/>
               </xsd:restriction>
          </xsd:simpleType>
          <xsd:simpleType name="marginOfError">
               <xsd:restriction base="xsd:integer">
                  <xsd:minInclusive value="0"/>
                  <xsd:maxInclusive value="999"/>
               </xsd:restriction>
          </xsd:simpleType>
</xsd:schema>
```

### 3.8.2.4. Notifications and interactions with the user

The application has no menu options to the user, only to an administrator (the person in charge of the system installation). An attempt was made to simplify their interaction showing only essential messages that require user action.

The FATE application icon is always shown in the status bar of the phone to confirm that it is working (see Figure 48).


**Figure 48.** Status bar of the mobile pone with the FATE icon.

The user only receives notifications when there are messages that require intervention as can be seen in Figure 49.

Notifications generate a message and audible alarm that appear when the the User Notification timer times out, showing the message and available actions.

All notifications have the following structure:

- **Message**: Shows the corresponding text message to the alert. There are six possible alert messages:
  - Fall Detected
  - No sensor with high battery
  - No sensor with low battery
  - Panic Button
  - To much time out of bed
  - Low Battery

- **Remaining Time**: Shows the time remaining to send alert message to the contacts configured.

- **Mute Button**: Allows the user to silence the alarm and generates the alert when the timer set ends. Not displayed to the Low Battery message.

- **Cancel Call Button**: Allows the user to cancel the alarm and avoid communicating alarm to contacts. Not displayed in the Low Battery message.

- **Close**: Closes the message. Only appears for low battery. All other messages will automatically close the message when timer times out.


**Figure 49.** Screenshots of notifications to the user in the application.

### 3.8.2.5. Setup

To access configuration on the mobile phone, a password is required that is known only by the person in charge of installing the system.

When the application is executed, it asks for password and accesses to a menu with the following options:

- **CONTACTS**: Sets the list of telephone numbers used for calling or sending SMS. They are used in the notification method setting.
- **MESSAGES:** Permits to specify the notification method for each type of message. Several options can be set.
- **PARAMETERS**: Sets parameters that affect the user notifications and arrangements for sending alert messages.

Figure 50 shows several screenshots of the configuration menu.

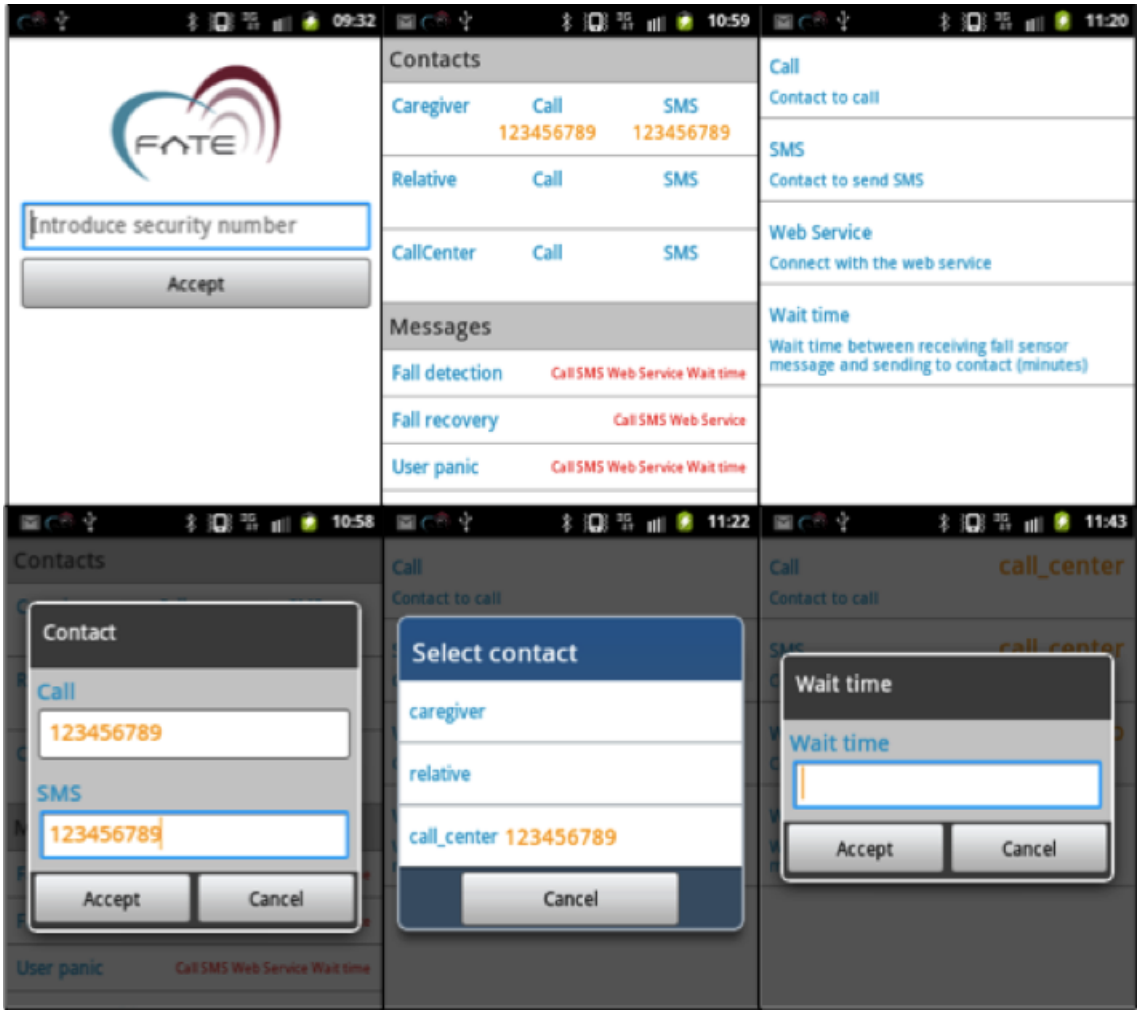



**Figure 50.** Screenshots of the configuration menu of the FATE application.

The contacts section permits to specify the phone number of three predefined profiles for both SMS or calls. It is also possible to setup the URL to send XML messages. The diferent profiles are presented in Table 31.

**Table 31.** Contact settings used in the FATE application.

| Profiles | Call | SMS | WebService |
|---|---|---|---|
| **Caregiver** | number | number | N/A |
| **Relative** | number | number | N/A |
| **Callcenter** | number | number | url |

The messages section permits to set up what methods will be used to contact with every type of message. An example is shown in Table 32.

**Table 32.** Settings for sending message by type of message in the FATE application.

| Type Messages (alerts) | Call | SMS | WebService | Wait Time |
|---|---|---|---|---|
| **FALL DETECTION** | Callcenter | caregiver | Yes | <minutes> |
| **FALL RECOVERY** | Callcenter | none | Yes | N/A |
| **USER PANIC** | none | relative | none | <minutes> |
| **NO SENSOR/LOW BATT** | none | caregiver | none | <minutes> |
| **NO SENSOR (FALL?)** | Callcenter | caregiver | Yes | <minutes> |
| **OUT OF BED (FALL?)** | Callcenter | caregiver | Yes | <minutes> |

Wait Time means the time between receiving a message from any sensor (fall or bed presence) and sending an alert to the specified contact.

The parameters section permits to set up other parameters necessary to use the service. These parameters are summarised in Table 33.

**Table 33.** Parameter settings for the FATE mobile application.

| Parameter | Values |
|---|---|
| **Message Language** | select: english, italian, spanish or catalan |
| **mobile phone number** | 999999999 |
| **Type of Call Center** | select: COOSS, TER or SEM |
| **url WebService (WS)** | url to send message (if exist) |
| **Time between WS and Call/SMS (seconds)** | 9999 |
| **Time between SMS and Call (seconds)** | 9999 |
| **Wait Time Before Notify on mobile (minutes)** | 9999 |
| **Night Time Start** | HH:MM |
| **Night Time Duration (hours and minutes)** | HH:MM |

# 4. FATE system use cases

This section describes how the FATE system should be operated by the end user once the complete system has been configured and installed. First of all the user interaction with the fall detector will be explained, and then its operational instructions will be provided.

## 4.1. Interaction with the fall detector

Figure 51 shows the four main sections of the fall detector: the multicolour LED, the action button, the reset button and the charging connector.

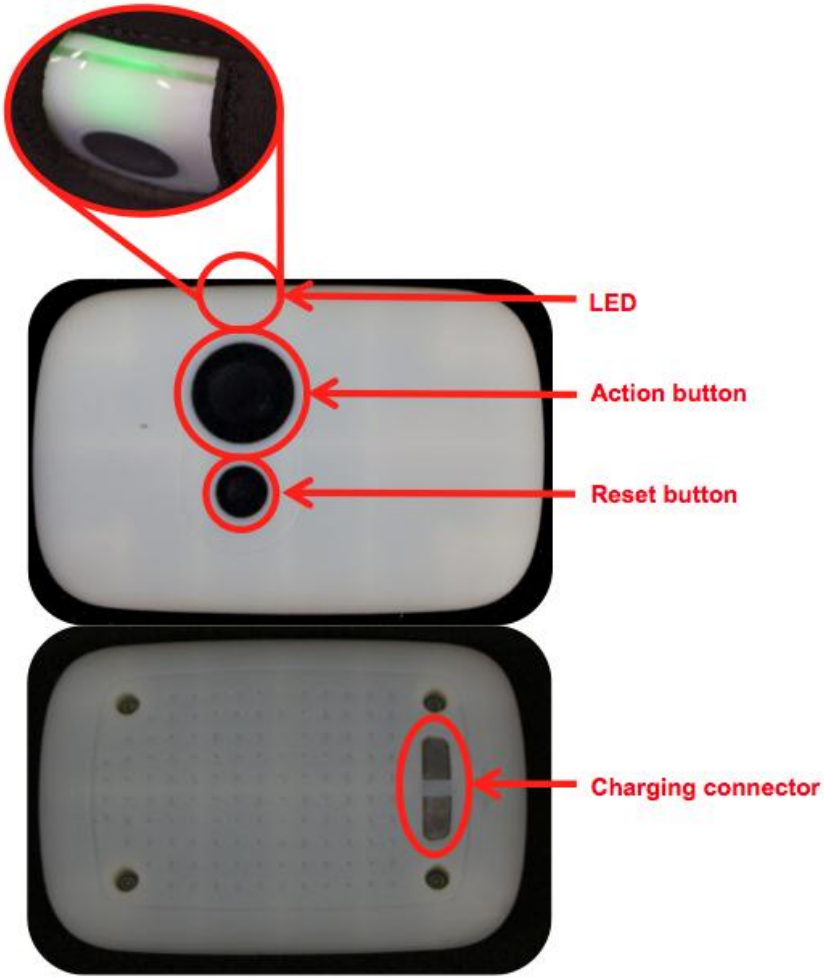



**Figure 51.** Main components of the fall detector.

The behaviour of the sensor is determined by two states: Off and On. These states are mutually exclusive; a third state, related to the supervision of the battery is compatible with the other two and can be treated as an independent state. Figure 52 shows the relationship between these states.

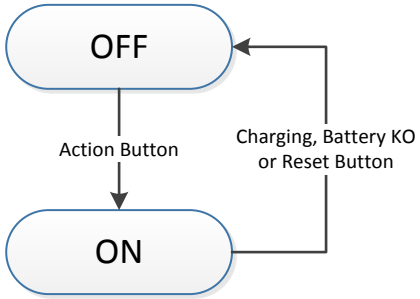



**Figure 52.** System states and its relationship.

In the OFF state the fall detector can not work. The sensor is located in this state during or after a charge, when it detects a critical battery level or after pressing the Reset button.

In the ON state, the fall detector initializes the microcontroller. When the sensor has already initialized all internal devices it is ready to begin the discovery process. Once ready, the inertial sensor reads the signals, processes them and communicates with the rest of the system.

When the detector enters the ON state has a courtesy period of 5 minutes in which the sensor does not process any signal or send any message or alarm. During this courtesy period the user must place the detector in its operative position.

Sensor status is indicated by a single intermittent multicolour LED. Different states may be indicated by the diversity of LED colour codes. For example, a green/red alternately flashing means that the detector is working but the charge level of the battery is low and needs recharging the device. Second, a light green/blue flashing means the detector is sending data. If the detector is charging the battery, this is indicated by a solid blue light. If not, a steady green light notifies that the battery is fully charged. Generally speaking if the LED is red this means that user intervention is needed or that it has detected an emergency state. Additionally the fall detector has a buzzer to inform the user of some situations where some action is required or where an alarm is sent to the call center. The colour coding of the LED and the buzzer are described in Figure 53.

Every night, when the sensor is removed to go to bed, it has to be charged. To charge the sensor the provided charger has to be connected to a plug and then it has to be connected to the fall detector. Figure 54 illustrates the charging sequence for the fall detector.

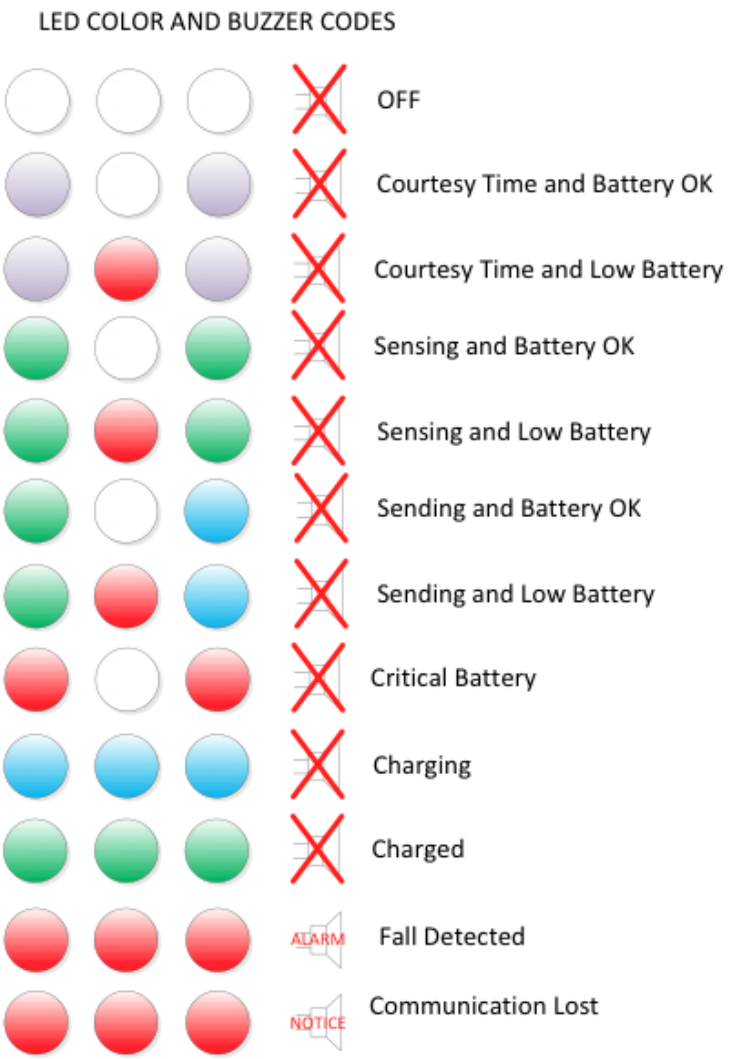



**Figure 53.** LED colour and buzzer codes.

**Figure 54.** Charging sequence for the fall detector.

## 4.2. Operational instructions

The fall detector is turned on by pressing the action button during 5. At that time the LED starts flashing. The sensor is turned off only when the system detects a critical battery level, when the user puts the system to charge or when the reset button is pressed. Then, if the fall detector shuts down due to a critical battery level, the device must be connected to the charger. Once the battery is fully charged, the sensor can be turned on using the action button.

It is worth noting that, for security reasons and in order to avoid an accidental shut down of the system, the reset button has been mechanically designed so that it can only be activated by

exerting a considerable force using a pricking object. This also avoids a possible confusion by the user between the two buttons, panic and reset, present in the device. Only the panic button can be activated by the user by pressing on it with the fingers.

Once the fall detector is running in normal operation and its battery charged it will wait a courtesy time of 5 minutes before sending any alarm. This permits the user to buckle the belt without raising any alarm. During this time the LED flashes magenta every second. When the courtesy time expires the system starts working in normal operation, activating alarms and making the LED to flash in green colour.

If the fall detector detects a fall a sound similar to an emergency siren will be producing, alerting the user that it is going to send an alarm to the emergency center. If it really is an emergency (i.e., there was an actual fall), DON'T PRESS THE ACTION BUTTON because this will cancel the alarm. If the user mistakenly cancelled the alarm this can be generated again with the action button as described in this section.

If the detector produces the alarm sound and the user is not in an emergency situation he/she can cancel sending alarm by pressing the action button. Also depending on the configuration of the alarm system a call may be received to confirm the user's status.

If the user is in an emergency situation and the detector has not detected a fall, it is possible to press the action button in order to generate an alarm and ask for assistance.

Although the fall detector is waterproof DO NOT SHOWER OR TAKE A BATH WHILE WEARING IT.

The fall detector should only be used when the user is out of bed, NEVER WHILE SLEEPING. A NIGHT YOU IT MUST BE LEFT CHARGING, EVEN IF THE DETECTOR DOES NOT INDICATE A LOW BATTERY STATUS.

From time to time the user must check if the LED indicates that the fall detector is in OFF or low battery status. If the LED indicates low battery the detector should be charged immediately. If the sensor is in OFF state it should be turned on immediately.

If the fall detector is not in low battery status, it checks every 5 minutes if it can communicate with the rest of the FATE system. If during 25 minutes no communication with the rest of the system is detected an intermittent audible warning sounds to indicate to the user that a communication problem exists. The most common scenario for this type of event is that the user leaves home with the detector dressed but left the mobile phone at home. The sound stops by pressing the action button. Depending on the configuration of the alarm system set an alarm to the emergency system may be sent to notify this situation.

**VERY IMPORTANT:**

IF THE USER IS IN AN EMERGENCY THE ACTION BUTTON SHOULD NOT BE PRESSED AND WHENEVER POSSIBLE THE FALL DETECTOR SHOULD NOT BE PINCHED BETWEEN THE GROUND AND THE BODY.

IF THE USER IS IN AN EMERGENCY SITUATION AND THE FALL DETECTOR DOS NOT PRODUCE THE ALARM SOUND THE ACTION BUTTON SHOULD BE PRESSED, IF POSSIBLE.

DO NOT WEAR THE FALL DETECTOR WILL TAKING A BATH OR SHOWER.

DO NOT GO TO BED WITH THE FALL DETECTOR.

REMEMBER TO CHARGE THE FALL DETECTOR EVERY NIGHT.

## 4.3. Description of the FATE use cases

The FATE system is to be installed at user's home by specialised technical personnel. They will take care of installing, testing and configuring the different system components so that they work as expected. The configuration will also permit to adapt the system to the specific requirements of the emergency services available.

Once the system is installed the user has to interact only with the fall detector and the mobile phone. A typical use case for the system at home (considering a complete 24 h. cycle) would be the following:

1. The user wakes up in the morning. The fall detector and the mobile phone are in the bedroom and were connected to their respective chargers last night before going to bed.
2. The user removes the fall detector from the charger and activates it by pressing the panic button. The user will see that the fall detector is switched on because the LED is blinking with a magenta colour, indicating that the courtesy period has started. Then the user places the fall detector inside the neoprene belt and then the belt is placed on the user waist. The user can wait until the LED starts blinking in green colour in order to confirm that the fall detector is working properly. If this does not happen within an interval of around five minutes the user should call the technical staff, since it means that the fall detector is not working properly.
3. The mobile phone should be left connected to its charger in the bedroom. Actually, the user only needs to disconnect the mobile phone from its charger and carry it when leaving home.
4. The user may start normal life activities. If the user is going to take a bath or a shower the belt with the fall detector should be taken off. If a fall is detected while dressing off the belt a sound alarm will be generated by the fall detector. The user can stop the alarm by pressing the panic button.
5. If a fall is detected while the user is at home the fall detector will generate an alarm sound. If it is an actual fall the user should not do anything. After a specified time interval an alarm message (call, SMS, XML file or a combination of them) will be sent to the emergency/telecare service, and the corresponding emergency/care protocol will be triggered. If a fall is detected but the user is not in an emergency situation the user can cancel the alarm message by pressing the panic button or by pressing a cancel button that will appear on the display of the mobile phone.
6. If for any reason the user feels in an emergency situation (even if a fall didn't occur), he/she may press the panic button and an alarm will be sent to the emergency/telecare service.
7. If the user leaves home he/she needs to carry on the mobile phone. If the user leaves home without the mobile phone the fall detector will signal this situation by means of an alert sound (different from the alarm sound). If required by the specific emergency/telecare protocol, the mobile phone may send also an alert message.

8. When the user is outdoors the situation is the same as indicated in stages 5 and 6, but in this case the alarm messages sent to the emergency/care service contain information about the localisation of the user when the alarm was triggered.
9. Even if the battery of the fall detector should last for at least 50 hours, the user may check periodically its status by inspecting the LED. If the LED is blinking in red colour then this means that the battery is low and that the fall detector should be connected to its charger. If required by the emergency/telecare service, an alert message indicating this situation can also be sent.
10. Before going to bed the mobile phone and the fall detector should be connected to their respective chargers.
11. The user can now go to bed. If the user leaves bed and does not come back for a given amount of time a sound alarm will be generated by the mobile phone, and an alert message will appear on its screen. If the user is not in an emergency situation he/she can cancel the alarm by pressing a button displayed on the screen of the mobile phone. If this button is not pressed an alarm will be sent to the emergency/care service.

The use cases of the FATE system at nursing homes are the same as those explained for the users living at home. The only difference is that there is no need for a mobile phone. This is due to the fact that the user is moving in a controlled environment where an RFID-based localisation system knows at any time the current position of the user within the building. The alarm messages are received at the central supervision node of the nursing home, where the corresponding assistance protocol will be triggered.