D4.3 Prototypes of wireless sensor networks

Detailed Architecture Design and Specification of the Wireless Sensor Network Nodes

EUROPEAN COMMISSION

DG Research and Innovation

Seventh Framework Programme

Theme [EeB.ENV.2010.3.2.4-1]

[Compatible solutions for improving the energy efficiency of historic buildings in urban areas]

Collaborative Project – GRANT AGREEMENT No. 260162

Technical References

Project Acronym	3ENCULT
Project Title	Efficient ENergy for EU Cultural Heritage
Project Coordinator	Alexandra Troi EURAC research, Viale Druso 1, 39100 Bolzano/Italy Alexandra.troi@eurac.edu
Project Duration	1 October 2010 – 31 March 2014 (42 Months)

Deliverable No.	D4.3
Dissemination Level	PU
Work Package	WP 4 “Monitoring and control”
Lead beneficiary	13 “UNIBO”
Contributing beneficiary(ies)	6 “USTUTT”
Author(s)	Giacomo Paci
Co-author(s)	Davide Brunelli
Date	28 February 2014
File Name	WP4_D4_3_20140228_P13_Prototypes of wireless sensor networks.doc

Table of Content

List of figures

0 Abstract

The document will treat the design, implementation and deployment of a Wireless Sensor Network targeting the energy efficiency and preservation of historical building.
In Europe there are 55 million dwellings dating before 1945 with more than 120 million Europeans living there. 150 towns and urban fragments are declared World Cultural Heritage site. Those buildings and sites usually have old and inefficient HVAC system that damages cultural heritage goods and provides users discomfort with high energy waste. Than a refurbishment should be performed in a less invasive way, preserving the heritage values but even providing the users comfort with the higher affordable energy efficiency.
With the above consideration the Wireless Sensor Network (WSN) can be a key tool to design and implement the cultural heritage building refurbishment. In fact the easy installation and pervasive deployment of the WSN, permit an extensive building analysis before the intervention. Moreover the reduced impact in term of cost, cable expense, installation masonry and maintenance, underline the capability of the WSN to realize a smart building control system with minimum invasive building renovation.
In the last decade significant progress has been made in the development of wireless sensor networks for different application scenarios. Last generation WSNs are usually made with low-cost electronic microsystem that incorporates both sensors and communication functions. Each microsystem is a node of the network that monitoring its surrounding environment, stores and process the data acquired and spread the information through the wireless link. Furthermore, they can include actuators which are able to receive commands issued by a central controller and act accordingly.
Several wireless systems addressing building automation with lighting and HVAC control are present in the market.
Philips launched a lighting WSN which exploit the Zigbee and 6LowPan technologies.
Siemens has introduced a building management WSN called APOGEE, that provides a WSN with self-healing capability.
Hitachi presented AirSense, a real time WSN monitoring system that controls the temperature, humidity and airborne particles.
Those systems have proved the reliability of the WSN for building management. However they are not able to interoperate with pre-existing sub-system and they cannot perform an extended pervasive WSN battery supplied able to operate for years, as should be in a cultural heritage building.
Thus we developed a new WSN able to perform an extended and pervasive network, completely battery supplied, that can last for years.
Our WSN is designed for cultural heritage building, presenting the following characteristics:
- Physical microsystem node size of few cubic centimeters, to reduce the aesthetic impact.
- Sensors lifetime from several years to tens of years, to reduce the maintenance.
- Field configurable and remote configurable, to easiness the installation and maintenance.
- Modularity, to provide higher exploitability.

- Dynamic deployable capabilities, in order to have an adaptable WSN.
- Easy extensibility, to support multi steps HVAC renovation.
- Multi sensors and external board sensors data collection capability, to support any kind of data acquisition.
- Easy pre-existing system integration.
- Optimized Trade-off between lifetime and quality of service (QoS).
- Standard protocols for WSN and thus ability to communicate with components already on-site.
- Wireless actuators, which can reduce installation effort considerably in historic building, where existing distributed autonomous components have to work together.

The document will continue with the:

- Specification required for a WSN deployed for energy efficiency, user comfort and cultural heritage preservation management in a historical building.
- Selection of microsystem electronic hardware.
- Selection of radio protocol.
- 3EncultWSN overview.
- New hardware design, to connect external sensors, actuators and sub systems.
- New firmware design.
- New radio protocol design.
- User interface: JavaTerminal.
- Deployment example.
- Testbed.
- 3Encult case studies installation.

1 Wireless Sensor Network specification

The development of a Wireless sensor Network (WSN) targeting the energy efficiency, user comfort and cultural heritage preservation of the historical building requires the following specification:

- **Real time operation**, the WSN has to collect the environmental data at real time to permit its integration with building management system. The data collected will be used by control HVAC algorithms inside a building management system which will control the HVAC preventing users discomfort, cultural heritage damage and energy waste. The climate and environmental dynamic of the building can be optimally monitored with a data period sample smaller than 15 minutes. A good tradeoff between building monitoring and control period and WSN network data traffic, data storage and energy consumption will be a sample period of 10 minutes. Therefore the new WSN will be optimized to work with data sample rate of 10 minutes.
- **Small component size**, a typical WSN is composed by several instances of an electronic microsystem able to collect data from sensors and send them remotely with wireless link. Those devices, called in a WSN terminology, nodes or motes, have to be a few cube centimeter size, in order to reduce their aesthetical impact in the cultural heritage building with painting and fresco, making the system embedded.
- **Easy installation**, the monitoring of environmental dynamic of the building requires the possibility to install a network node in a not easy accessible area. Then an easy installation characteristic will reduce the installation cost and increase the WSN deployment.
- **Battery supplied**, the monitoring of environmental dynamic of the building requires the possibility to install a network node in every place of the building. Then will be impossible to provide AC electrical supply to each node, especially in historical building with poor electrification. Moreover the WSN could be deployed in cultural heritage buildings with no electrification at all.
- **Excellent power management,** The battery supplied WSN devoted to monitoring cultural heritage building needs the state of the art power management system to permit its operability for years without battery replacement, reducing the system maintenance. The power management has to be able to delivery power only during the data sensors acquisition a data transmission. Between two samples acquisition and transmission the system has to go in an ultra-low power state, usually called *sleep* state, and go back to higher power state to perform the sensing and transmission, *waking up*, as fast as possible.
- **Standard radio protocol with low energy consumption**, the WSN have to be compliant with the radio emission law of each European countries. Therefore the use of a standard radio protocol will permit the operation of the WSN in all the countries. The IEEE802.15.4 radio protocol is worldwide accepted. Moreover this protocol is designed for Wireless Sensor Network

application with low energy consumption, which will permit the system to operate with batteries for years. The IEEE802.15.4 has great power management with excellent time response. Most of the commercial WSN systems are based upon the IEEE802.15.4 protocol, like Zigbee application. Then the use of IEEE802.15.4 permits the interoperability of the new WSN with already in the market systems.

- **Extended and pervasive monitoring distribution**, the monitoring of building environment dynamic requires the completely coverage of the building with high density data collection net. Therefore the WSN has to able to perform a wide distributed network with hundreds of nodes connected to each other.
- **Routing capability**, The IEEE802.15.4 protocol deploy an ultra-low power radio that can connect nodes far tens of meters in indoor condition. The radio link usually will be able to pass through a few walls. Then to permit the network to cover the entire building, the WSN should have the capability to retransmit the data of the far nodes to deliver the data to the data collector.
- **Multi sensors support**, the monitoring of environmental dynamic of the building to perform energy efficiency, user comfort and cultural heritage preservation management, requires the deployment of several kinds of sensors, like: air temperature and humidity; wall surface temperature and moisture; inner wall temperature and moisture; light intensity; electromagnetic radiation intensity; heat flux; air quality and pollution; wind speed; sound noise level etc. Therefore the WSN has to have the capability to read, analyze and manage several kinds of sensors, providing their power supply and electrical data acquisition interface.
- **High computation performance**, the mote of the WSN needs a higher computational capability to perform complex network structure and to manage the data acquisition from several kinds of sensors. In fact each kind of sensor needs its own procedure, which usually imply the electrical signal and power management. Moreover several types of sensor need a first stage data analysis algorithm to convert the data collected in a meaningful data with suitable resolution. The execution of the first stage analysis algorithm in the mote will reduce the data transmission size, decreasing the network traffic and total power consumption, due to the fact that the transmission is the most power hungry action of the network.
- **Local data storage**, each mote should be able to collect the sensors data in a local data storage, like microSD card, in order to preserve the information gathered. The local data storage capability can be deployed to recover error in wireless data transmission and even to use a single mote as data logger to perform cheap monitoring of building for off line behavioral analysis.
- **Low maintenance**, the low maintenance is a key feature that is a must to keep the WSN monitoring system cost extremely low. The system have to require maintenance only to replace the battery after a long period of life, like years. The use of energy harvesting system can permit to deploy less cheap nodes able to collect energy form the surrounding environment in order to

work perpetually. The harvested motes can operate theoretically forever without maintenance.

- **Ultra low power consumption**, the battery powered WSN must have the ultra-low power consumption in order to operate for years. Each aspect of the network work has to be addressed in term of power consumption, like data transmission, data sensors acquisition and wait state. Each working phase has to be performed in the most energy efficiency way exploiting the affordable state of the art technology.
- **In field configuration**, the configuration of the network has to be performable in field, permitting the technician to do the correct setting. Both automatic and manual configuration should be performable to obtain the most suitable WSN deploy in a short time. In fact the most efficient deployment configuration is strictly correlated to the building architecture and WSN usage. Therefore a manual setting can sometime over perform the automatic setting. Automatic setting is essential to let everyone to use the new WSN system.
- **Over the air firmware update**, the capability to perform over the air firmware update permits the continuous system development and improvement, even in already installed WSN. The firmware update can add functionality, change behavioral network characteristic and outcome deadly errors. The over the air firmware up date has to addresses all the nodes in the network or only a few that need special treatment.
- **Remote control**, to reduce the maintenance expense and provide user support, remote control is a feature that permits the developers, system management and specialized technicians to provide cheap assistance. Moreover the remote control can be deployed by remote system to interact actively with the network. The remote control has to have the capability to modify the network parameter and configuration at run time.
- **User friendly interface**, the user friendly interface is a feature which is essential to let normal users to install and deploy the new WSN system.
- **Good system integration capability**, the open structure of the new WSN has to provide a good integration capability to interaction with: Building Management Systems; building automation systems (KNX, LonWorks) and old fashion system with relays.

2 Hardware selection

UNIBO team has analyzed several platforms already present in the market like iMote, CrossBow, xBee ecc, searching a good platform to develop the WSN. The parameters analyzed to select the platform were: architecture flexibility, power consumption, software accessibility, hardware programmability, platform connectivity and ability to expanse, internal sensors.

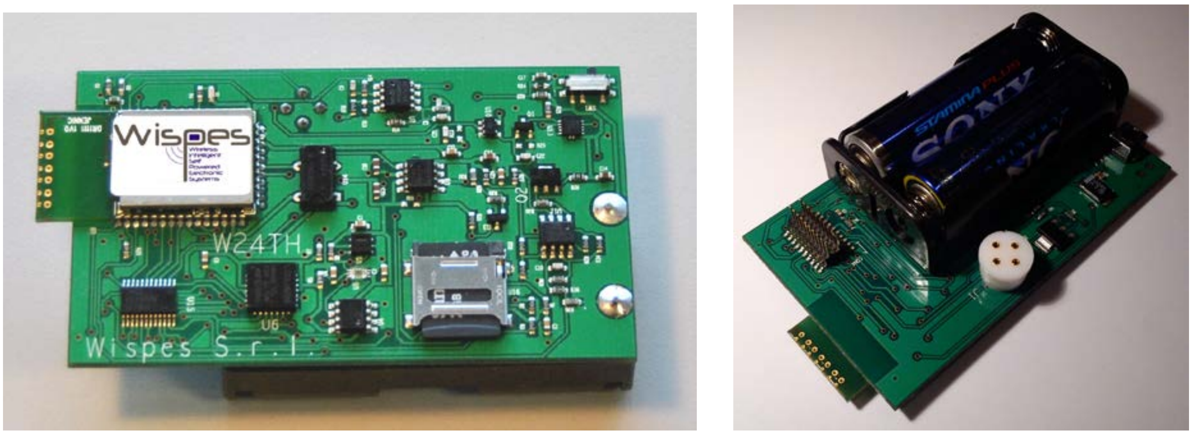

Fig. 2.1 Wispes W24TH mote

The choice has been the Wispes W24TH node, which is a wireless sensor device compliant to Zigbee PRO protocol and IEEE802.15.4. The node can act as coordinator, router or end-device depending on the downloaded firmware and can build-up network with **thousands** of nodes (ZigBee capability).

The core of the system is a **32bit** microcontroller with a **2.4GHz** radio transceiver. **32MHz** CPU clock permits the development of complex applications and **distributed data processing**. The system is provided with a series of on-board sensors:

- Temperature sensor;
- Humidity sensor;
- 3-axis Accelerometer sensor;
- Ambient light sensor;
- Mox gas sensor (VOC, O3, NOx, NH4, CO ...).

Moreover, W24TH is equipped with a microSD card reader for **local storing, data logging and backup**, a USB battery charger, a 32kHz quartz oscillator with **real time clock** library, a 20-pin expansion connector and a power management subsystem.

The state of the art **power management** subsystem provides the facility to switch off via software the whole system with the exception of the microprocessor, enabling 8µA power consumption in sleep mode for **extremely long battery life**.

The 20-pin expansion connector features UART bus, SPI bus, I2C bus, 10 GPIOs, 12bit ADC and 3.3V power supply. This allows the W24TH to be connected to several expansion boards such as multi-channels acquisition interfaces, actuators, USB, KNX devices, energy harvesters and more.

The microSD card reader is connected to the microprocessor and can be completely electrically disconnected by the power management subsystem.

The Node is provided with free license development tool kit, ZigBee APIs, 802.15.4 APIs, W24TH sensor APIs and power management APIs. Firmware upgrade is permitted by using download over-the-air (OTA) facility.

2.1 Power management system

The power management system provides:

- MicroSD reader power control
- Board power supply control
- I2C and SPI electrically disconnection and connection
- USB Battery charger
- USB detection
- DC/DC converter control
- Battery charge status

2.2 On board sensors and facilities

The sensors embedded on the board are:

2.2.1 Temperature sensor

- Codification 14bit
- Resolution 0.01°C
- Accuracy ±0.3°C
- Operating range -40 to 125°C

2.2.2 Humidity sensor

- Codification 12bit
- Resolution 0.04 %RH
- Accuracy 2 %RH
- Operation range 0 to 100RH

2.2.3 Ambient light sensor

- I2C interface
- Spectral responsibility is approximately the human eye response
- luminance to digital Converter
- Wide range and high resolution (0.23 – 100000 lx)
- Small measurement variation (±15%)

2.2.4 3-axis ±2g/±6g digital output voltage linear accelerometer

- I2C/SPI interfaces
- Programmable 12 or 16 bit data presentation
- Interrupt activated by motion
- Programmable interrupt threshold
- Resolution 1mg
- 12 bit data converter

2.2.5 Multi brand full controlled MOX gas sensor interface

- TO-39 (solid TO-5) socket
- Heater software voltage regulator (3% resolution)
- Heater current reader (7% resolution)
- Sensor resistance reader (10% resolution)
- Multi brand sensors facility
- Gas measured: **VOC, CO, CO2, NOx, O3, NH4**

All the sensors can be electrically disconnected by the power management subsystem.

2.3 Package

The W24TH can be allocated in a 98x54x29mm box with cuts to let the external air come in. The cuts have the right size to let the internal air to be at equilibrium with the external environment with application of an acceptable delay. Therefore the built-in temperature and humidity sensors can read the external values correctly. Moreover the box has a transparent cover that permit the internal light sensor to read the surrounding environment light intensity.

Fig. 2.2 Wispes W24TH inside the 98X54X29mm box

2.4 W24TH Expansion Connector

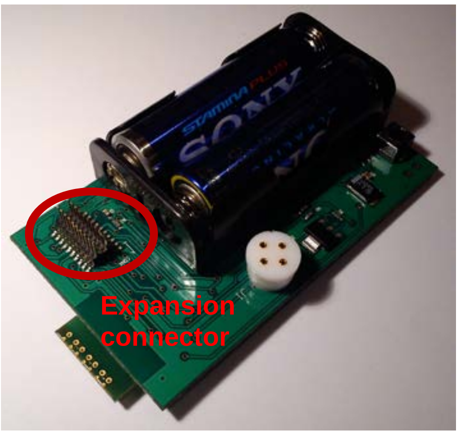

Fig. 2.3 W24TH expansion connector

The W24TH has a 20-pins expansion connector that provide analog and digital interface to connect external sensors and digital system. In detail the connector provides:

- Digital buses, featuring the most common communication protocols (UART, SPI, I2C) which directly connect any digital device to the internal microprocessor (e.g. digital temperature sensors, USB interface, KNX interface, accelerometer, GPS, electronic compass, etc.).

- 10 General Purpose Digital Input/Output. It is possible to interface up to 10 digital signals both as input and as output. Furthermore GPIOs allow to count digital pulses and to read variable frequencies. With GPIO it is possible to connect finite-states sensors like open/close door and window sensors, or finite-state actuators like on-off switches. Moreover it is possible to generate pulses with variable length that can be used to control dimmer actuators.

- Input analog signals with 12bit Analog Digital Converter provides the capability to read any analog sensor. (e.g. temperature sensors, light sensors, etc.)

- DC power supply. The connector provides a filtered voltage supply to power directly external sensors or expansion boards. The maximum output power is 1.65Watt at 3.3V (which means a maximum output current of 500mA). By the exploitation of common DC/DC converters it is possible to extract higher output voltage at the cost of reduced current limits (e.g. 5V at 330mA max) or viceversa (e.g. 1V as output power supply with a peak of 1.65A). The DC power supply can be switched on and off by the microcontroller.

- Mote power supply, to externally provide energy to the W24TH allowing the exploitation of harvesting system like solar cell, wind micro turbine, etc. The mote can be supplied from 0.9 to 6.5V.

- JTAG for Firmware uploads and debug. Direct connection to the microcontroller permits a faster firmware upload and debug that is essential to support the firmware development.

The features provided by the expansion connector allow extensive system enrichment and heterogeneity. In fact will be possible to realize a WSN based upon the W24TH where each node has its own expansion board that provides different extension solution. For example it is possible to connect directly simple sensors or specific designed expansion boards such as:

- Multi-channel acquisition interfaces (e.g. for several external temperature sensor, humidity sensors, moisture sensors, etc.).

- Multi-channel actuators (e.g. on-off switches, dimmer actuators, etc.).

- Interface with commercial home and building automation protocols (e.g. LonWorks ,KNX);

- Interface with industrial and home data buses (RS232, RS422, RS485);

- USB and Bluetooth interface to directly manage the WSN using PC, handheld devices or Smartphones;

- Enhanced gateways to the internet (e.g. providing Ethernet, WiFi, GRPS/UMTS modem);

- Additional and enhanced power modules. For example, Energy Harvesters for indoor and outdoor scenarios which keep charged the internal batteries exploiting environmental energy (e.g. solar, wind, thermal gradients, vibrations etc.).

2.4.1 Expansion connector specification

Here is a list of the pin functionalities extract from the datasheet of the node.

The expansion connector is 50x100 mils pin spacing, 2 rows, and 20 ways. The pins are:

1) GPIO4; UART CTS0; JTAG TCK
2) GPIO5; UART RTS0; JTAG TMS
3) GPIO6; UART TXD0; JTAG TDO
4) GPIO7; UART RXD0; JTAG TDI
5) GPIO13
6) GPIO18
7) GPIO17
8) GND
9) $\overline{\text{RESET}}$
10) $\overline{\text{PROGRAM}}$
11) Vcc_USB (input from 0.9 to 6.5V)
12) Vccd (output controlled by the power management)
13) GPIO1; SPI SEL2; Pulse counter0 input
14) SPICLK
15) SPIMISO
16) SPIMOSI
17) GPIO14; I2C CLK

18)GPIO15; I2C DATA
19)Vdcdc
20)ADC4

3 Radio Protocol selection

The radio protocol selection has been performed taking into account the specification listed in the chapter 1.The main keys features for the new WSN radio protocol has to be: The compliancy with the national law for electromagnetic radiation, low energy consumption, several hundred of network motes supported, pervasive and extended network, routing capability, all nodes batteries supplied. The standard radio protocol IEEE802.15.4 is defined for WSN application permitting the deployment of 65,535 nodes. This protocol has low power consumption, smaller than WiFi and Bluetooth, and fast wake up response time, faster than WiFi and Bluetooth. However the IEE802.15.4 does not support routing capability and a radio protocol built on top of the IEEE802.15.4 should be added to the firmware to manage routing capability. In the market ZigBee is an 802.15.4 based protocol suitable for cultural heritage building WSN, however it cannot permit to the router mote to sleep. Then we realized our protocol layer on top of the IEEE802.15.4 to perform the most suitable WSN for cultural heritage building. Hereafter there is an explanation of ZigBee protocol and our WSN protocol called 3EncultWSN protocol.

3.1 ZigBee radio protocol

ZigBee is a specification for a suite of high level communication protocols used to create personal area networks built from small, low-power digital radios. ZigBee is based on an IEEE 802.15 standard. Though low-powered, ZigBee devices can transmit data over long distances by passing data through intermediate devices to reach more distant ones, creating a mesh network; i.e., a network with no centralized control or high-power transmitter/receiver able to reach all of the networked devices. The decentralized nature of such wireless ad hoc networks make them suitable for applications where a central node can't be relied upon.

ZigBee is used in applications that require only a low data rate, long battery life, and secure networking. ZigBee has a defined rate of 250 kbit/s, best suited for periodic or intermittent data or a single signal transmission from a sensor or input device. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range wireless transfer of data at relatively low rates. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth or Wi-Fi.

ZigBee networks are secured by 128 bit symmetric encryption keys. In home automation applications, transmission distances range from 10 to 100 meters line-of-sight, depending on power output and environmental characteristics.

ZigBee was conceived in 1998, standardized in 2003, and revised in 2006. The name refers to the waggle dance of honey bees after their return to the beehive.

ZigBee is a low-cost, low-power, wireless mesh network standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications. Low power usage allows longer life with smaller batteries. Mesh networking provides high reliability and more extensive range. ZigBee chip vendors typically sell integrated radios and microcontrollers with between 60 KB and 256 KB flash memory.

ZigBee operates in the industrial, scientific and medical (ISM) radio bands: 868 MHz in Europe, 915 MHz in the USA and Australia and 2.4 GHz in most jurisdictions worldwide. Data transmission rates vary from 20 kilobits/second in the 868 MHz frequency band to 250 kilobits/second in the 2.4 GHz frequency band.

The ZigBee network layer natively supports both star (each device is radio connected to the coordinator device) and tree (there are routers devices able to repeat the radio signals coming from other devices and leaves devices that can not repeat radio signal. The network has a tree structure with a parenthood to each device and the root device is called coordinator) typical networks, and generic mesh networks (router devises can repeat the radio signal to other routes without respect the parenthood as in the tree network). Every network must have one coordinator device, tasked with its creation, the control of its parameters and basic maintenance. Within star networks, the coordinator must be the central node. Both trees and meshes allow the use of ZigBee routers to extend communication at the network level.

ZigBee builds upon the physical layer and media access control defined in IEEE standard 802.15.4 (2003 version) for low-rate WPANs. The specification goes on to complete the standard by adding four main components: network layer, application layer, ZigBee device objects (ZDOs) and manufacturer-defined application objects which allow for customization and favor total integration.

Besides adding two high-level network layers to the underlying structure, the most significant improvement is the introduction of ZDOs. These are responsible for a number of tasks, which include keeping of device roles, management of requests to join a network, device discovery and security.

ZigBee is not intended to support powerline networking but to interface with it at least for smart metering and smart appliance purposes.

Because ZigBee nodes can go from sleep to active mode in 30 ms or less, the latency can be low and devices can be responsive, particularly compared to Bluetooth wake-up delays, which are typically around three seconds.[4] Because ZigBee nodes can sleep most of the time, average power consumption can be low, resulting in long battery life.

The application layers are:

- ZigBee Home Automation 1.2
- ZigBee Smart Energy 1.1b
- ZigBee Telecommunication Services 1.0
- ZigBee Health Care 1.0
- ZigBee RF4CE – Remote Control 1.0
- ZigBee RF4CE – Input Device 1.0
- ZigBee Light Link 1.0

- ZigBee IP 1.0
- ZigBee Building Automation 1.0
- ZigBee Gateway 1.0
- ZigBee Green Power 1.0 as optional feature of ZigBee 2012

However the Zigbee protocol does not permit the routers to sleep, than a Tree or Mesh network should have the router always on reducing their operative time with small commercial batteries.

In a typical Zigbee network with Tree or Mesh topology, the router and the coordinator are not batteries supplied.

3.2 3EncultWSN radio protocol

The 3EncultWSN is built upon the IEEE802.15.4 radio protocol, permitting the network to be compatible with ZigBee devices.

The new radio layer is able to manage and create both star (each device is radio connected to the coordinator device) and tree network topology (there are routers devices able to repeat the radio signals coming from other devices and leaves devices that can not repeat radio signal. The network has a tree structure with a parenthood to each device and the root device is called coordinator). The new protocol permits the router to sleep. The capability of the router to sleep is essential to monitoring Cultural Heritage Building. In fact the monitoring can be performed for years in building without electrical current or with reduced current plug. To cover the whole building the network should have a tree structure due to the low radio power of each mote. Moreover having routers able to sleep avoids the realization of “leaves” devices making the network structure and installation easier. The 3EncultWSN has the capability to auto configure the network structure like ZigBee, furthermore gives to the user the possibility to customize the network with manual configuration. The 3EncultWSN provides all the information to analyse the network performance and permit to the user to do the network maintenance and configuration remotely. With respect to Zigbee, the new wireless system has a strong time synchronization that avoids data collision and provides better data time stamp. Others functionalities introduced are: Remote network management, remote device configuration, over the network firmware update and network clock time.

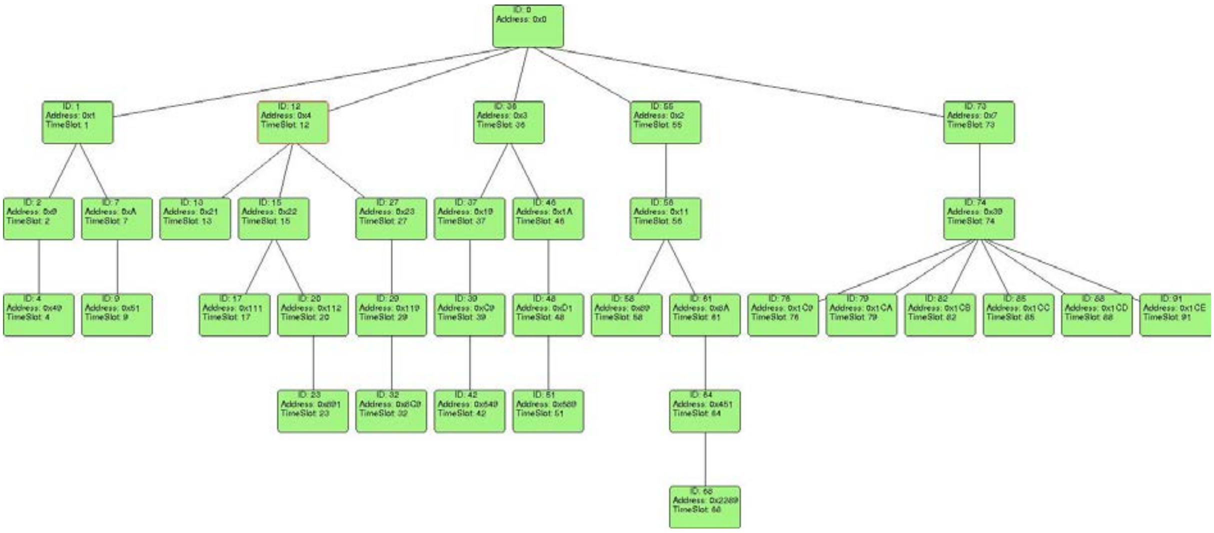

Fig. 3.1 3EncultWSN Tree network topology deployed in the "Palazzina della Viola"

4 3EncultWSN overview

The 3Encult project has the aim to develop a new Wireless Sensor Network designed for cultural heritage building. The system developed is network of small electronic boxes capable to collect environment data and send them via radio. The system is called 3EncultWSN.

4.1 Hardware

Fig. 4.1 Wispes W24TH mote

The engineers involved in the design of the 3Encult WSN has chosen as hardware platform the “Wispes W24TH” mote, due to its high computational power, the ultra-low power capability (8µA), SD card reader and several environment sensors on board.

In particular the W24TH has a 32bit 32MHz microcontroller, 128Kbyte of RAM, battery recharger, 802.15.4 radio transceiver, accelerometer, temperature sensor, humidity sensor, light intensity sensor, gas sensor connector and expansion connector. Everything is enclosed in a 98x54x29mm box and supplied from 2 AA batteries.

4.1.1 Sensor Specifics:

Temperature: 14bit codification, 0.01°C resolution, ±0.3 accuracy, -40 to 125°C operative range.

Humidity: 12bit codification, 0.04% RH resolution, ±2% RH accuracy, 0 to 100% RH operative range.

Ambient light sensor: 0.23 to 100,000 lux operative range, ±15% Accuracy.

4.2 3Encult WSN software

The 3Encult WSN software has been designed to completely satisfy the specifics required to monitoring an heritage building, like: easiness of installation, long monitoring period, low maintenance, remote control, remote maintenance, remote setting, remote software update, high data collection reliability, system compatibility, large network support and long battery last.

The Software developed is able to manage the entire network, from its installation and configuration, to the data collection and visualization.

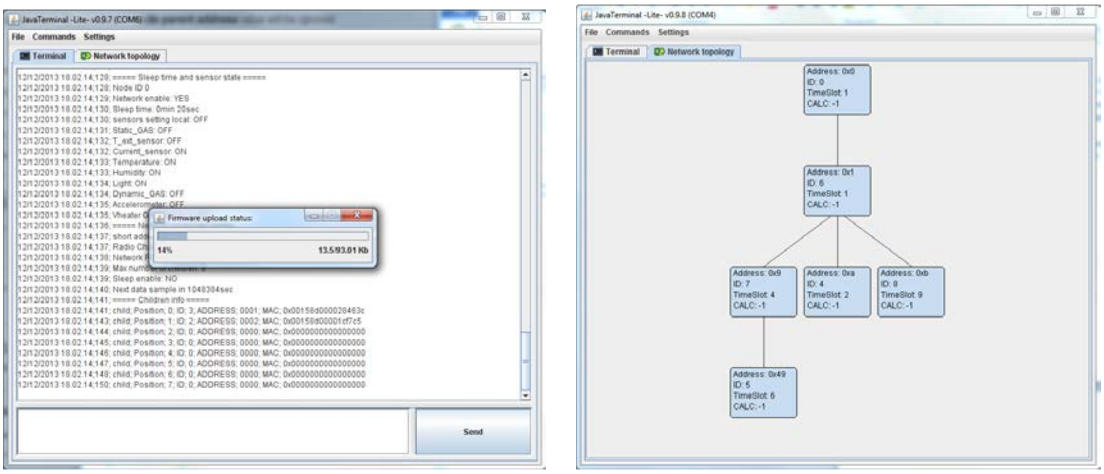

Fig. 4.2 Java Terminal Application

The installation part is controlled from the JavaTerminal application. The JavaTerminal application is able to run in several platform like PC, Tablet, PC board and Smartphone. Using the terminal the user can install, configure and send data to a remote database or web server.

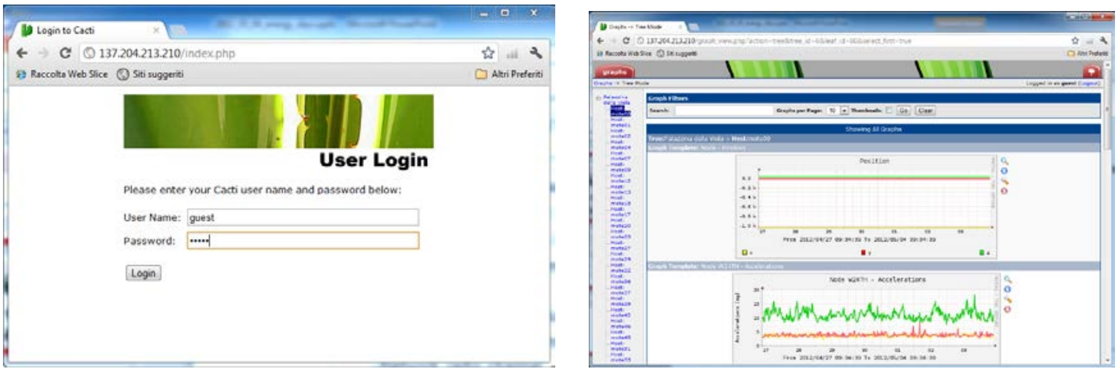

Fig. 4.3 Web visualization interface

The data collection and visualization is done from a web server using the “Cacti” interface.

Finally, the 3EncultWSN can be interfaced with the Building Management System developed by Cartif partner and with KNX building automation standard.

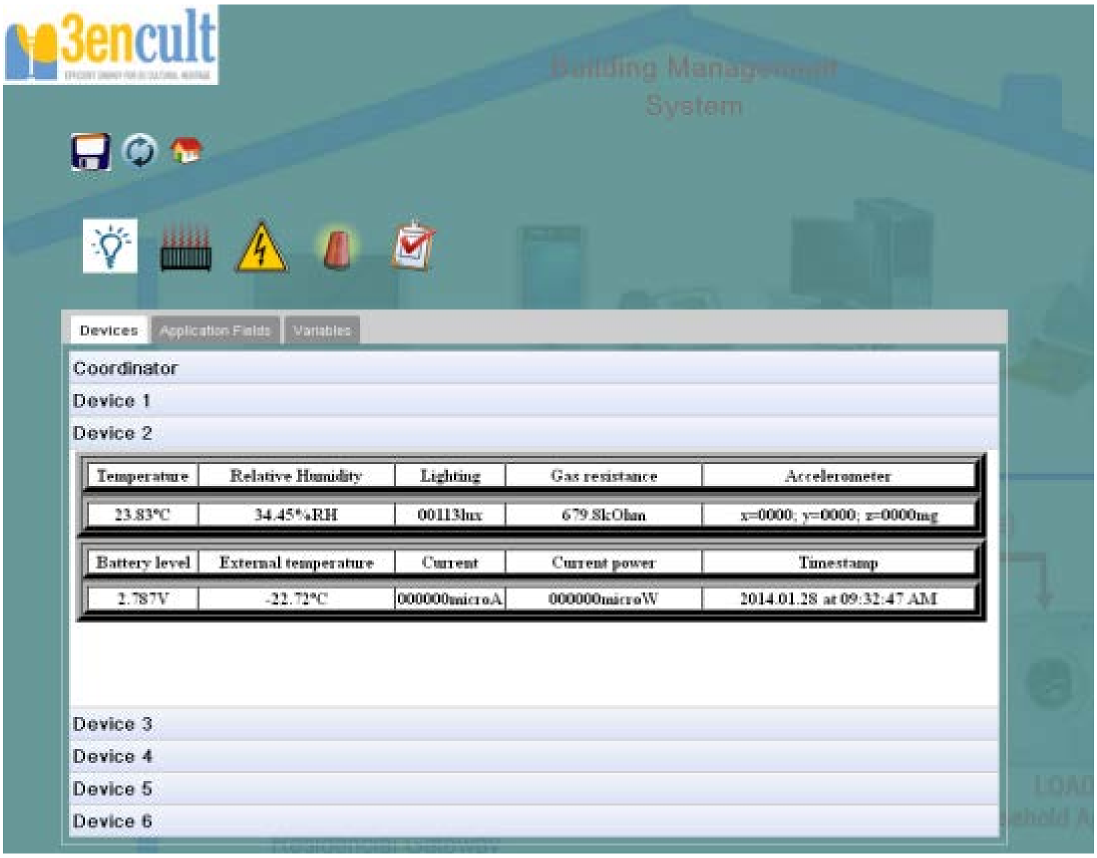

Fig. 4.4 3Encult Building Management System

5 Hardware design

The hardware design has concerned the realization of extension boards for the W24TH motes. Those boards aimed to provide all the features needed to address the cultural heritage building management in terms of energy consumption, user comfort and cultural heritage preservation.

The expansion board are divided in three types, sensors, actuators and energy harvesting. The expansion structure permit the plugging of more than one expansion boards in a stack configuration, therefore a single mote can be expanded with energy harvesting capability, actuators and external sensors.

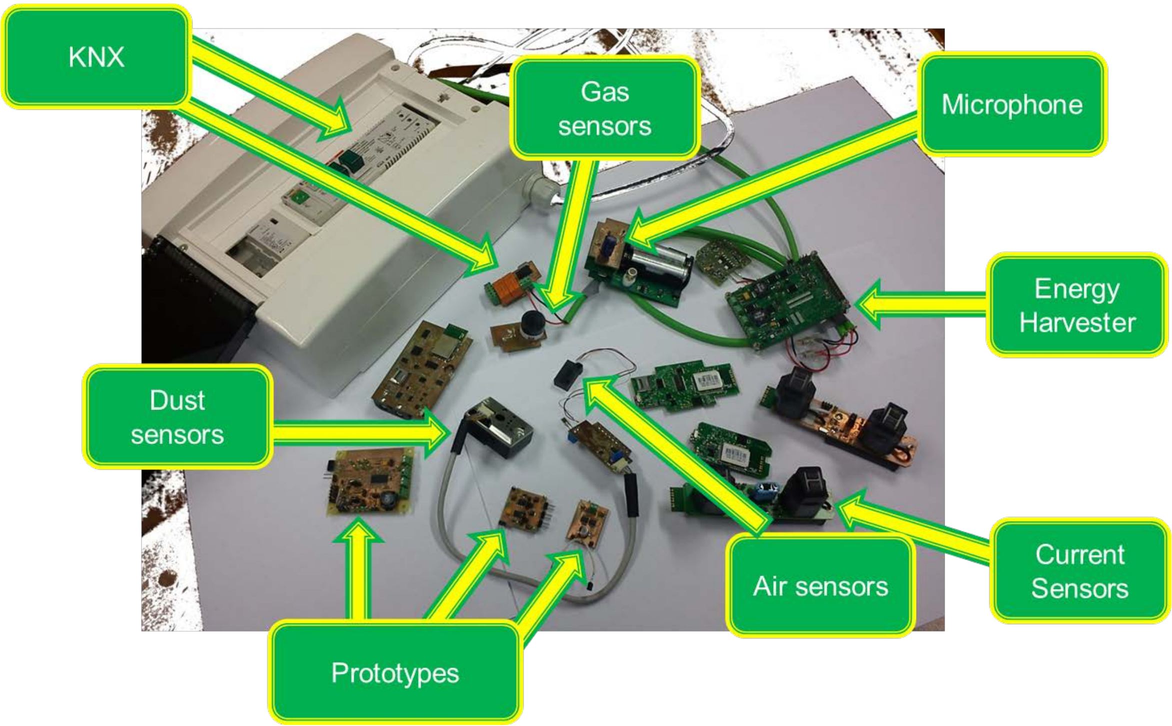

Fig. 5.1 Expansion board prototypes

5.1 Sensor expansion boards

The sensors expansion boards design has focused the realization of sensors interfaces to collect data from external sensors. The developed boards are as follow.

5.1.1 Eight channel analog sensors interface

The eight channel analog sensors interface is a board able to read data from analog sensors. The interface is designed to have eight input channel usable as single interface or in a couple to read sensor with differential output. Each channel can read voltage, current, resistance or admittance sensors output, therefore it is possible to plug analog sensors of temperature, AC-DC current, heat flux, light intensity, etc.

5.1.2 Gas sensors interface

The gas sensors interface is a board able to deploy chemical gas sensors. Those sensors have low power consumption and high resolution at a bit expensive cost. The board can operate with several sensors brand and can cover several type of gas detection, from CO to formaldehyde.

5.1.3 Humidity sensors interface

The humidity sensors interface permits to read analog capacitive humidity sensors.

5.1.4 Air velocity interface

The air velocity board permits the reading of air velocity MEMs sensor from Omron with a range from 0 to 3 m/s.

5.1.5 Microphone board

The microphone board has an on board mems microphone capable to measure the noise level. Moreover the board has the support to manage its power consumption and to wake the W24TH microcontroller when a noise is detected.

5.1.6 Dust sensors interface

The dust sensor interface is a board able to collect data from the Panasonic dust sensor. The sensor can detect dust and PM10.

5.2 Actuators expansion boards

The actuators expansion boards design has focused the realization of actuator interfaces to perform wireless control. The developed boards are as follow.

5.2.1 Relay expansion

The relay expansion board has two relays and their control electronics. With the relay expansion board is possible to control every kind of electrical actuator like windows automation, electrical heater, air conditioner and lights.

5.2.2 KNX interface

The KNX interface has the capability to interface the 3EncultWSN with a standard KNX network. The interface and KNX W24TH firmware permit the WSN to be seen as a branch of the KNX network. Therefore the 3EncultWSN can be manage as all the standard KNX devices.

5.3 Energy harvesting boards

The energy harvesting board is able to manage power transducers which convert the surrounding environment energy in electrical power. The board has all the electronics to perform high efficiency energy collection from AC current, solar cell and micro wind turbine.

6 Firmware design

The firmware design concerns the code algorithms and functions which have to run in the W24TH microcontroller to perform all the required actions by the 3EncultWSN.

Here in the following all the macro areas of firmware developed are described, with the exception of the part of wireless protocol management that will be treated in the next chapter.

6.1 Task scheduler

The task scheduler is a code function that permits to manage and execute several code tasks. Each task is a code algorithm that performs some process, data analysis, data collection and action.

The developed scheduler is a non-preemptive and task priority algorithm. Every task has the execution priority that instructs the scheduler how import is the execution of the task. More the priority is higher and sooner the scheduler runs the task. The non-preempitve characteristic is that the schedule will execute the task until is end, therefore each task should take smaller execution time as possible, to let other tasks to run. The scheduler has a task queue where there is the list of waiting tasks with their priority. The scheduler checks the queue and executes the task with highest priority. At the end of execution the task is removed from the scheduler queue and the scheduler performs again the queue scan and execution until to clear the whole queue. The scheduler has functions to insert tasks in the execution queue. Those functions can insert task in the queue instantaneously, cyclically or after a fixed amount of time. Those functions can be used from every task that can call the execution of itself or of other task. The scheduler functions use the internal microcontroller tick timer with a precision of 62.5nsec. Therefore it is possible to delay a task execution from 62.5nsec to 268.43sec or cyclically execute a task every 62.5nsec to 268.43sec. For example the LED blink is managed by a task that calls itself cyclically every 1 sec to switch on the LED and delayed 0.1sec to switch off the LED.

6.2 Power management

The power management firmware controls the power consumption of the mote in each execution phases. The code acts to the hardware power management system to switch on and off the peripheral power supply, to put in sleep, run states the microcontroller and to turn on and off the radio system. Moreover the power management control the battery charge state and the battery recharge process. The power management system is able to put the entire system in an ultra-low power mode for a programmable amount of time. The programmable time can be set from 30µsec to 1.5days. In the 3EncultWSN the power management system: turns the microcontroller on (9mA consumption) during the sensors reading and transmission phases; turns on the sensors only in the sensors reading phases; turn on the radio system only during transmission time (20mA consumption); switch off the entire system between to samples interval (8µA consumption). The power management is able to estimate the battery charge consumption at run time. The battery charge consumption is a data sent from the mote to the gateway.

6.3 Sensors management

The sensors management task executes the reading procedure of each sensor. The task is programmable to execute only the sensors procedure selected by the end user in the way to save battery energy. Each procedure manages the sensor interface and executes a first layer of data analysis to make the read values meaningful. The sensors procedures developed are:

- On board temperature sensors. The procedure starts up and reads the on board digital temperature sensors. The data collected are converted in 0.01C units.
- On board humidity sensors. The procedure starts up and read the on board digital humidity sensors. The data collected are converted in 0.01%Rh units.
- On board light sensors. The procedure starts up and read the on board digital light intensity sensor. The data collected are converted in 1lx units.
- On board accelerometer, position. The procedure starts up and read the instant accelerometer values to the three axis. The data collected are converted in 1mG units.
- On board accelerometer, vibration. The procedure read 100 samples in 0.15 sec from each axis and calculates the maximum variation. The data collected are converted in peak to peak vibration intensity with frequency spectrum from 6.67Hz to 350Hz.
- On board Gas sensor, static. The procedure manages the heater voltage of the gas sensor and provides the resistance value of the sensor after 1sec of heating.
- On board Gas sensors, dynamic. The procedure manages the heater voltage of the gas sensors and collects 500 samples of the resistance value of the sensor during the first second of heating. The procedure returns the minimum value of the sensor resistance read.
- External temperature sensor. The procedure setups the external temperature sensor and read 20 samples of it resistance value. The procedures returns the mean value converted in 0.01C units.
- External humidity sensor. The procedure setups the external capacitive humidity sensor and reads the sensor capacitive value. The capacitive value is converted in 0.01%Rh units.
- External AC current sensor. The procedure setups the external AC sensor and collect 200 samples for 40msec. The procedure returns the root mean value converted both in 1µA units and 1µW units.

- External gas sensors. The procedure setups the external gas sensor providing the correct voltage polarization, then reads the current sensor output after 1 second polarization and converts it in 1ppm units.
- External air velocity sensor. The procedure setups the external air velocity sensor and reads 10 samples of its voltage output. The procedure returns the mean value converted in 0.01m/s units.
- External dust sensor. The procedure setups the external dust sensor and reads 10 samples of it voltage output. The procedure returns the mean value converted in 1ppm units.
- External Microphone sensor. The procedure setups the external microphone sensor and collect 1600 samples in 20msec. The samples are analysed to return the noise dbm value and to detect the presence of human speech.

6.4 SD local storage

The SD local storage is a set of functions that permits to save data collected in the local microSD card. The functions are able to create and manage the SD FAT file system, to create, read and write files and directories. The mote firmware can be set to store locally the sensors data with their time stamp expressed in day:month:year hour:minute:secod:millisecond. The local data are stored in a ID.csv file where ID is the user set mote identifier number. Moreover the mote firmware can store in the local SD the log of its activity in the log.txt file.

6.5 Stand-alone operative mode

Each node has the possibility to work in a standalone mode and collect the data from the environment in a micro-SD card. This feature permits to monitoring remote place that does not need real time data transmission, avoiding the installation of network gateway and network router nodes.

- Each device is able to collect data in a micro-SD card.
- Data will be saved in a CSV file readable by excel, where the first line is the sample time stamp.
- The device is configurable via USB and it is possible to set:
    - Sensor enable and power (internal temperature, internal humidity, light intensity, external temperature, vibration, gas concentration, current sensor)
    - Sensor sample period (samples every 4 sec to 1.5 days)
    - Date
    - Device ID
- Capability to work for several years with two commercial AA batteries.

6.6 Software interface

The software interface is a set of tasks and functions that permits the mote configuration and data and mote state delivery with the UART connection. The UART can be converted afterword to both USB and Bluetooth connection with their expansion boards. Therefore the software interface is able to receive command from the user and to provide mote state information and data. The communication is based upon text exchange between mote and external devices like PC, smartphone, tablet etc. Each exchange will be further referring as a print. Each mote print line starts with the time stamp of its execution. The mote has a list of commands and several output information, like the data collected, internal state and networking state. The communication between the mote and a PC can be seen by the end user with a terminal application like putty, hyperterminal, minicom and our developed terminal JavaTerminal.

6.7 Data print format

The mote prints its own data and the received data from its children using the following format:

27/02/2014 02.47.44;717; DATA NODE ID;0001;0001;00049;0001;0000;-039;2253;5682;00064;-2287;0000;000000;000000;0000;0000;0000;0000;0000;0000;2604;02102047;000000e1;S4.1

The data from a mote is expressed in a line where there are fields separated by semicolon. Afterward the explanation of each field content related to the above example.

1) Time stamp: day/month/year hour.minute.second.millisecond (e.i. 27/02/2014 02:47:44 717 millisec.).
2) DATA NODE ID it is the start print that univocally identify the data line.
3) Node ID, it is a number that univocally identify the WSN node in the network (e.i. node ID 1).
4) Time slot, it is a number that univocally identify at which time the node has to send its data packet. (e.i. Time slot 1, is the first node to send data).
5) Packet number, identify how many sensor sample the node has done till now (e.i. 49 samples performed).
6) Short address, it is the network address assigned to the node (e.i. short address 0x1).
7) Parent short address, it is the short address where the node will send its data (e.i. parent address 0x0, the coordinator).
8) Link Quality, it is the dbm value of the radio signal strength. (e.i. Link quality -39dbm)
9) Temperature, it is the temperature collected from the sensor present on board, the value it is expressed in 0.01C units (i.e. Temperature 2253 = 22.53C)
10) humidity, it is the humidity collected from the sensor present on board, the value it is expressed in 0.01%RH units (i.e. Humidity 5682 = 56.82%RH)

11) light intensity, it is the light intensity collected form the on board sensor and it is expressed in lux (e.i. Light intensity 64lux).
12) External temperature, it is the temperature collected from an external temperature sensor, the value it is expressed in 0.01C. (i.e. External temperature -2287=-22.87C, the sensor is not connected).
13) Gas sensor, provide the resistance of the gas sensor and it is expressed in 100ohm units (e.i. Gas sensor 0, sensor off).
14) AC current sensor for monitor appliances (current clamp), current intensity expressed in microA units (e.i. Current 0, sensor off).
15) AC current sensor expressed in watt for 230Vac, power expressed in microwatt units (e.i. Power 0, sensor off).
16) On board accelerometer, vibration intensity in the x axis expressed in mG (e.i vibration 0, sensor off).
17) On board accelerometer, vibration intensity in the y axis expressed in mG (e.i vibration 0, sensor off).
18) On board accelerometer, vibration intensity in the z axis expressed in mG (e.i vibration 0, sensor off).
19) On board accelerometer, static acceleration to detect the gravity direction, axis x expressed in mG. (e.i acceleration 0, sensor off).
20) On board accelerometer, static acceleration to detect the gravity direction, axis y expressed in mG. (e.i acceleration 0, sensor off).
21) On board accelerometer, static acceleration to detect the gravity direction, axis z expressed in mG. (e.i acceleration 0, sensor off).
22) Battery voltage expressed in mV units (e.i. Voltage 2604 = 2.604V).
23) Battery charge burned, the battery charge burned until now, expressed in microCoulomb (divide to 3.600.000,00 to have mAh) (e.i. 2.102.047µC = 0,58mAh).
24)Debug field.
25)Firmware version (e.i. S4.1, SleepyEnd 4.1).

6.8 Mote commands

The mote has a list of command. Each command can be executed at run time and interact with the user producing an output reply print. In the most of cases the command as an explanation line with usage example. Some command has a sub menu that performs complex action. Hereafter the exhaustive list of the commands present in the firmware. In red are the prints sent from the pc and in black the mote reply prints. Each reply line starts with the time stamp print expressed in day/month/year hour.minute.second;millisecond like 26/02/2014 15.18.24;489 is the 26 February 2014 at 3:15:18 p.m.

6.8.1 Command h

The command h prints the help menu.

```
h
```

26/02/2014 13.28.35;292; ===== Help =====

26/02/2014 13.28.35;292; Firmware version S4.1

26/02/2014 13.28.35;293; h: help

26/02/2014 13.28.35;293; w: sleep time in seconds

26/02/2014 13.28.35;294; q: local sensor state

26/02/2014 13.28.35;295; p: program sensor state

26/02/2014 13.28.35;295; i: select node ID

26/02/2014 13.28.35;296; l: select node transmission time slot

26/02/2014 13.28.35;297; r: reset

26/02/2014 13.28.35;297; o: set date and time

26/02/2014 13.28.35;298; s: enable/disable sensor sample

26/02/2014 13.28.35;298; a: set node parent short adddress

26/02/2014 13.28.35;299; ns: perform a network scan

26/02/2014 13.28.35;300; n: enable/disable network

26/02/2014 13.28.35;301; m: control remotely the node, press m to print the help

26/02/2014 13.28.35;301; k: reset configuration (flash reset)

26/02/2014 13.28.35;302; x: change network: pan ID, channel, max children

26/02/2014 13.28.35;303; t: print the node status (human readable)

26/02/2014 13.28.35;304; u: print the node status (parser readable)

26/02/2014 13.28.35;305; v: Set Vheater GAS sensor

26/02/2014 13.28.35;305; c: Reset child position

26/02/2014 13.28.35;306; format: format sd card

26/02/2014 13.28.35;307; fp: print sd card file content

26/02/2014 13.28.35;307; fl: list sd card files

26/02/2014 13.28.35;308; fd: delete sd card files

26/02/2014 13.28.35;309; fm: move sd card file

26/02/2014 13.28.35;309; ff: create sd card folder

26/02/2014 13.28.35;310; cntrl+b: to update the sleepy end firmware

6.8.2 Command w

Command w set the sensor sample time expressed in seconds.

If the w is sent alone the mote replies with usage explanation.

w

26/02/2014 13.30.06;987; Use: (i.e.) w=20 (sleep for 20 sec instead 60)

In order to set the desired sample time the command has to be w=30 to set 30 second sample interval w=30

26/02/2014 13.35.00;055; *** Setting sleep to 30 sec

6.8.3 Command q

Command q set the local sensor on off state setting, therefore when the local setting is on the sensors on/off state can be different respect the sensors on/off state imposed by the network.

The q command is a switch command; its execution toggles its state on/off and vice versa.

q

26/02/2014 13.40.15;808; Set the configuration of the sensors' state locally=YES

q

26/02/2014 13.41.01;352; Set the configuration of the sensors' state locally=NO

6.8.4 Command p

Command p sets the sensors state on/off. If it is executed with local sensor state on (command q) sets the sensors state of the mote, otherwise set the sensors state of the whole network.

If the p is sent alone the mote replies with usage explanation.

P

26/02/2014 14.38.06;982; Use: binary (1/0) to enable disable:

Static_GAS

T_ext_sensor

Current_sensor

Temperature

Humidity

Light

Dynamic_GAS

Accelerometer

In order to set the desired sensors state the command has to be a 01 string where 0 is off and 1 is on, with new state position in accordance with the sensors list plot from the usage explanation.

p=00110101

26/02/2014 14.48.53;539; *** Setting sensors state to 00110101

26/02/2014 14.48.53;540; Static_GAS: OFF

26/02/2014 14.48.53;540; T_ext_sensor: OFF

26/02/2014 14.48.53;541; Current_sensor: ON

26/02/2014 14.48.53;542; Temperature: ON

26/02/2014 14.48.53;542; Humidity: OFF

26/02/2014 14.48.53;543; Light: ON

26/02/2014 14.48.53;543; Dynamic_GAS: OFF

26/02/2014 14.48.53;544; Accelerometer: ON

6.8.5 Command i

Command i set the mote identifier number ID value.

If the i is sent alone the mote replies with usage explanation.

i

26/02/2014 14.52.16;738; Use: (i.e.) i=9 (it selects node ID 9 instead 1)

In order to set the desired ID value the command has to be i=5 to set the ID to 5.

i=5

26/02/2014 14.53.37;775; *** Node ID set to 5

6.8.6 Command l

Command l set the mote time slot value.

If the l is sent alone the mote replies with usage explanation.

l

26/02/2014 14.52.16;738; Use: (i.e.) i=9 (it selects node ID 9 instead 1)

In order to set the desired time slot value the command has to be l=4 to set the time slot to 4.

l=4

26/02/2014 14.57.23;216; *** Time slot set to 4

6.8.7 Command r

Command r reset the system. When the command is performed the system reset instantaneously r

26/02/2014 14.58.07;144; System reset

6.8.8 Command o

Command o set the internal mote clock.

If the o is sent alone the mote replies with usage explanation.

o

26/02/2014 14.59.07;871; Use: (i.e.) o=2011:05:16-15:32:54 (May 16 2011 3p.m. 32min 54sec)

In order to set the desired internal clock date and time the command has to be performed like o=year:month:day-huor:minute:second.

o=2014:02:26-15:18:21

26/02/2014 15.03.32;094; Date setted to 2014:02:26-15:18:21

6.8.9 Command s

Command s set the mote to start or stop the sensors sample if in stand alone mode. Otherwise set the whole network to start or stop the sensor sample.

The s command is a switch command; its execution toggles its state on/off and vice versa.

With network enable and executed to the coordinator

s

26/02/2014 17.21.16;738; Set the network node to sleep=YES

s

26/02/2014 17.21.34;770; Set the network node to sleep=NO

With network enable and executed to the sleepyEnd device

s

26/02/2014 17.22.51;617; Network enabled, not possible to sleep

With network disable.

s

26/02/2014 17.24.25;865; Sleep state=ENABLE

s

26/02/2014 17.26.54;072; Sleep state=DISABLE

6.8.10 Command a

Command a forces the mote to associate its radio link to another node. If the association will succeed the mote will have a shot address assignment from the selected parent.

If the a is sent alone the mote replies with usage explanation.

a

01/01/2011 00.00.10;895; Use: (i.e.) a=0xAA to associate this node (set node parent short address to AA)

In order to force the mote to associate with a desired parent the command has to be a=parent_address.

a=0x0

01/01/2011 00.07.33;460; assagned network short address=0x0001; parent address 0x0000; aptended parent 0x0000

If the command succeeds the mote will have a short address assignment, otherwise nothing change.

a=0x3

01/01/2011 00.10.27;456; no node association

6.8.11 Command ns

The command ns forces the mote to execute a network scan that will list which motes are visible with which radio strength.

ns

01/01/2011 00.17.31;274; There radio scan has seen the follow:

01/01/2011 00.17.31;375; node 1/2 panID=0xeeee, shortAddress=0x0000, LQ=-44dbm

01/01/2011 00.17.31;376; node 2/2 panID=0xeeff, shortAddress=0x0000, LQ=-52dbm

6.8.12 Command n

Command n will start or stop the mote network mode.

The n command is a switch command; its execution toggles its state on/off and vice versa.

The n command has different behaviour if it is performed in the coordinator or in the SleepEnd device.

In the coordinator the n command set a new network opening an interacting section where the user can select all the network parameters.

n

26/02/2014 17.38.09;361; Network state=ENABLE

26/02/2014 17.38.09;362; Default PAN ID=0x0

26/02/2014 17.38.09;363; Would you like to change PAN ID? (y/n)=y

26/02/2014 17.38.14;209; CHECK_PAN_ID

26/02/2014 17.38.14;210; new PAN ID=0x3a

26/02/2014 17.38.17;489; saved PAN ID=0x3a

26/02/2014 17.38.17;490; Default Radio CHANNEL=13

26/02/2014 17.38.17;491; Would you like to change CHANNEL? (y/n)=y

26/02/2014 17.38.20;321; CHECK_CHANNEL

26/02/2014 17.38.20;322; new CHANNEL (from 11 to 26)=11

26/02/2014 17.38.21;690; saved CHANNEL=11

26/02/2014 17.38.21;690; Default MAX_CHILDREN per node=8

26/02/2014 17.38.21;691; Would you like to change MAX_CHILDREN? (y/n)=y

26/02/2014 17.38.24;529; CHECK_MAX_CHILDREN

26/02/2014 17.38.24;530; new MAX_CHILDREN=6

26/02/2014 17.38.27;817; saved MAX_CHILDREN=6

If n is sent again in the coordinator the coordinator will not switch off the network to avoid accidental network workout.

n

26/02/2014 17.41.53;278; Network state=ENABLE (the network cannot be disabled in the coordinator!)

In the Sleepend the n command set the mote to join an existing network opening an interacting section where the user can select all the network parameters to join if the network was not previously set.

n

01/01/2011 00.13.37;068; Network state=ENABLE

01/01/2011 00.13.37;069; Choose a node to try to connect, node short address (press 'p' to print the result, 'n' to repeat the scan or 'x' to exit from the procedure)

01/01/2011 00.13.39;162; Default PAN ID=0xeeee

01/01/2011 00.13.39;163; Would you like to change PAN ID? (y/n)=y

01/01/2011 00.13.43;492; new PAN ID=0x3a

01/01/2011 00.13.47;084; saved PAN ID=0x3a

01/01/2011 00.13.47;085; Default Radio CHANNEL=11

01/01/2011 00.13.47;086; Would you like to change CHANNEL? (y/n)=y

01/01/2011 00.13.50;540; new CHANNEL (from 11 to 26)=11

01/01/2011 00.13.51;540; saved CHANNEL=11

01/01/2011 00.13.51;541; Default MAX_CHILDREN per node=8

01/01/2011 00.13.51;542; Would you like to change MAX_CHILDREN? (y/n)=y

01/01/2011 00.13.54;052; new MAX_CHILDREN=6

01/01/2011 00.13.54;556; saved MAX_CHILDREN=6

01/01/2011 00.13.55;556; There radio scan has seen the follow:

01/01/2011 00.13.55;657; node 1/2 panID=0x003a, shortAddress=0x0000, LQ=-47dbm

01/01/2011 00.13.55;658; node 2/2 panID=0xeeff, shortAddress=0x0000, LQ=-52dbm

01/01/2011 00.13.55;659; Choose a node to try to connect, node short address (press 'p' to print the result, 'n' to repeat the scan or 'x' to exit from the procedure)=0x0

01/01/2011 00.14.10;468; assagned network short address=0x0001; parent address 0x0000; aptended parent 0x0000

If the command n is selected again the SleepyEnd mote leaves the network.

n

01/01/2011 00.17.26;083; Network state=DISABLE

If the command n is selected again the SleepEnd rejoin the previous network providing a new network scan to select a new parent.

n

01/01/2011 00.17.27;179; Network state=ENABLE

01/01/2011 00.17.27;180; Choose a node to try to connect, node short address (press 'p' to print the result, 'n' to repeat the scan or 'x' to exit from the procedure)

01/01/2011 00.17.30;272; There radio scan has seen the follow:

01/01/2011 00.17.30;373; node 1/3 panID=0xeeff, shortAddress=0x0000, LQ=-55dbm

01/01/2011 00.17.30;374; node 2/3 panID=0xeeff, shortAddress=0x000b, LQ=-89dbm

01/01/2011 00.17.30;375; node 3/3 panID=0x003a, shortAddress=0x0000, LQ=-46dbm

01/01/2011 00.17.30;376; Choose a node to try to connect, node short address (press 'p' to print the result, 'n' to repeat the scan or 'x' to exit from the procedure)=0x0

01/01/2011 00.17.35;352; assagned network short address=0x0001; parent address 0x0000; aptended parent 0x0000

6.8.13 Command m

Command m performs a set of remote settings and network settings.

Performing the command m the mote displays a help sub menu with instruction explanation and examples.

m

01/01/2011 00.21.58;858; Use: (i.e.) m=0x1a (to connect remotely to node with address 0x1a)

01/01/2011 00.21.58;859; Use: (i.e.) mx=0x1;2;3;0x3;0x1 (to remotely set the node with short address=0x1 with ID=2, time slot=3, short address=0x3, parent address=0x1)

01/01/2011 00.21.58;861; Use: (i.e.) my=1;2;3;0x3;0x1 (to remotely set the node with ID=1 with ID=2, time slot=3, short address=0x3, parent address=0x1)

01/01/2011 00.21.58;862; Use: (i.e.) mt=1;1;0x1 (to start time slot assignment procedure from ID=1 with time slot=1, short address=0x1)

01/01/2011 00.21.58;864; Use: (i.e.) mr to restart all the nodes

01/01/2011 00.21.58;865; Use: (i.e.) ms=0xa reset the child with address 0xa and all its children short address, parent address and time_slot

01/01/2011 00.21.58;866; Use: (i.e.) mf to reset all the nodes flash and restart the network

01/01/2011 00.21.58;867; Use: (i.e.) ma=-80 to autoconfigure new node address with radio strength higher than -80dbm

Command m=short_address permits to remotely access to the terminal interface of a remote mote.

m=0x1

26/02/2014 17.48.32;878; connected remotely to node address 0x1

remote=0x0001>26/02/2014 17.48.32;971; !client remote mode enabled!

remote=0x0001>

Now it is possible to execute commands in the remote mote, like network scan (n, ns, a commands) and sensors setting (p, q commands). The entire command sent is available.

Performing m command again will close the remote terminal connection.

remote=0x0001>m

remote=0x0001>26/02/2014 17.53.50;175; !client remote mode disabled!

The command mr will restart all the nodes present in the network.

mr

26/02/2014 17.59.38;673; reset all the nodes

The command ms=short_address reset the parenthood of the short_address mote and all its children to permit a new network auto setting and configuration. With short_address equal 0x0 all the nodes in the network will reset their parenthood keeping the network parameter like radio channel, panID and max children intact.

ms=0x0

01/01/2011 00.21.46;906; reset the child with address 0x0 and all its children short address, parent address and time_slot

The command mf reset to the default setting all the motes, except their ID. After the execution of this command all the nodes will be at the configuration stage waiting to be configured locally or remotely.

mf

01/01/2011 00.00.10;883; reset all the nodes flash and restart the system

The command ma=minimum_link_quality automatically configure the network to construct a tree topology with link strength higher than minimum_link_quality.

ma=-96

26/02/2014 19.15.38;252; start autoconfig procedure with radio strength higher than -96dbm

26/02/2014 19.15.38;314; Short address node 0x0 searching children

26/02/2014 19.15.46;928; DATA NODE ID;0006;0001;00000;0001;0000;-074;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.15.47;046; DATA NODE ID;0007;0001;00000;0002;0000;-079;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.17.09;513; Short address node 0x1 searching children

26/02/2014 19.17.18;100; DATA NODE ID;0004;0001;00000;0009;0001;-078;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.17.19;634; Short address node 0x2 searching children

26/02/2014 19.17.28;821; DATA NODE ID;0005;0001;00000;0012;0002;-083;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.19.01;440; Short address node 0x12 searching children

26/02/2014 19.19.11;586; DATA NODE ID;0008;0001;00000;0011;0002;-092;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.20.34;169; Short address node 0x11 searching children

The command mt=ID;time_slot;short_address assigns time slot equal time_slot to the mote ID and short address short_address and start the automatic procedure to assign time slot the rest of the network with higher short address.

mt=6;1;0x1

26/02/2014 19.24.05;042; start time slot assignment procedure

26/02/2014 19.24.05;043; send to node ID=6 new_time_slot=1 address=0x1

26/02/2014 19.24.05;105; DATA NODE ID;0006;0001;00000;0001;0000;-098;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.24.05;373; DATA NODE ID;0004;0002;00000;0009;0001;-098;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.24.06;535; DATA NODE ID;0007;0004;00000;0002;0000;-098;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.24.06;803; DATA NODE ID;0008;0005;00000;0011;0002;-098;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

26/02/2014 19.24.07;433; DATA NODE ID;0005;0007;00000;0012;0002;-098;0000;0000;00000;0000;0000;000000;000000;0000;0000;0000;0000;0000;0000;0000;00000000;00000000;

The command mx=short_address;new_ID;new_timeslot;new_short_address;new_parent_address force the mote with short address short_address to set its ID to new_ID, time_slot to new_time_slot, short address to new_short_address and parent address to new_parent_address.

mx=0x1;1;1;0x2;0x0

01/01/2011 00.26.00;992; change remote node network setting

01/01/2011 00.26.00;993; send to node address=0x1 new_id=1 new_time_slot=1 new_address=0x2 new_parent_address=0x0

01/01/2011 00.26.01;331; Node S.A.=0x1 new 1;1;0x2;0x0

The last print is the reply of the remote node that has executed the command.

The command my=ID;new_ID;new_timeslot;new_short_address;new_parent_address force the mote with ID ID to set its ID to new_ID, time_slot to new_time_slot, short address to new_short_address and parent address to new_parent_address.

my=1;2;1;0x1;0x0

01/01/2011 00.26.43;861; change remote node network setting

01/01/2011 00.26.43;862; send to node ID=1 new_id=2 new_time_slot=1 new_address=0x1 new_parent_address=0x0

01/01/2011 00.26.44;200; Node ID=1 new 2;1;0x1;0x0

The last print is the reply of the remote node that has executed the command.

6.8.14 Command k

The k command deletes the no volatile mote memory of the mote without restart the system. This command will be useful if the end user would like to bring back the system to the default setting. Performing a reset the mote will go back to the default setting.

k

27/02/2014 00.07.26;925; Flash reset

6.8.15 Command x

Command x force the node to have the user defined network setting.

The execution of command x is done with an interactive session where the user can configure ID, pan ID, radio channel, short address, parent address, max children.

Moreover the command can propagate the new pan ID radio channel and max children number to all the network motes.

x

27/02/2014 00.11.25;514; Default PAN ID=0xeeee

27/02/2014 00.11.25;515; Would you like to change PAN ID? (y/n)=y

27/02/2014 00.11.30;594; new PAN ID=0x2a

27/02/2014 00.11.41;090; saved PAN ID=0x2a

27/02/2014 00.11.41;091; Default Radio CHANNEL=11

27/02/2014 00.11.41;091; Would you like to change CHANNEL? (y/n)=y

27/02/2014 00.11.44;506; new CHANNEL (from 11 to 26)=12

27/02/2014 00.11.46;786; saved CHANNEL=12

27/02/2014 00.11.46;787; Default ADDRESS=0x1

27/02/2014 00.11.46;787; Would you like to change ADDRESS? (y/n)=y

27/02/2014 00.11.52;834; new ADDRESS=0x2

27/02/2014 00.11.56;242; saved ADDRESS=0x2

27/02/2014 00.11.56;243; Default address of PARENT=0x0

27/02/2014 00.11.56;243; Would you like to change PARENT? (y/n)=y

27/02/2014 00.12.00;130; new PARENT=0x0

27/02/2014 00.12.06;642; saved PARENT=0x0

27/02/2014 00.12.06;643; Default MAX_CHILDREN per node=8

27/02/2014 00.12.06;643; Would you like to change MAX_CHILDREN? (y/n)=y

27/02/2014 00.12.13;234; new MAX_CHILDREN=10

27/02/2014 00.12.16;546; saved MAX_CHILDREN=10

27/02/2014 00.12.16;547; Would you like to restart the setting procedure?=n

27/02/2014 00.12.22;898; Would you like to change PAN ID, CHANNEL and MAX_CHILDREN to the entire network? (y/n)=y

27/02/2014 00.12.30;986; YES

net channel=12 pan id=0x2a max children=10

6.8.16 Command t

Command t displays all the setting of the motes, from sensors setting to network setting.

t

27/02/2014 00.35.12;118; ===== DATE =====

27/02/2014 00.35.12;119; 27/02/2014 00.35.12

27/02/2014 00.35.12;120; ===== Sleep time and sensor state =====

27/02/2014 00.35.12;121; Node ID 0

27/02/2014 00.35.12;121; Network enable: YES

27/02/2014 00.35.12;122; Sleep time: 0min 20sec

27/02/2014 00.35.12;123; sensors setting local: OFF

27/02/2014 00.35.12;124; Static_GAS: OFF

27/02/2014 00.35.12;125; T_ext_sensor: ON

27/02/2014 00.35.12;126; Current_sensor: OFF

27/02/2014 00.35.12;126; Temperature: ON

27/02/2014 00.35.12;127; Humidity: ON

27/02/2014 00.35.12;128; Light: ON

27/02/2014 00.35.12;129; Dynamic_GAS: OFF

27/02/2014 00.35.12;129; Accelerometer: OFF

27/02/2014 00.35.12;130; Vheater GAS sensor 2.40V

27/02/2014 00.35.12;131; ===== Network parameter =====

27/02/2014 00.35.12;132; short address: 0x0

27/02/2014 00.35.12;133; Radio Channel: 11

27/02/2014 00.35.12;134; Network PAN ID: 0x2b

27/02/2014 00.35.12;134; Max number of children: 8

27/02/2014 00.35.12;135; Sleep enable: NO

27/02/2014 00.35.12;136; Next data sample in 131078sec

27/02/2014 00.35.12;137; ===== Children info =====

27/02/2014 00.35.12;138; child; Position; 0; ID; 2; ADDRESS; 0001; MAC; 0x00158d000028463c

27/02/2014 00.35.12;140; child; Position; 1; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.35.12;141; child; Position; 2; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.35.12;143; child; Position; 3; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.35.12;144; child; Position; 4; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.35.12;145; child; Position; 5; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.35.12;147; child; Position; 6; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.35.12;148; child; Position; 7; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

If the network is disabled the print of the network parameter is omitted.

t

27/02/2014 00.37.17;409; ===== DATE =====

27/02/2014 00.37.17;410; 27/02/2014 00.37.17

27/02/2014 00.37.17;411; ===== Sleep time and sensor state =====

27/02/2014 00.37.17;411; Node ID 2

27/02/2014 00.37.17;412; Network enable: NO

27/02/2014 00.37.17;412; Sleep time: 0min 20sec

27/02/2014 00.37.17;413; sensors setting local: OFF

27/02/2014 00.37.17;414; Static_GAS: OFF

27/02/2014 00.37.17;414; T_ext_sensor: ON

27/02/2014 00.37.17;415; Current_sensor: OFF

27/02/2014 00.37.17;415; Temperature: ON

27/02/2014 00.37.17;416; Humidity: ON

27/02/2014 00.37.17;417; Light: ON

27/02/2014 00.37.17;417; Dynamic_GAS: OFF

27/02/2014 00.37.18;418; Accelerometer: OFF

27/02/2014 00.37.17;418; Vheater GAS sensor 2.40V

27/02/2014 00.37.17;419; Sleep enable: NO

6.8.17 Command u

Command u displays all the setting of the motes, from sensors setting to network setting with a more easiness parsing format.

u

27/02/2014 00.39.41;303; NODE INFO;0;27/02/2014 00.39.41;2;0;20;0;1;0;1;1;1;0;0;2.40;0;0;11;43;131078;8;

Where the fields are: ID;DAY;MONTH;YEAR HOUR.MINUTE.SECOND;NETWORK ENABLE; SLEEP MINUTES;SLEEP SECONDS;STATIC GAS SENSORS STATE; EXTERNAL TEMPERATURE SENSOR STATE; CURRENT SENSOR STATE; TEPERATURE SENSOR STATE; HUMIDITY SENSOR STATE; LIGHT SENSOR STATE; DYNAMIC GAS SENSOR STATE; ACCELEROMETER SENSOR STATE; GAS HEATER VOLTAGE; SLEEP ENABLE STATE;SHOORT ADDRESS; RADIO CHANNEL; NET PAN ID; MAX CHILDREN;

6.8.18 Command v

Command v set the heater voltage of the gas sensor.

If the v is sent alone the mote replies with usage explanation.

v

27/02/2014 00.54.55;092; Set Vheater GAS sensor (Use: v=270 for (Vheater=2.7V))

In order to set the desired gas sensor heater voltage the command has to be performed like v=251 to set the voltage to 2.51V.

v=251

27/02/2014 00.58.12;091; Vheater=2.51

6.8.19 Command c

The command c deletes a child in the children list.

If the c is sent alone the mote replies with usage explanation and children information.

c

27/02/2014 00.59.45;664; Use: (i.e.) c=2 (clean child position 2)

27/02/2014 00.59.45;665; ===== Children info =====

27/02/2014 00.59.45;665; child; Position; 0; ID; 2; ADDRESS; 0001; MAC; 0x00158d000028463c

27/02/2014 00.59.45;667; child; Position; 1; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.59.45;669; child; Position; 2; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.59.45;670; child; Position; 3; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.59.45;672; child; Position; 4; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.59.45;673; child; Position; 5; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.59.45;674; child; Position; 6; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 00.59.45;676; child; Position; 7; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

To delete a child a command has to be c=child_position to delete the child in the position child_position.

c=0

27/02/2014 01.03.30;848; Delete child in position 0

27/02/2014 01.03.30;848; ===== Children info =====

27/02/2014 01.03.30;849; child; Position; 0; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 01.03.30;851; child; Position; 1; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 01.03.30;852; child; Position; 2; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 01.03.30;854; child; Position; 3; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 01.03.30;855; child; Position; 4; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 01.03.30;856; child; Position; 5; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 01.03.30;858; child; Position; 6; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

27/02/2014 01.03.30;859; child; Position; 7; ID; 0; ADDRESS; 0000; MAC; 0x0000000000000000

6.8.20 Command format

Command format formats the FAT file system of the micro SD card.

6.8.21 Command fp

Command fp prints the text inside a file in the microSD card.

If the fp is sent alone the mote replies with usage explanation.

fp

27/02/2014 02.57.52;621; Use: (i.e.) fp=try.txt to print try.txt file content

To print the file content the command has to be fp=file_name.

p=data0001.csv

27/02/2014 02.52.49;656; file: data0001.csv opened

DATE;ID;Packet num;Time slot;Short Address;Parent Address;Link Quality;Temperature;Humidity;Light;Ext_Temperature;Gas_sensor;CS_current;CS_power;AccX;AccY;AccZ;PosX;PosY;PosZ;Battery Voltage;charge;Debug

01/01/2011 00.04.16;0001;0001;00001;0001;0000;-036;2257;5661;00122;-2282;0000;000000;000000;0000;0000;0000;0000;0000;0000;2595;00000200;000000ef

27/02/2014 02.32.00;0001;0001;00002;0001;0000;-036;2256;5655;00128;-2280;0000;000000;000000;0000;0000;0000;0000;0000;0000;2598;00475250;000000e1

27/02/2014 02.32.20;0001;0001;00003;0001;0000;-039;2263;5689;00121;-2285;0000;000000;000000;0000;0000;0000;0000;0000;0000;2598;00523515;000000e1

27/02/2014 02.32.40;0001;0001;00004;0001;0000;-040;2269;5675;00122;-2280;0000;000000;000000;0000;0000;0000;0000;0000;0000;2598;00551197;000000e1

27/02/2014 02.33.00;0001;0001;00005;0001;0000;-040;2274;5640;00124;-2286;0000;000000;000000;0000;0000;0000;0000;0000;0000;2598;00590342;000000e1

6.8.22 Command fl

Command fl list the content of a microSD card directory.

If the fl is sent alone the mote replies with usage explanation.

fl

27/02/2014 03.07.15;746; Use: (i.e.) fl=nome_dir (use fl= to list the root directory)

To print the directory content the command has to be fl=directory or fl= to print the root directory.

fl=

27/02/2014 03.08.49;779; directory opened

27/02/2014 03.08.49;803; /log.txt 8503 byte

27/02/2014 03.08.49;804; /data0001.csv 7067 byte

6.8.23 Command fd

Command fd delete a file or directory in the microSD card.

If the fd is sent alone the mote replies with usage explanation.

fd

27/02/2014 03.10.12;265; Use: (i.e.) fd=try.txt to delete try.txt file

To delete a file or directory the command has to be fd=file_name or fd=directory_name.

fd=log.txt

27/02/2014 03.12.52;789; file: log.txt deleted

6.8.24 Command fm

Command fm move a microSD file from directories. The command can be used even to rename files.

If the fm is sent alone the mote replies with usage explanation.

fm

27/02/2014 03.16.06;463; Use: (i.e.) fm=try.txt=ops.txt to move try.txt to ops.txt

To move a file the command has to be fm=directory_name_1/file_name =directory_name_2/file_name.

fm=data0001.csv=backup/data0001.csv

27/02/2014 03.22.28;416; moved data0001.csv to backup/data0001.csv

To rename a file the command has to be fm=file_name =new_file_name.

fm=backup/data0001.csv=backup/0001.csv

27/02/2014 03.24.11;820; moved backup/data0001.csv to backup/0001.csv

6.8.25 Command ff

The command ff makes a directory in the sd card.

If the ff is sent alone the mote replies with usage explanation.

ff

27/02/2014 03.26.41;947; Use: (i.e.) ff=try to create try folder

To create a directory the command has to be ff=directory_name.

ff=backup

27/02/2014 03.21.27;359; directory: backup created

6.8.26 Command cntrl+a

Command cntrl+a start the procedure to upload the Coordinator firmware. This command is managed by the JavaTerminal.

6.8.27 Command cntrl+b

Command cntrl+b start the procedure to upload the SleepEnd firmware. This command is managed by the JavaTerminal. If the command is executed in the Coordinator the procedure starts the upload over the air to update all the SleepyEnd motes in the network.

7 Radio protocol Design

A WSN designed for building energy efficiency has to:

- Be fully configurable with multi-hop radio link.
- Work for several years with two commercial AA batteries.
- Provide all the network information.
- Be Flexible and extensible.
- Be easy installable.
- Has low maintenance.

To do that:

- Each device has to give to the installer the possibility to:
  - Check the list of neighbor device with radio link quality.
  - Select the network parent device in order to build up a network custom tree.
  - Set the device ID.
  - Check sensor state (disable/enable), sample time and batteries voltage.
- The network coordinator has to give to the installer the possibility to:
  - Set each node sensor state (enable/disable)
  - Set sample time
  - Collect data from every device connected to the network
  - Set date
  - Automatically configure the network
  - Manually configure the network
  - Upload the firmware over the air

The design of the 3ENCULT radio stack has follow the above points. The structure of the network is a tree frame as in the below figure.

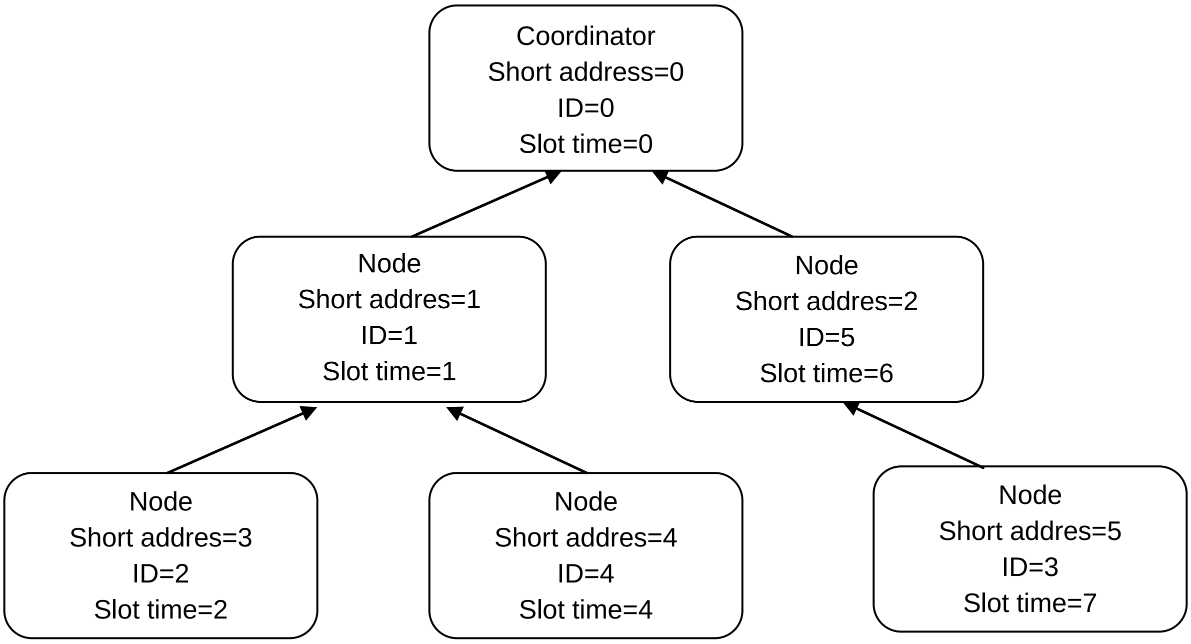

Fig. 7.1 WSN tree frame structure

The ID is the identifier number that will correlate the data with the physical position of the sensors. The user can select which ID to assign to a WSN node, which will send its data marked by the assigned identifier.

The short address it is the network address of each node. It is assigned automatically by the radio protocol, however the user can configure this parameter manually to force the network to work as he she wish.

During the installation time the user can execute a network scan, where the new node analyze the presence of other nodes in the network and provide to the user which is the radio signal strength with the nodes in the radio range. This action permits the user to understand which node already in the network would be good to be the network parent of the new node. The user at the end can set the new node to join the network with the parent node he she prefer.

Moreover the user can force the node to change the parent node in the way to build up the wished network.

The parent node it is the short address of the node that the new one will send its data, so looking to the image below the node with ID=2 has short address 3, it will send data to node with ID=1 and short address 1 which is the parent of node with ID=2. The data then will be retransmitted by the node ID=1 to the node ID=0, which it is the coordinator of the network and which it is connected to a pc, tablet or smartphone that will store the data and let them available to internet.

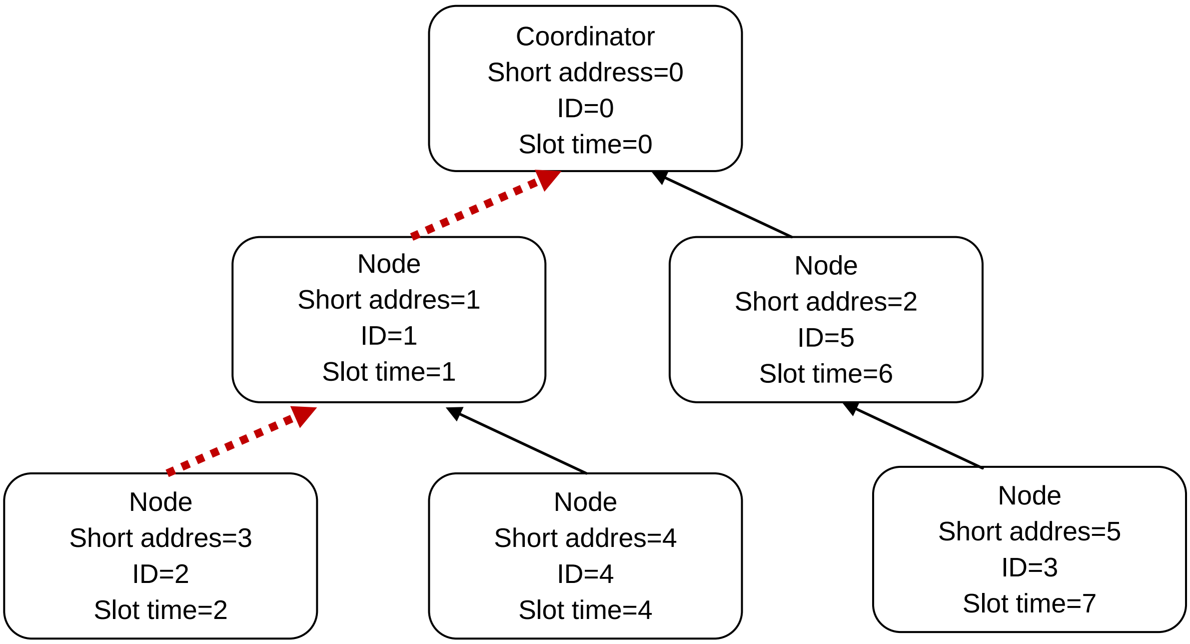

Fig. 7.2 WSN tree frame with transmission data example

To avoid data collision between many data transmission each node has its assigned time slot. During a time slot the data will be sent from a node through its parent until will reach the coordinator. Looking to the previous image the node ID=2 will send the data at time slot 2 and the node ID=1 will receive them. During time slot 3 the node ID=1 will retransmit the data of the node ID=2 to the node ID=0 which will provide them to the data storage system. Thereafter the node ID=4 can use the time slot 4 to transmit the data, which will reach the coordinator in the time slot 5.

The time slot can be set from the user or from the automatic procedure with command mt or using the JavaTerminal application. If the user sets manually the time slot, he/she has to ensure the right interval between transmission in the way to avoid data collision. However the radio protocol has the capability to avoid data collision even if more the one node will transmit data in the same time slot. The radio protocol in fact will transmit a data only if the radio channel is empty. The node before transmit listens the channel, and if it is not empty, it will wait until the channel is free.

The coordinator can communicate with the other nodes using a broadcast transmission. The broadcast permits to reach all the nodes in the network even if nodes have not assigned short address and ID. In fact the broadcast it is a transmission delivered to every nodes in the network, and all the nodes will repeat the transmission to let the packet reach everyone. The repeated transmission it is done simultaneously with a strong synchronization to have a constructive interference.

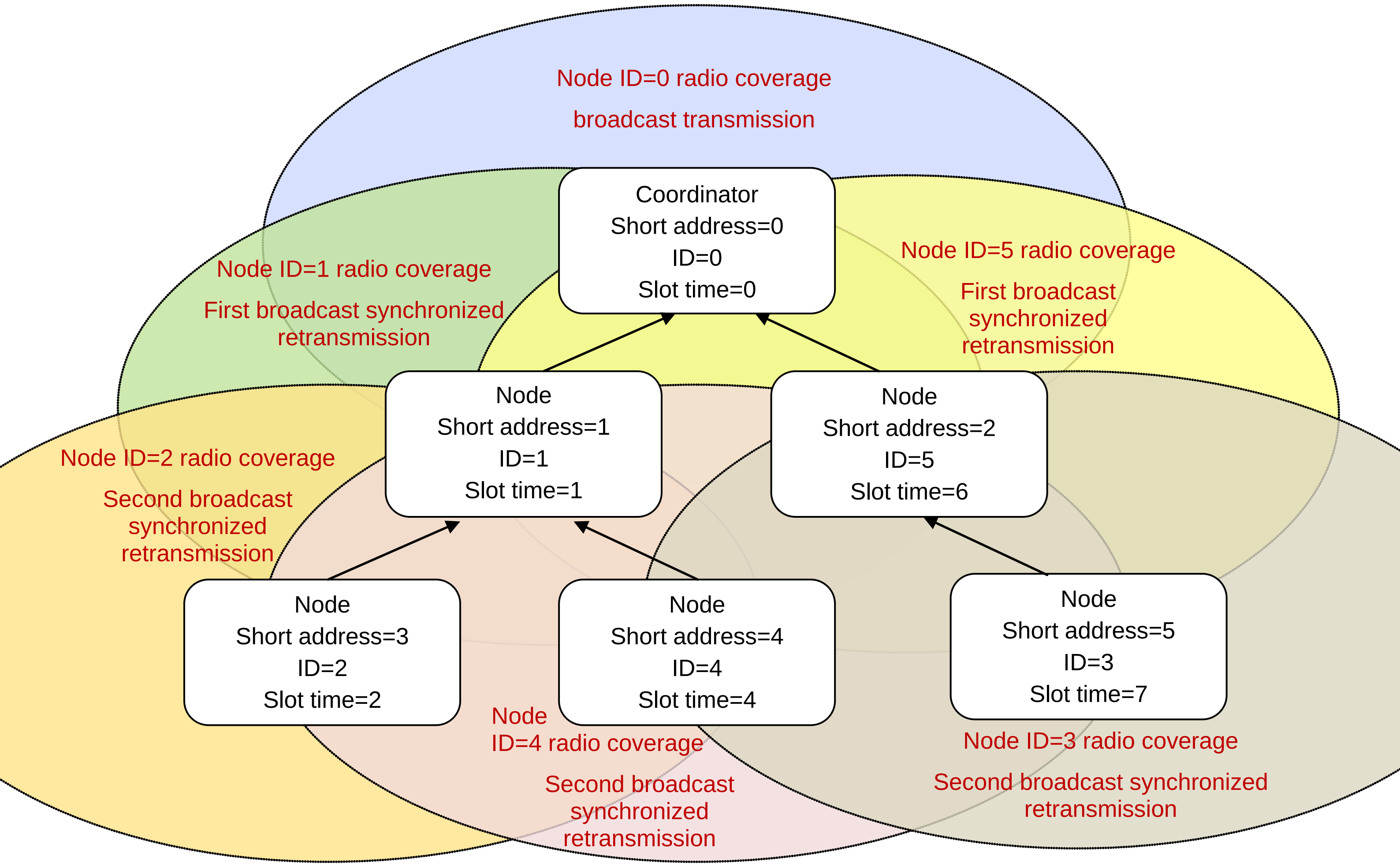

Fig. 7.3 WSN tree frame with broadcast transmission example, where every node retransmit the packet

This transmission solution it is highly reliable and permits to reach every part of the network without configuration. However it is energy efficient only to transmit information to the whole network. Then this kind of transmission it is used by the coordinator to communicate to the whole network which sensors have to sample, which is the sample rate and to synchronize the network in the way that each node can respect its own time slot.

The broadcast signal it is widely used for installation purpose from both coordinator and SleepyEnd devices. In fact it is used to automatically configure the network, to set or force a node setting, to remote control mote command interface and to perform over the air firmware update.

8 User interface: JavaTerminal

The users can interact with the 3EncultWSN using the java application “javaTerminal” able to run in several commercial system like, pc, pc board, smartphone and tablet called afterword “gateway”.

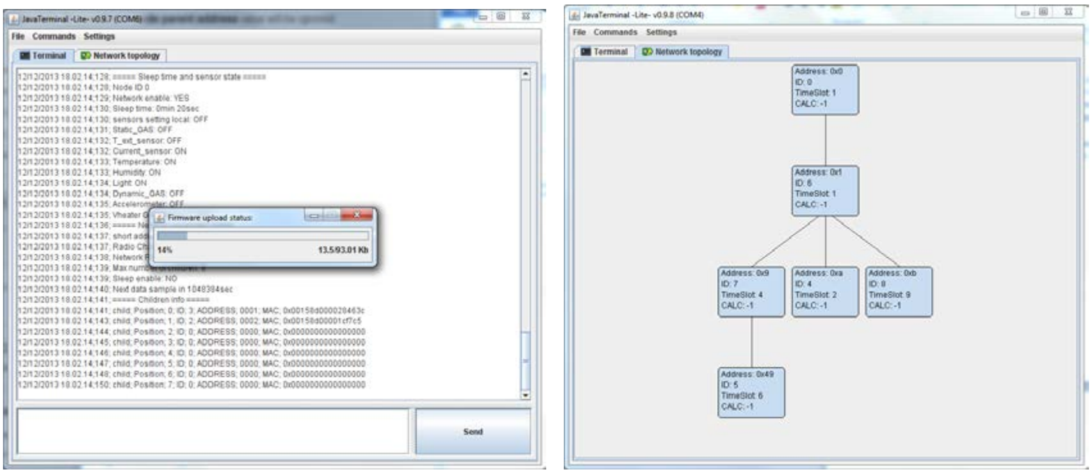

Fig. 8.1 JavaTerminal application

With the javaTerminal application the user can: Configure a new network; collect and visualize the data; remote control each mote in the network; set the sensors enables; select the time acquisition interval; visualize the network structure; update the system software; send the data to a remote server for internet base application and building management application.

8.1 Firmware update

To update a node connected with the cable to a gateway, the user has to check the firmware type with command h.

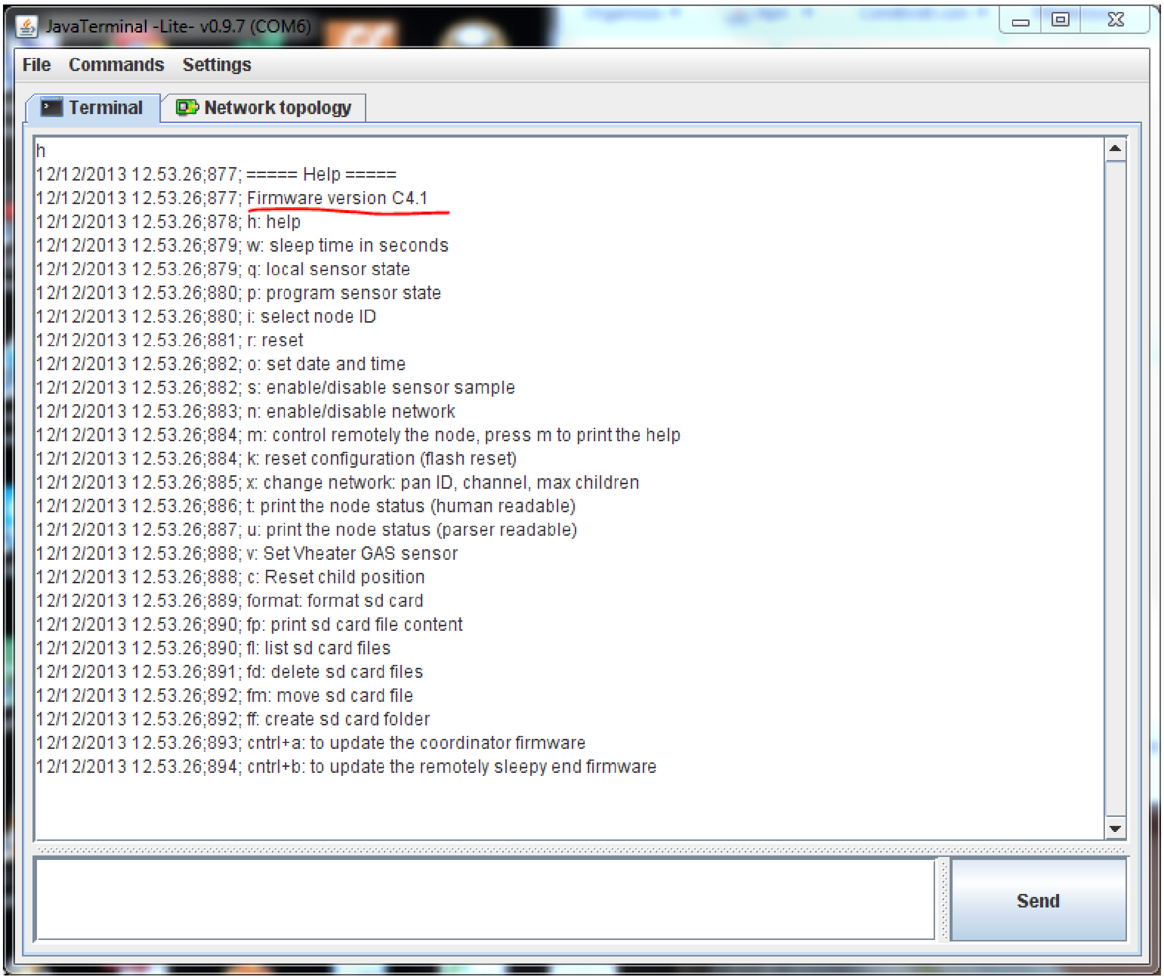

Fig. 8.2 Firmware version check

If the firmware version is Cx.x the node is a coordinator, otherwise it is a SleepyEnd router.

Now the user has to select Commands Upgrade Coordinator

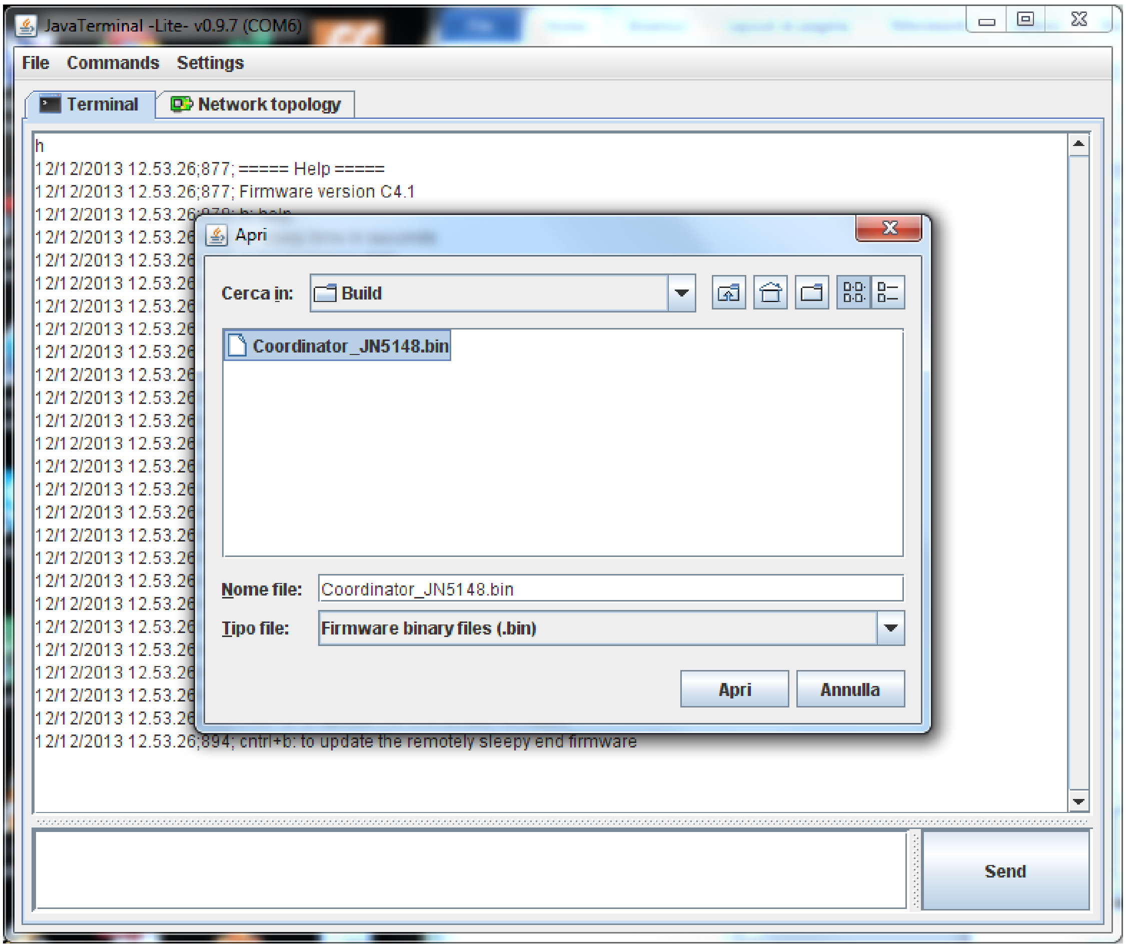

Fig. 8.3 Firmware binary file selection

The user has to select the right Coordinator_JN5148.bin file with the desired firmware version and press Open.

The new firmware will be written in a few minutes.

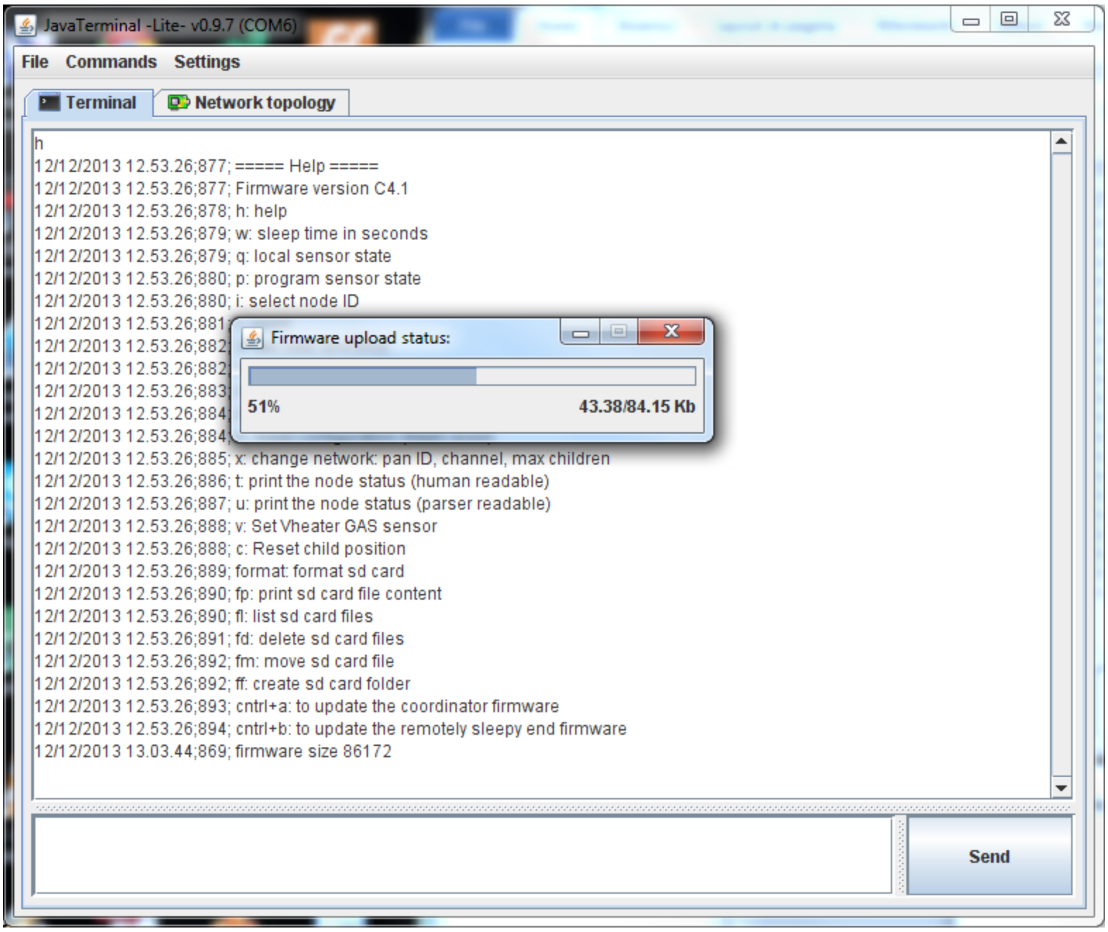

Fig. 8.4 Firmware upload

8.2 Over the air firmware update

Before to perform firmware update over the air, the user has to collect the network information from the java terminal connected to the coordinator.

Most important information are: Every node firmware version, radio channel and network pan ID.

Every node firmware version can be collected from the incoming data.

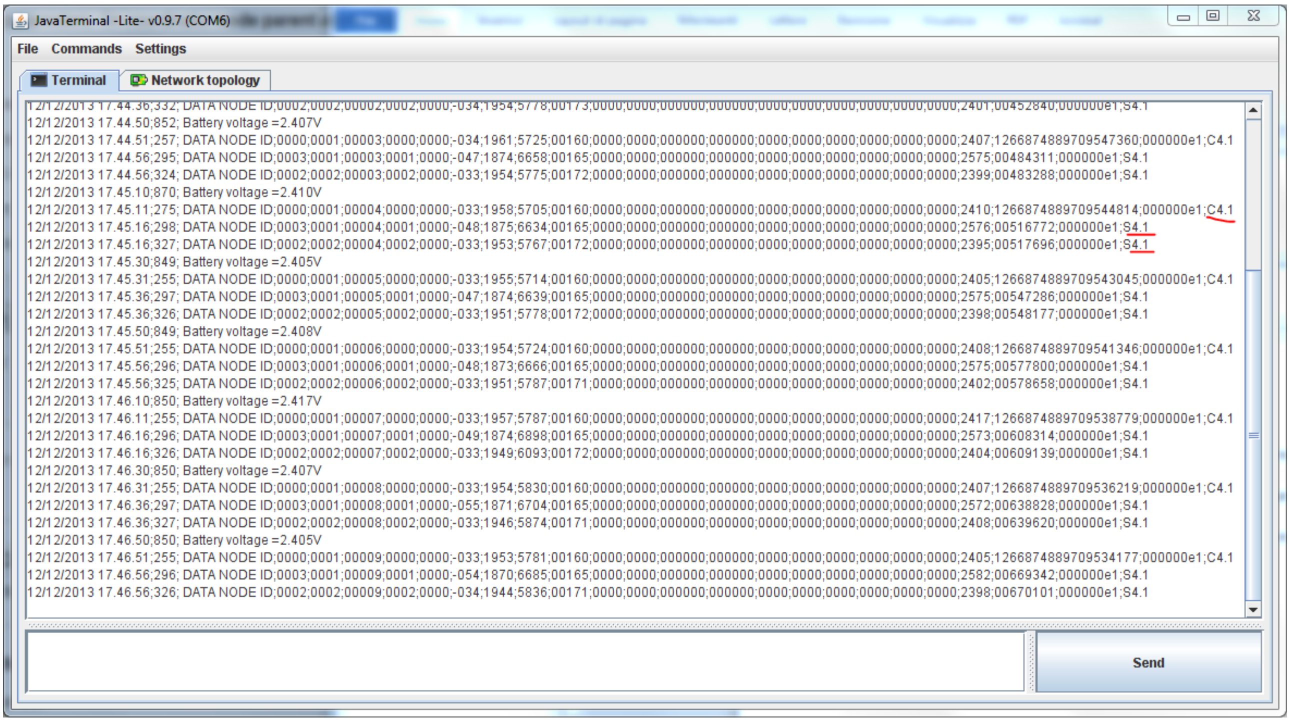

Fig. 8.5 Firmware version plot of each mote in the network

In the example all the sleepy end nodes have the firmware S4.1 and the coordinator has C4.1.

Then the user has to stop the network sampling with command s.

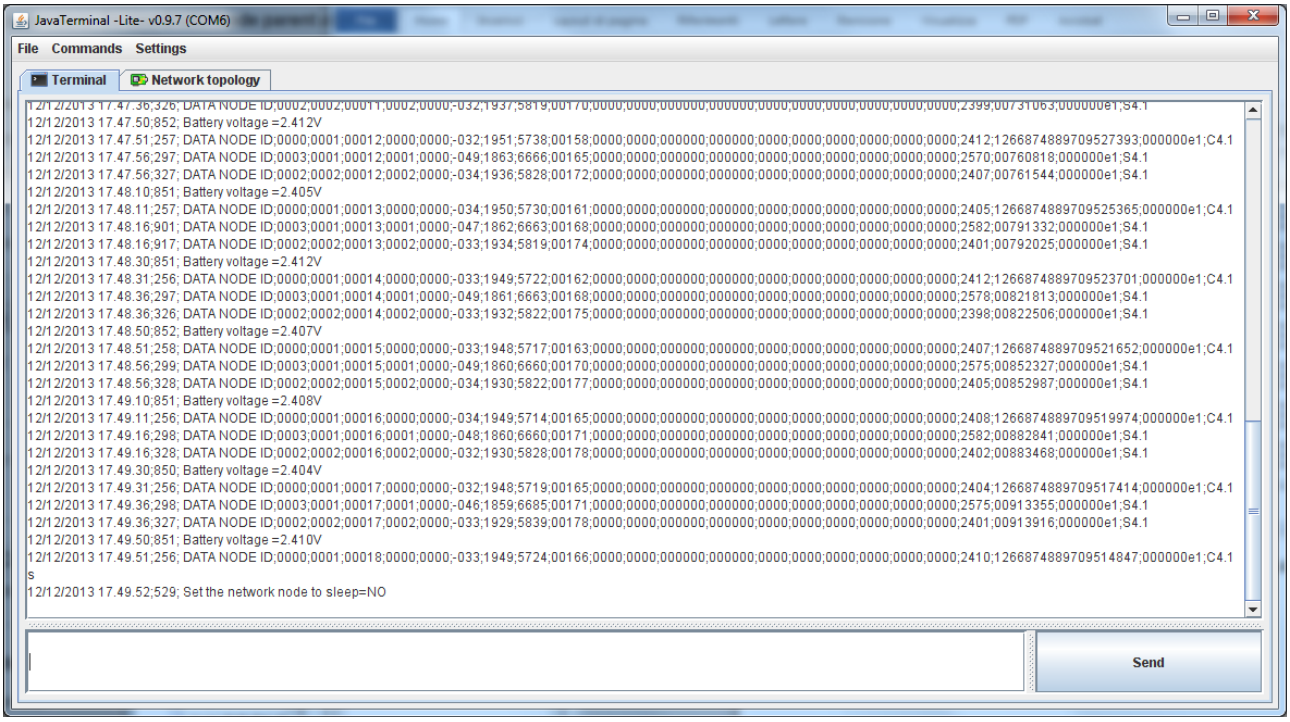

Fig. 8.6 Command s to stop the data sampling

Afterward the user has to check the network radio channel and network pan id with command t.

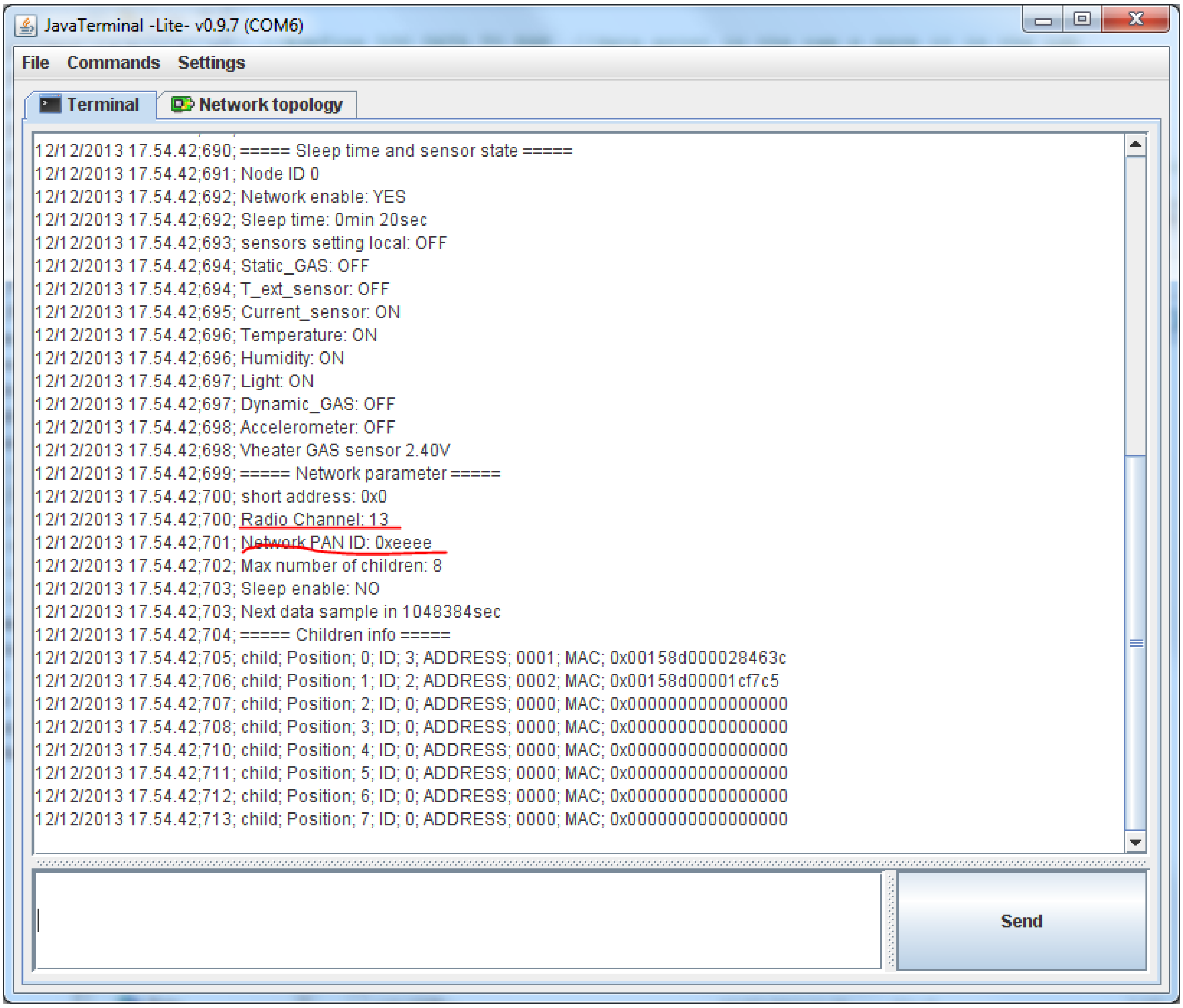

Fig. 8.7 Command t to collet radio channel and PAN ID information

In the example the network has radio channel 13 and PAN ID 0xeeee.

If the new firmware has default radio channel and net PAN ID different from the current one, the user has to change the current network radio channel and PAN ID with the command x.

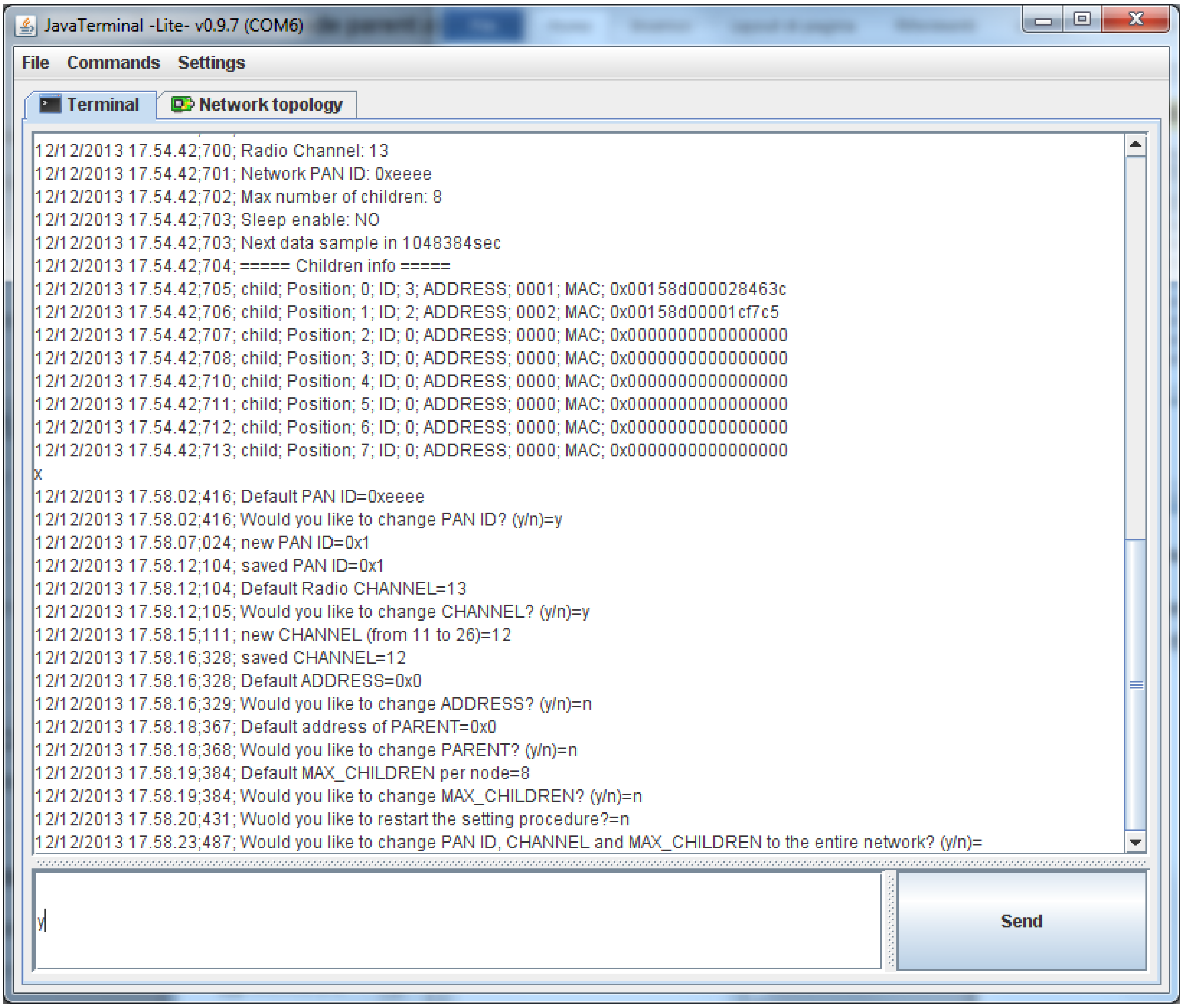

Fig. 8.8 x to set the network parameters

After set the network parameters the network will restart with new radio channel and PAN ID.

Now it is possible to upload the new firmware over the air.

The user has to select commands Upgrade SleepyEnd Firmware.

Select the desired SleepEndRouter_JN5148.bin firmware file.

Press Open to start the procedure.

The procedure will take 10min to upload the entire firmware.

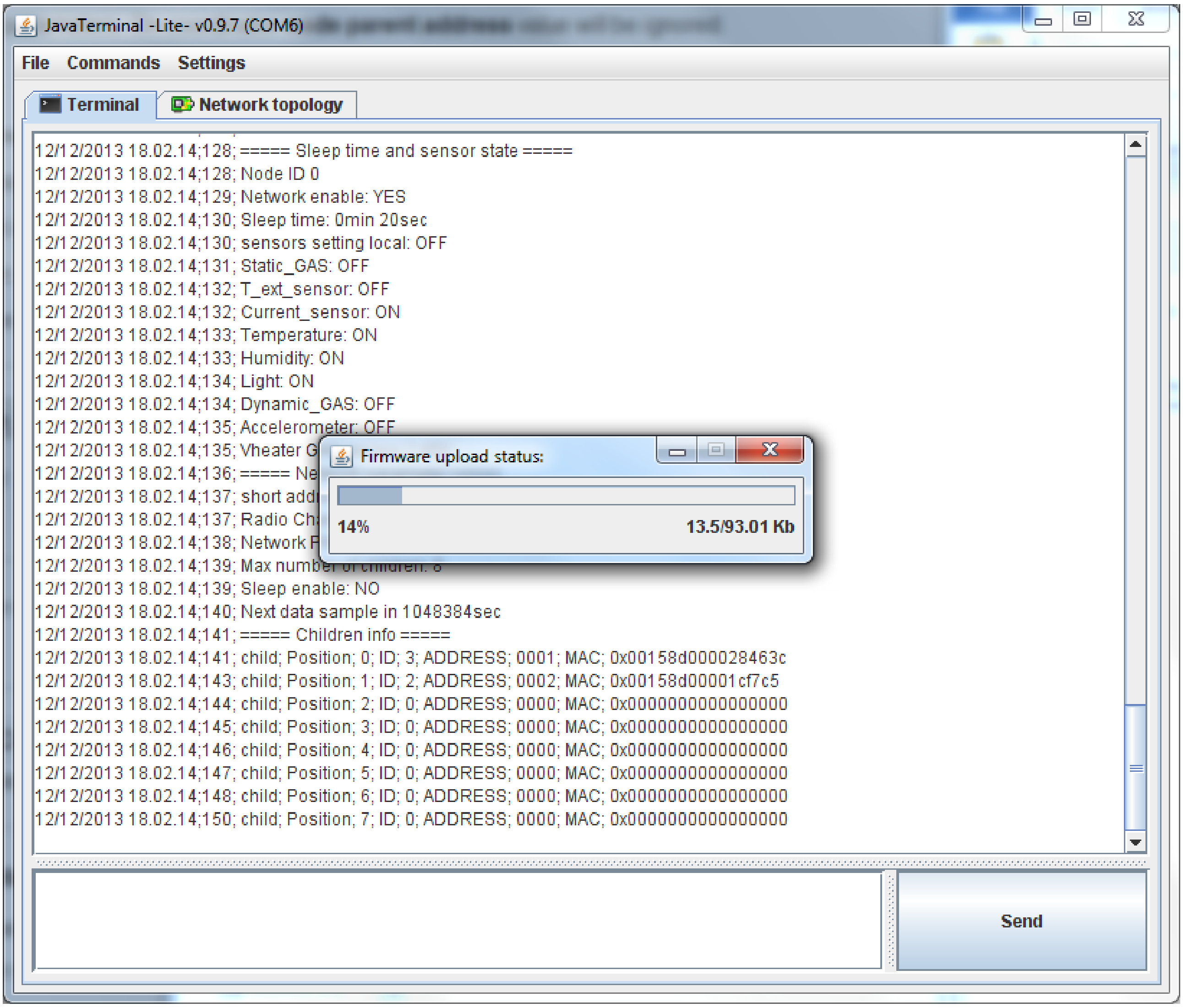

Fig. 8.9 Firmware update over the air

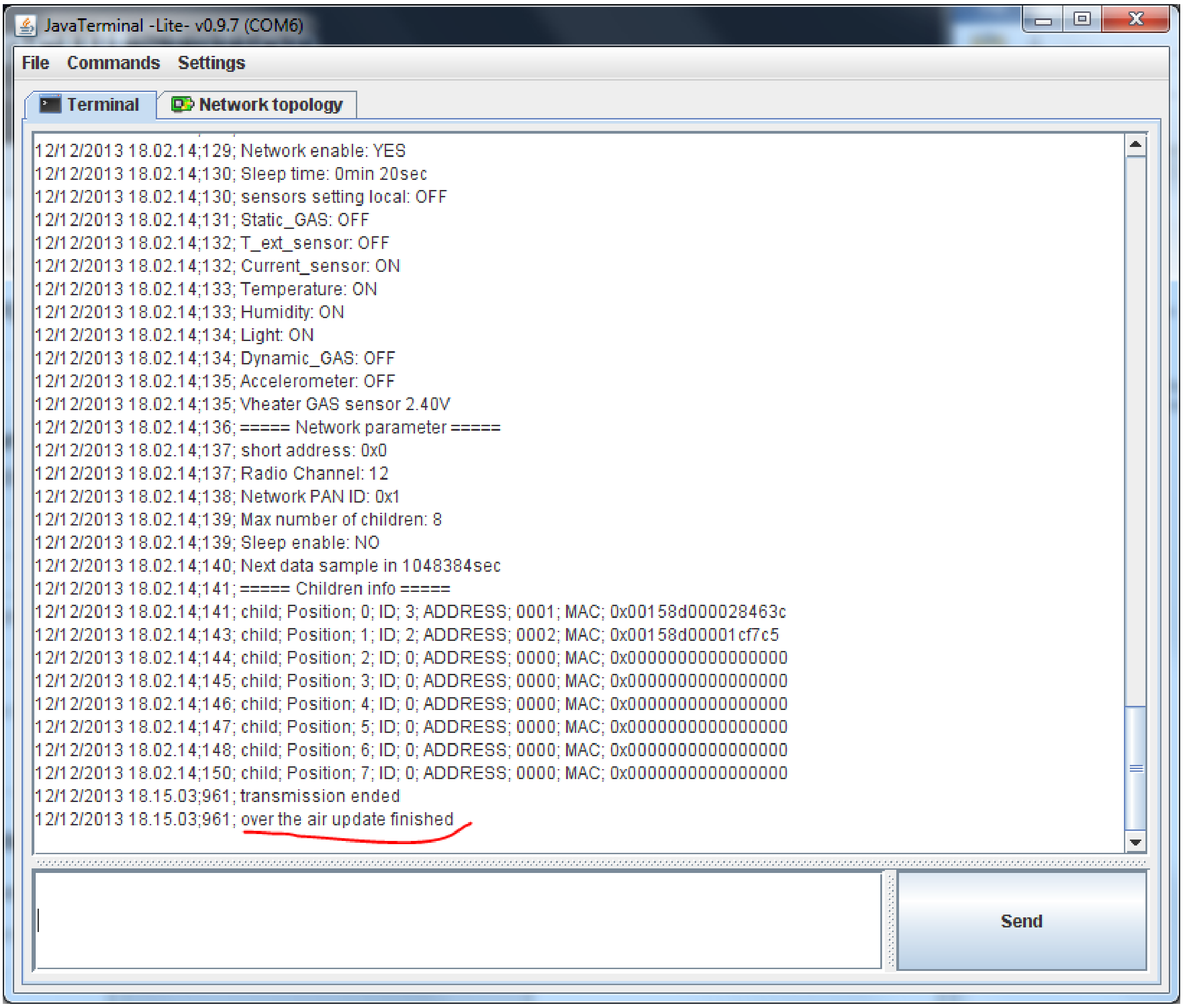

Fig. 8.10 Over the air firmware update

When the procedure end the system print over the air update finished.

Some firmware update will erase the network setting (like update to firmware 4.1 from previous version). To quickly reconfigure the network the commands “ma” and “mt”could be used, as explained below.

9 Deployment

To install a new network the user has to connect the coordinator mote (which is a W24TH with coordinator firmware inside) to the gateway by USB cable.

Fig. 9.1 3EncultWSN installation and test with 100 motes

After that he/she has to position the others W24THs in the building where he/she would like to collect data (the others W24THs must have the SleepyEnd firmware inside). During the installation the user have to switch on each mote. Every mote has an ID number that identify the mote physically. The coordinator has ID equal zero.

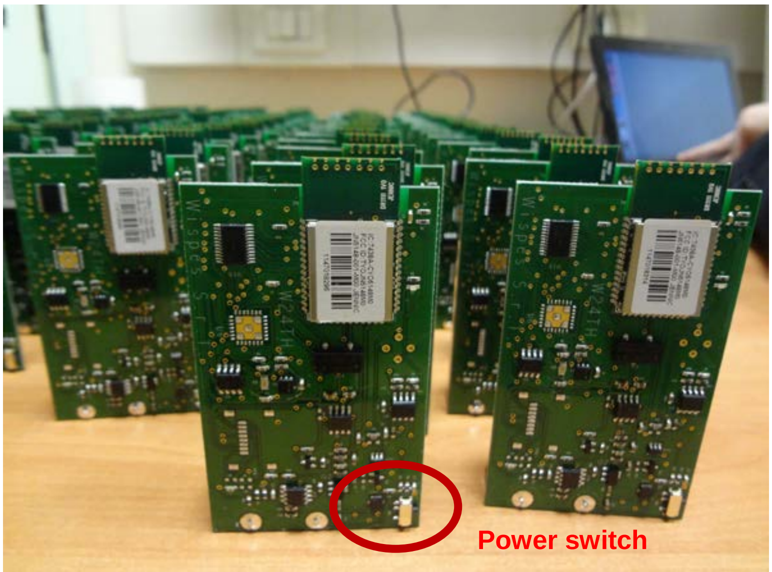

Fig. 9.2 W24TH switched on

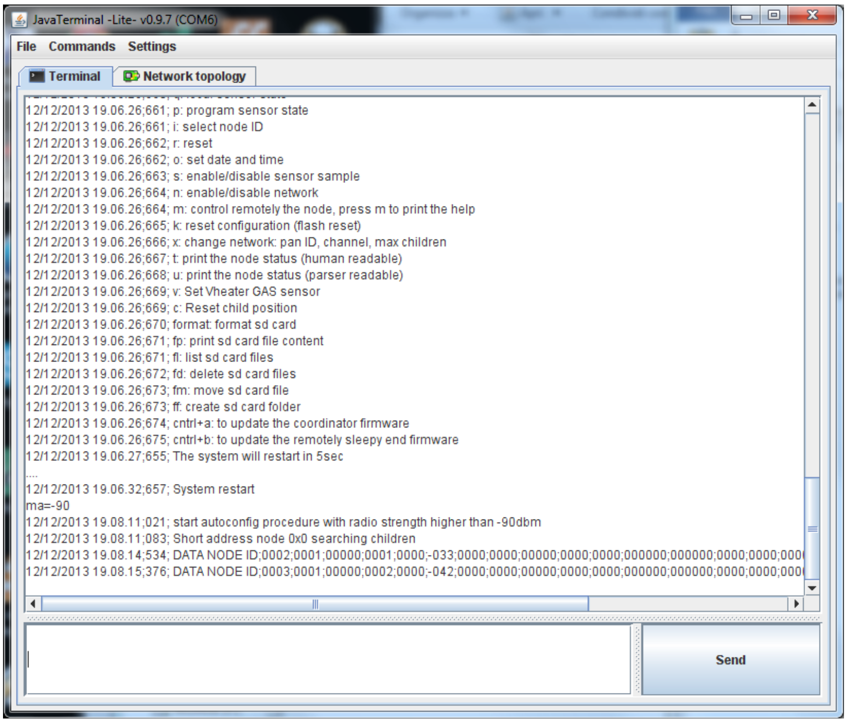

Fig. 9.3 JavaTerminal with command ma=-90 typed and executed

After the positioning, the user can configure the network automatically typing the command ma=-90 to the javaTerminal. The network start to construct itself and the network structure appears in the network topology tab.

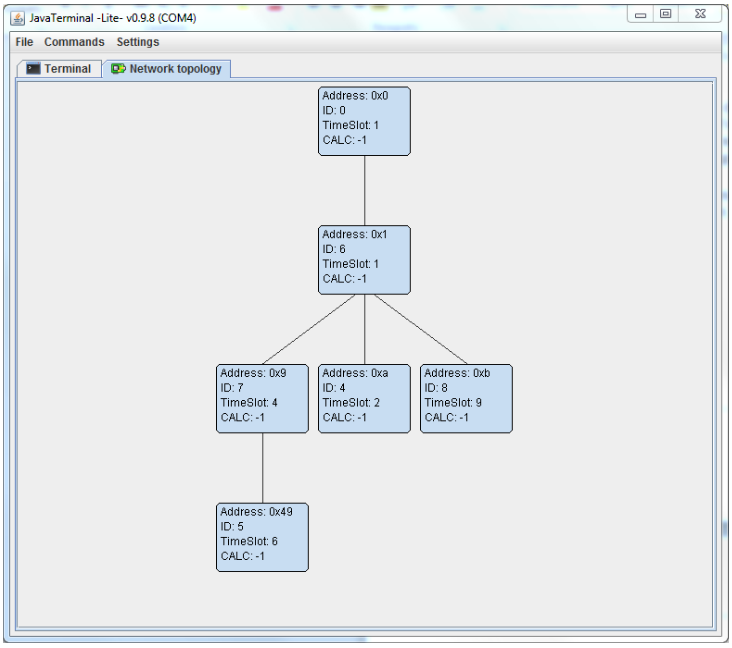

Fig. 9.4 JavaTerminal with the network structure

When each mote is inside the topology, the user has to select “Commands” – “Calculate Time Slot” to set the transmission time of each mote.

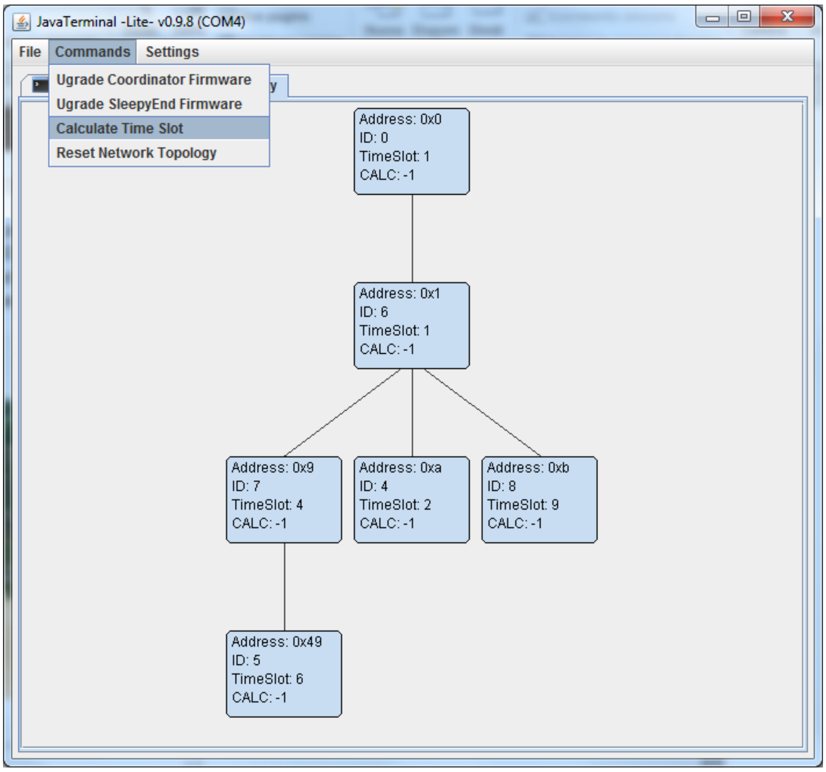

Fig. 9.5 JavaTerminal Commands – Calculate Time Slot selection

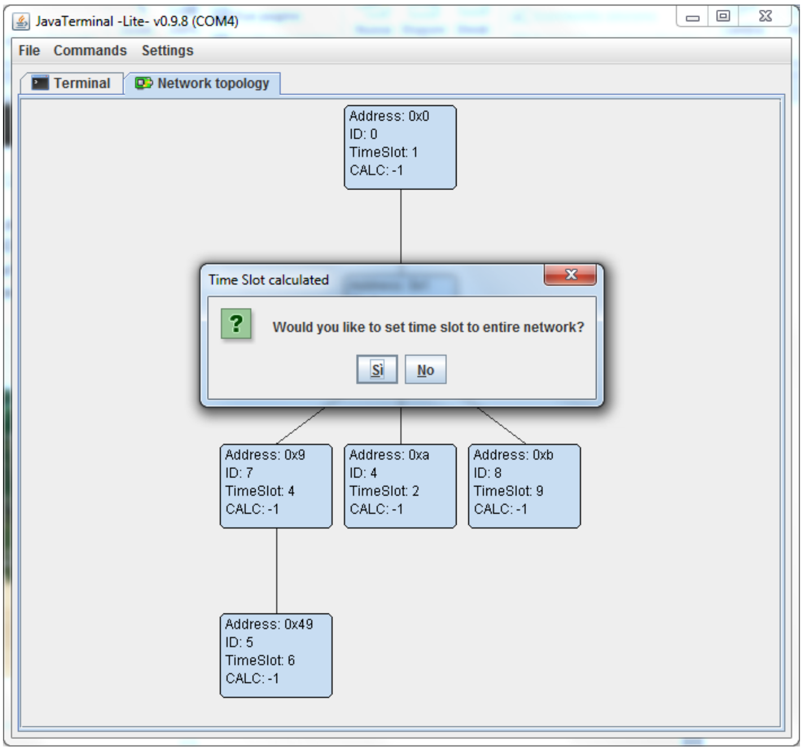

Fig. 9.6 JavaTerminal Time Slot calculation

Select yes and each mote will be set with the calculated time slot.

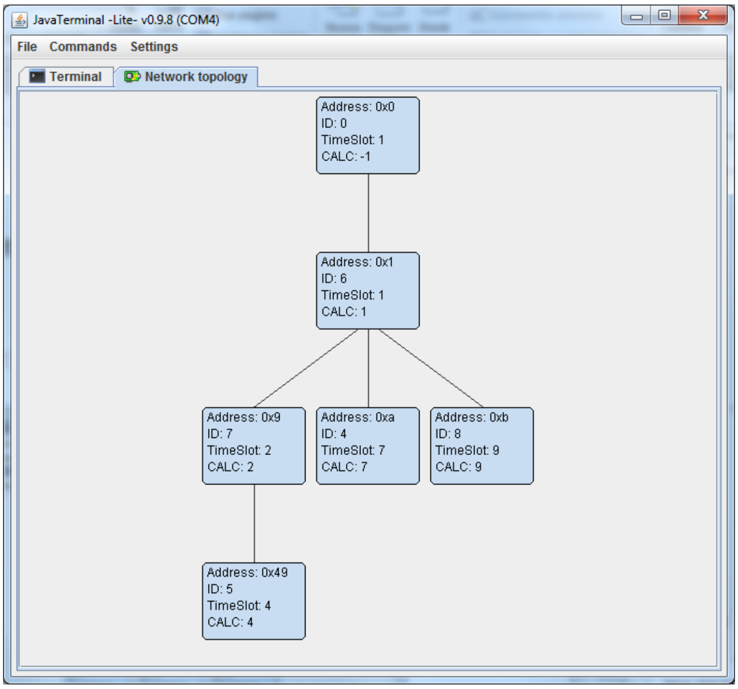

Fig. 9.7 JavaTerminal with network topology with right time slot

Now the network is ready to collect data and the user can set the time sample interval typing w=60 to receive a data every minute.

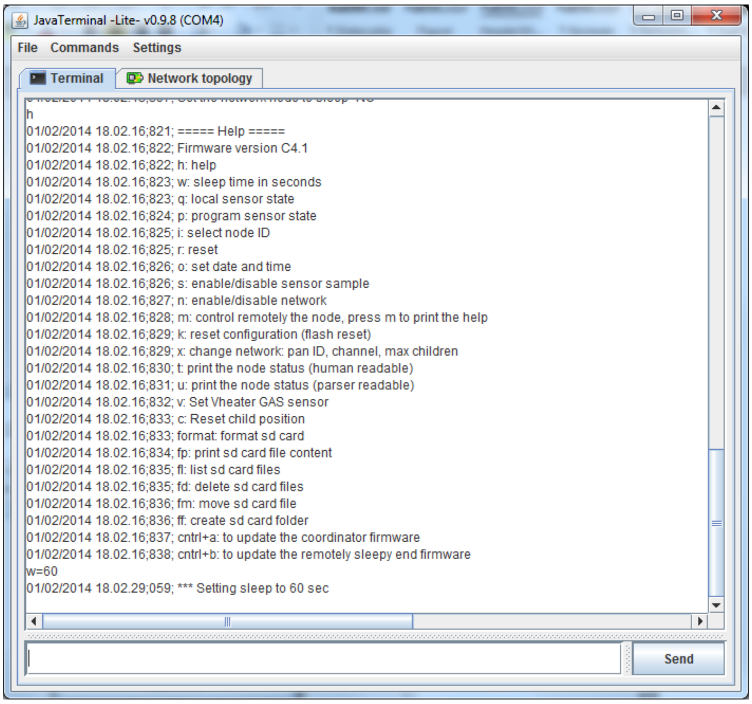

Fig. 9.8 JavaTerminal with command w=60 executed

To start the data sample the user has to type s. The user can stop the data acquisition to perform a network maintenance typing again s.

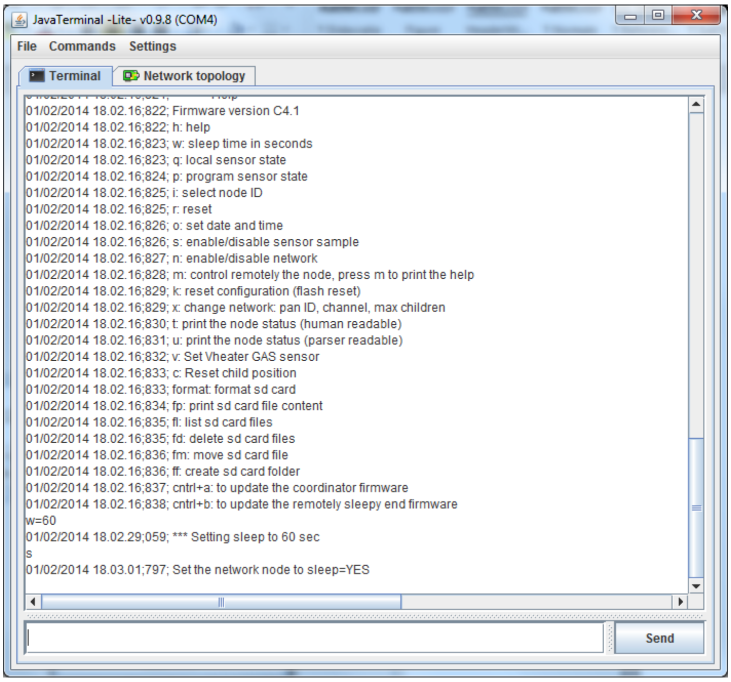

Fig. 9.9 JavaTerminal with command s executed

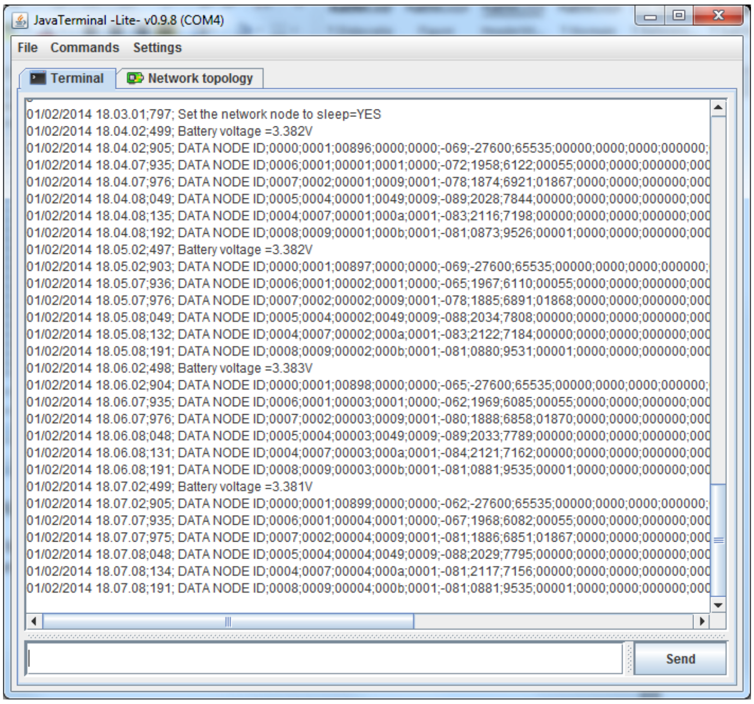

Fig. 9.10 JavaTerminal with incoming data

The user can select to store the data incoming locally in a text file, or and remotely to a FTP server. The FTP server configuration is inside the file ftp.xml that must be in the same directory of the javaTerminal.

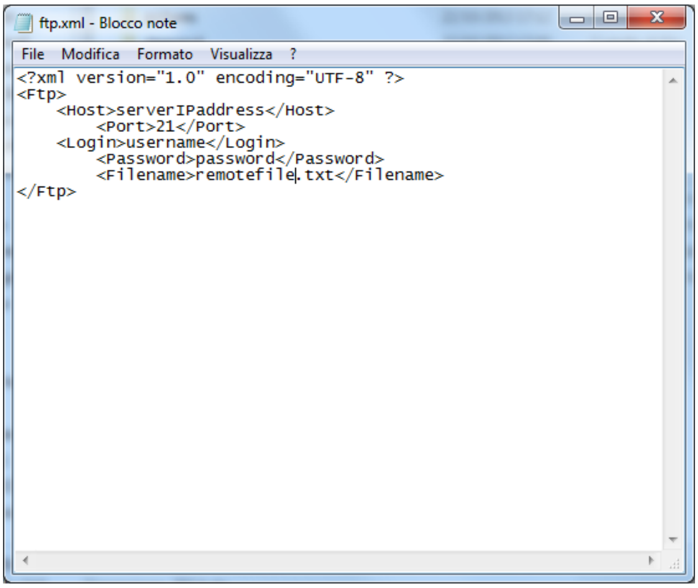

Fig. 9.11 ftp.xml file opened with windows block notes

In the ftp.xml file the user has to set the FTP server url, user name and password and the name of the remote file that will be update with the new sensors' data collected.

To activate the both local and remote data logging selects the menu “Setting” – “Log” as shown in the picture.

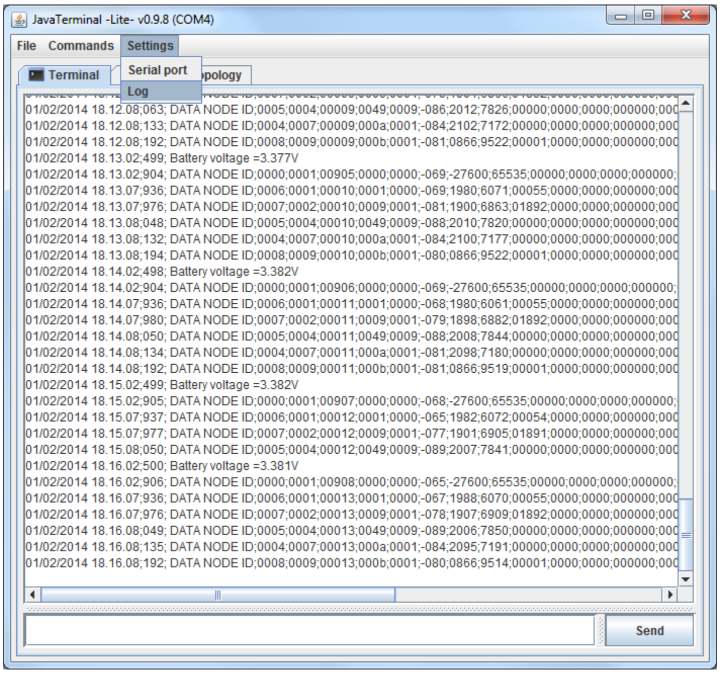

Fig. 9.12 JavaTerminal Setting – Log selection

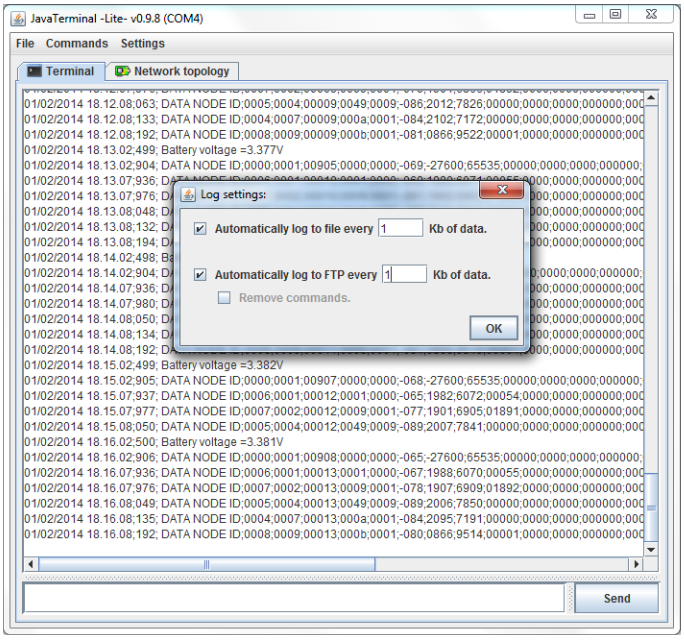

Fig. 9.13 JavaTerminal log setting

In the “Log setting” window checks the automatically log to file and FTP, moreover the user can select every kbyte the log will be performed. To have a fast log select

1kbyte, whereas if you have a slow internet connection or connection not available every time, you can choose to log data with higher interval like 100kbytes.

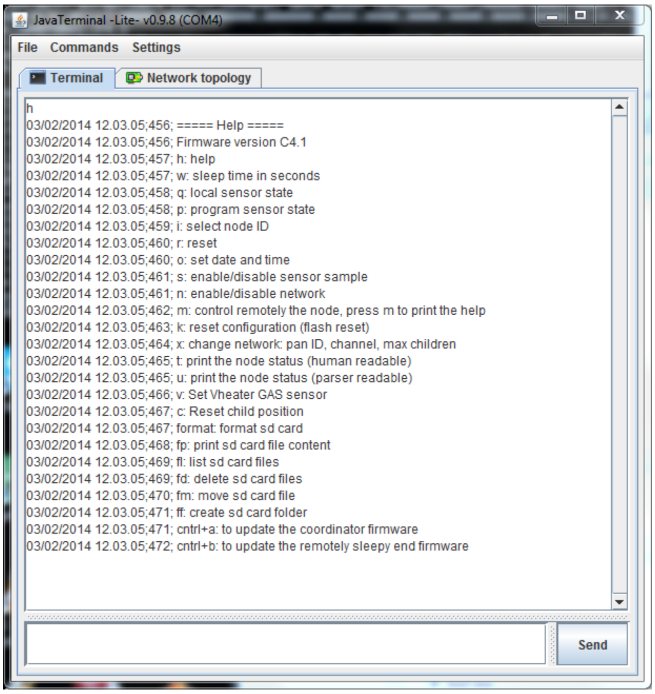

Fig. 9.14 JavaTerminal help view

The javaTerminal has a built in help viewable typing h. With the set of commands present in the system the user can customize the whole network to reach the desired performance and configuration.

10 Testbed

The network stack and the hardware have been tested in the Unibo labs. In particular few test networks have been used, from 20 to 100 nodes. The test has been conduct for months keeping the data sample time every 30 sec, in the way to have a faster network response in terms of reliability and wear out. The tests have underlined bugs and permit to refine the firmware in the way to improve the network life time, the easiness of installation and the sensors connectivity to the node.

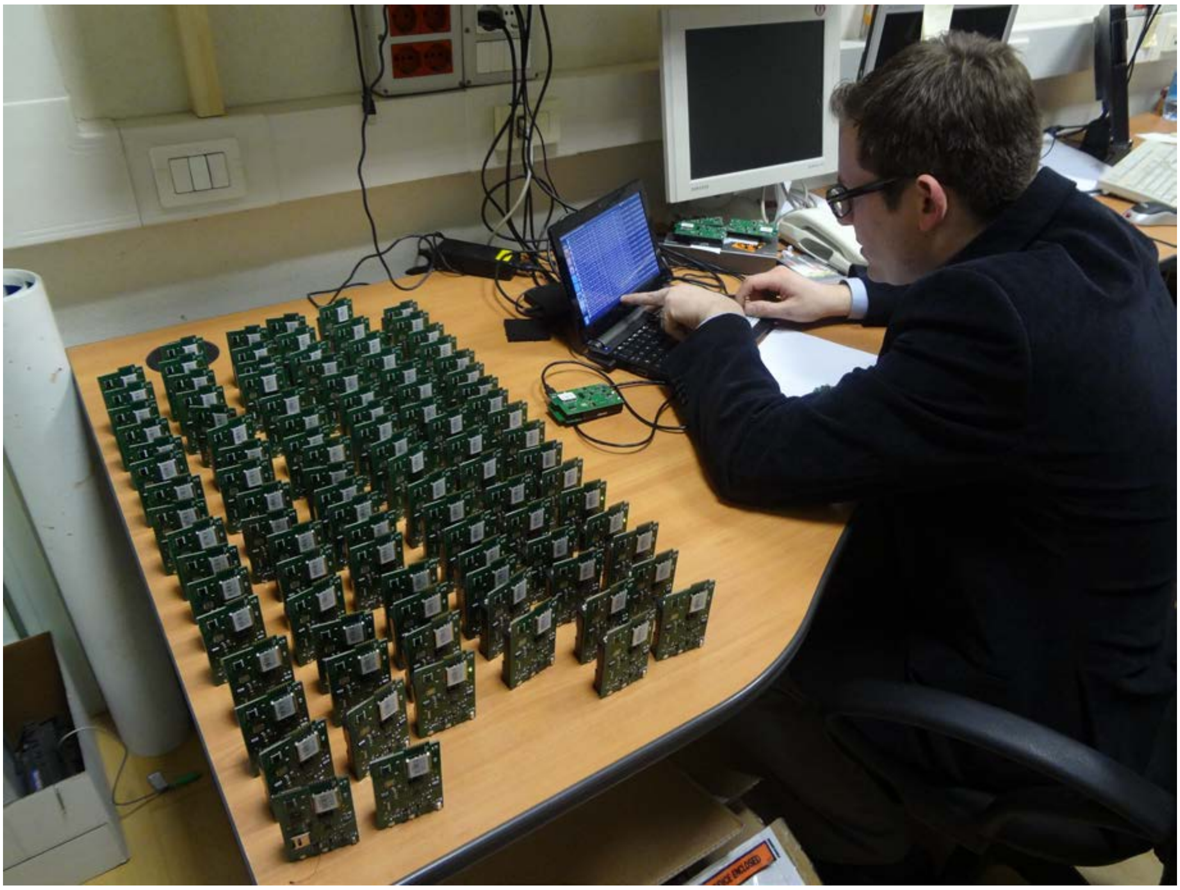

Fig. 10.1 100 motes testbed

The exhaustive tests conducted have brought the underdeveloped WSN to highest quality of service and reliability. Right now the WSN has the capability to operate for years without maintenance. The tests conducted have addressed even the on board sensors accuracy, that has been compared with commercial instrument, proving the exploitability of the developed system.

Moreover the prototypes boards have been internally produced and tested to guarantee the extensibility and modularity of the 3EncultWSN.

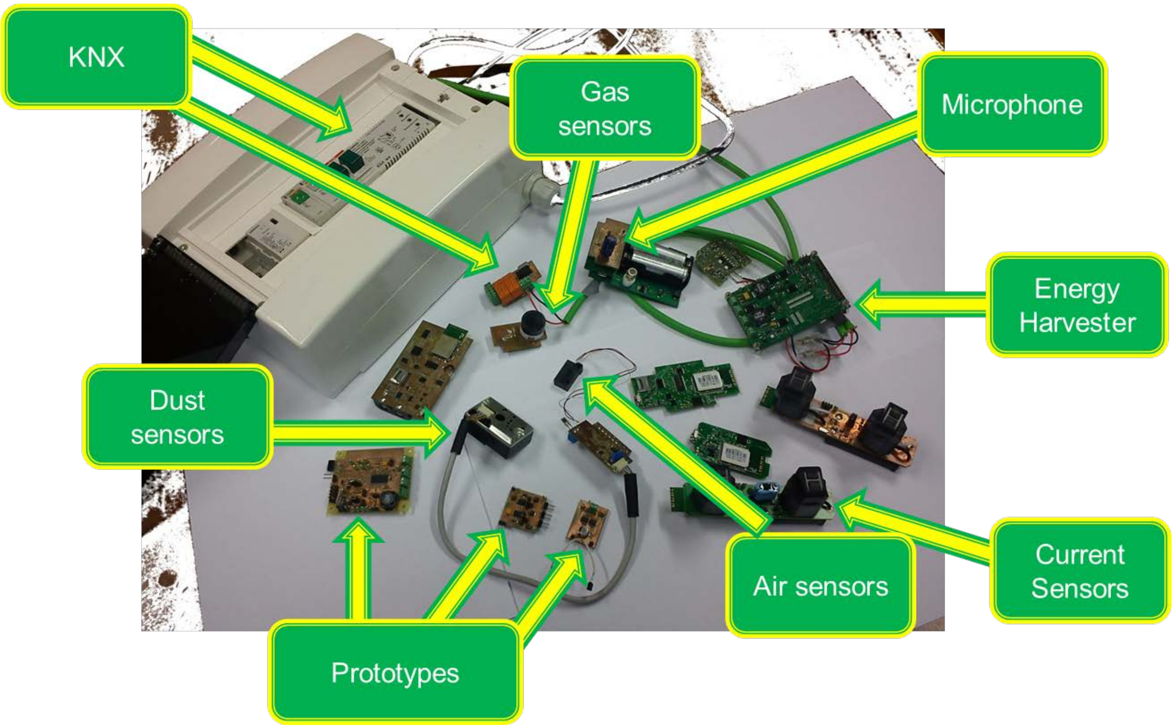

Fig. 10.2 Expansion boards prototypes.

The prototype developed and tested are:

- 8 Channel analog acquisition board connected to temperature, AC current sensors and heat flux.
- Chimical gas sensor interface.
- Air velocity sensor interface.
- Dust sensor.
- Microphone board.
- Capacitive humidity sensor interface.
- Relay actuator interface.
- KNX interface.
- Energy harvesting interface connected to solar cell, wind micro turbine and AC current clamp to collect energy from the surrounding environment.

Morever the CS3 “Palazzina della viola” has been used as permanent test bed and living lab to test the reliability of the 3EncultWSN. Till March 2012 the network has worked almost uninterruptedly, except a few interruptions, which most of them due to gateway supply problem.

Others installations of the 3EncultWSN are in the:

CS7: Industrial Engineering School of Béjar, Salamanca, Spain where the WSN is connected with the BMS developed by project partner Cartif in the Task4.3.

11 3Encult case studies installations

11.1 CS3: Palazzina della Viola, Bologna, Italy

Fig. 11.1 Palazzina della Viola

The “Palazzina della Viola” is a 15th century building with a unique architecture structure. The building has a large room in the first floor with frescos distributed in four walls. Moreover there are several celling painted all over the building. The Palazzina has been renewed in the 2011 with a new HVAC system based of fun coil units, air ventilation and electric heater. The system has a central control unit in the basement and could be managed remotely by the Department of Energy of University of Bologna.

The building has been monitored by a Wireless Sensor Network before and during refurbishment and it is monitoring with 3EncultWSN since its utilization after renewed. The aim of monitoring is to analyse the building climate in order to find hazard condition for fresco, discomfort for the users and energy waste.

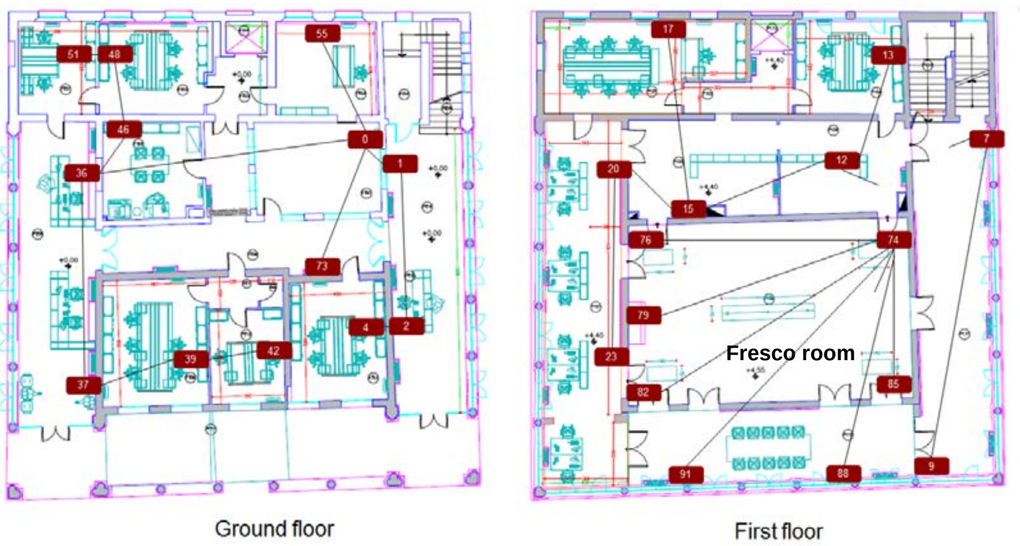

Fig. 11.2 CS3: Palazzina della Viola after refurbishment 3EncultWSN installation

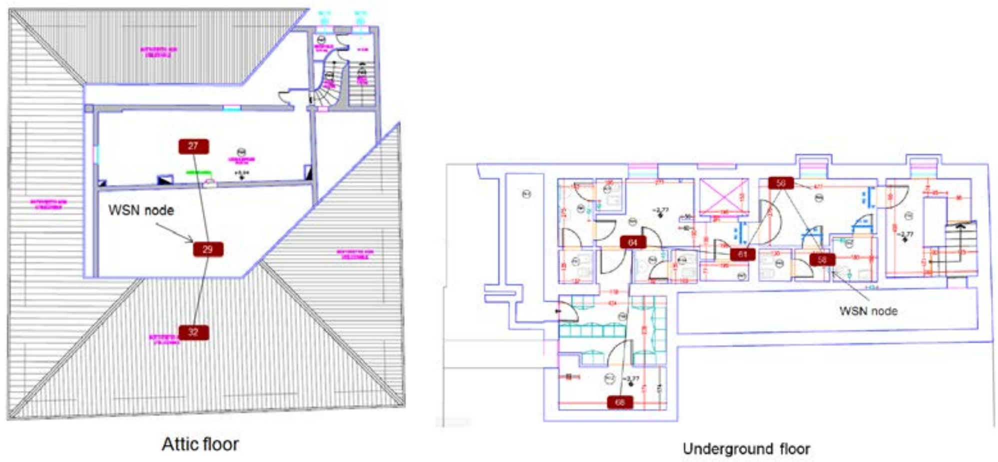

Fig. 11.3 CS3: Palazzina della Viola after refurbishment 3EncultWSN installation

The 3EncullWSN is sending the data collected to a web server that permits to the users to visualize them. The web system has several user types that can have a restricted data access or complete data access. The reader can access to a restricted session to the url http://137.204.213.210 user: guest password: guest.

The data has been used to analyse the building climate condition during the entire monitoring period. The analysis has involved the plot of humidity and temperature maps and graphs to underling critical condition in order to find better HVAC scheduling and utilization.

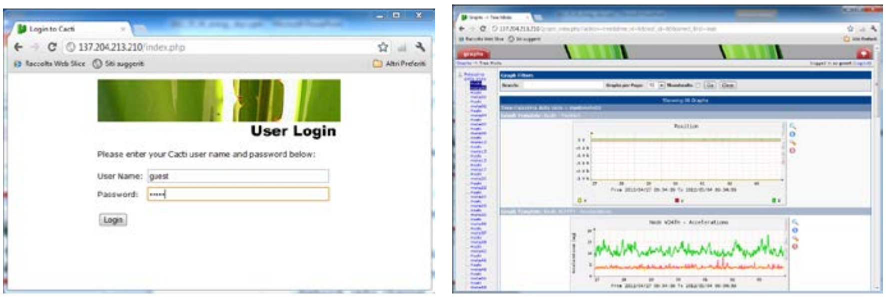

Fig. 11.4 Web data visualizer interface

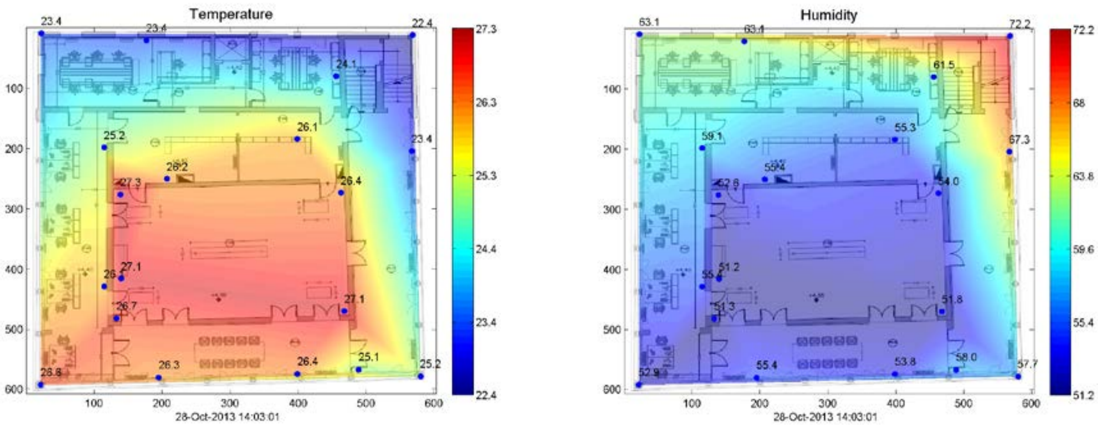

Fig. 11.5 Temperature and Humidity maps

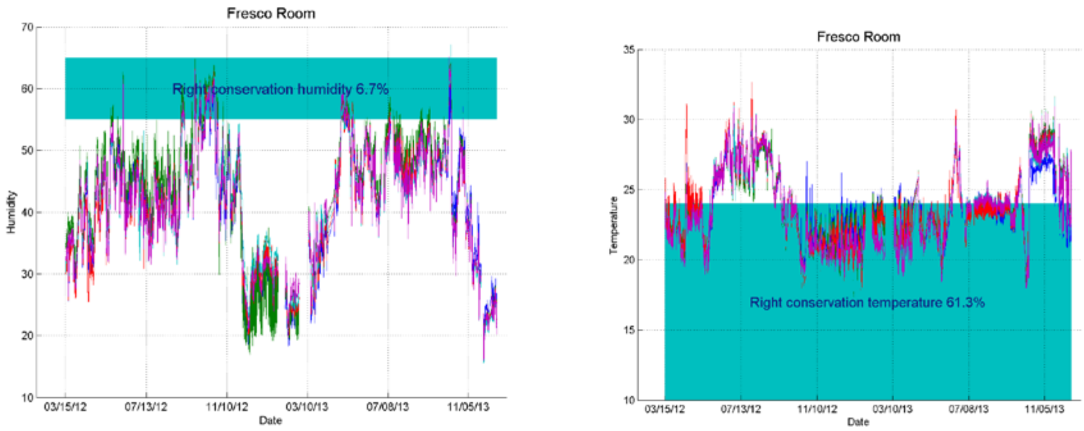

Fig. 11.6 Fresco room Temperature and humidity after refurbishment

The analysis has shown a not good HVAC scheduling that puts the fresco room in a critical temperature climate for 38.7% of the time. Moreover the fresco room is in a critical humidity climate for the 93.3% of the time underling the need to introduce a humidifier system. The department of energy of Bologna University is studying the data and result to perform a better scheduling of the HVAC system and to define how to integrate the 3EncultWSN monitoring system in the HVAC system control in order to have better users comfort, heritage goods preservation and reduced power consumption.

11.2 CS7: Industrial Engineering School of Béjar, Salamanca, Spain

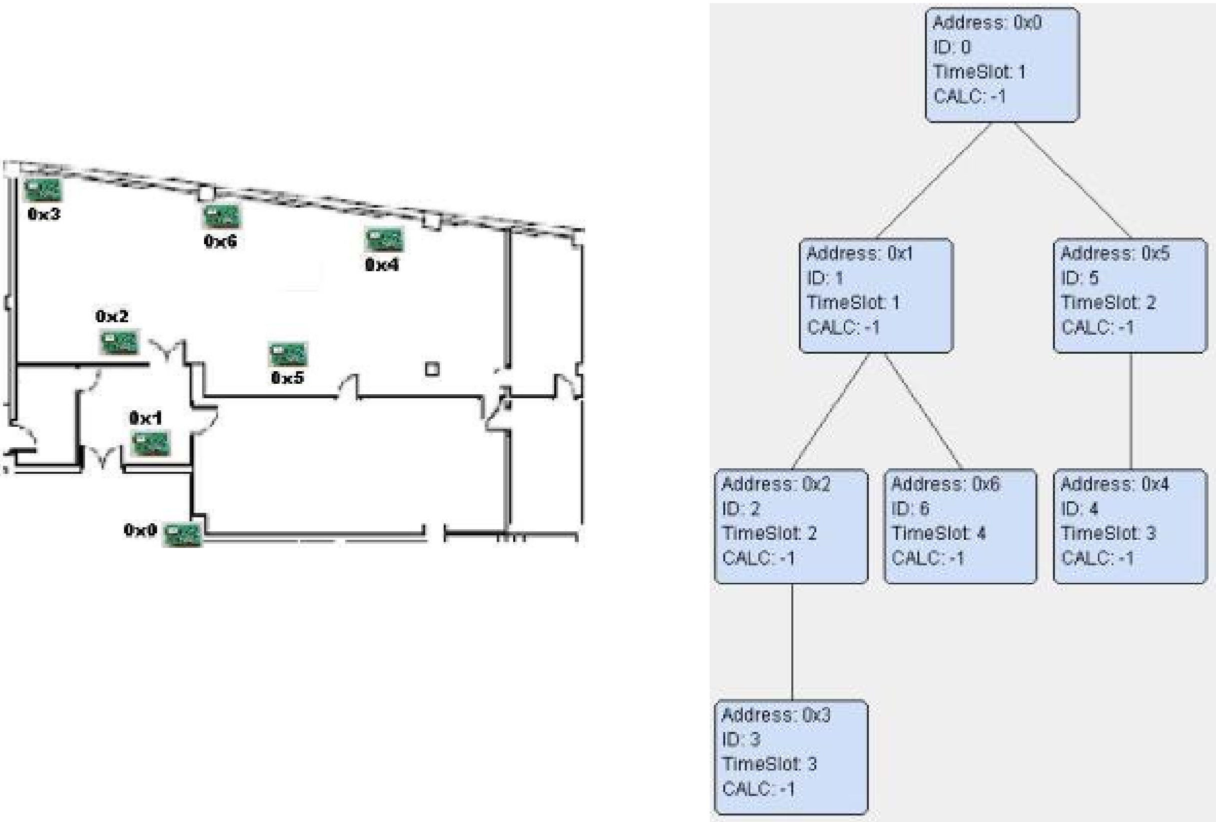

Fig. 11.7 WSN installation in the Industrial Engineering School of Béjar, Salamanca, Spain.

The 3EncultWSN is deployed in the case study CS7 with seven motes collecting data from 21 sensors. The network cover the library of the Engineer school and it is directly connected with the BMS. The installation has been realized the October 28th 2013 and it is working fine till now. The aim of this installation it is to prove the compatibly between the 3EncultWSN and the BMS.

11.4 CS2: Palazzo d’Accursio, Nologna, Italy

An installation of seven motes is planned for March 2014. The installation will cover the “Sala degli Stemmi”, the “Coat of harms” room, to collect environmental data like temperature, humidity and light intensity in order to support the under installation KNX windows management system. The core of the system wil be the KNX building automation part that will be integraterd by the WSN system. The WSN will use the KNX interface that will bring the collected data to the KNX system.The final application has to perform a smart window management that will increase the user comfort preserving the cultural heritage goods present in the room.

Fig. 11.8 Coat of Harms room windows detail

Fig. 11.9 Coat of Harms room outside windows detail