University of Camerino

School of Science and Technology

Course: Computer Science Class LM-18

Environment Monitoring and Management of forest fire prevention using WSN technology

Experimental Degree Thesis in Software Engineering

Advisor
Prof. Fausto Marcantoni

Author
Luca Pennacchietti

Coadvisor
Dott. Francesco De Angelis

Academic Year 2011 - 2012

Ho visto più lontano degli altri, perché stavo sulle spalle di giganti.

Isaac Asimov

Ancora una volta, i miei ringraziamenti vanno alla mia famiglia che mi ha permesso di raggiungere questo obiettivo, il loro incoraggiamento e sostegno sono stati fondamentali, e ai miei Amici, senza i quali mi sarei sentito perso.
Ringrazio di cuore il Dott. Francesco De Angelis, guida instancabile e buon amico, e il Prof. Fausto Marcantoni per il suo aiuto sempre presente.
Un ringraziamento va a Paolo Senigagliesi per la collaborazione nella realizzazione di questa tesi.
Infine, ma assolutamente prima di ogni altra cosa, tutto l'amore e la gratitudine alla mia ragazza, Manuela, che ha percorso con me ogni singolo passo.

Contents

List of Figures

Listings

List of Tables

Abstract

The main objective of the work is to design and develop a forest monitoring system can provide early warnings of any changes in those values that may indicate the presence or risk of a fire. The system was developed to be readapted in similar circumstances, namely, all those types of situation where it's required the use of sensors to investigate and prevent potentially dangerous phenomena. It is chosen to carry out the study on survey data, in particular in the field of forest fire prevention, or at least to make the actions of securing the faster possible, thanks to constant monitoring of areas by means of devices equipped with different types of sensors. The monitoring is done through various sensors of different kinds, assembled together in a single detection unit, able to report the slightest variation in levels of variables that may indicate the real risk of fire or even presence. These units are then linked together in a wireless sensor network capable of transmitting data from each device to a server on which there is a demon who has the task of collecting the data received and sent them to the database. The WSN will not be intrusive, because is based on camouflaged devices, to have a low impact on the environment. The prototype includes the software for monitoring WSN deployed on the territory, allowing the recovery of data and sending commands to nodes. The data collected into a database are analyzed to generate real-time alerts for operators, this can alert the authorities with response times less than the normal sightings or reports. Through a simple web interface, the data are displayed to the user indicating the presence of abnormalities in values obtained. The ultimate goal of the thesis is to demonstrate the feasibility of a monitoring system based on wireless sensor network capable of sending data to a server, allowing to intervene promptly, in potentially hazardous situations for the environment and human activities.

Abstract

Il principale obiettivo del lavoro svolto é quello di progettare e sviluppare un sistema di monitoraggio boschivo in grado di garantire tempestive segnalazioni su eventuali variazioni di quei valori che possono indicare la presenza o il rischio di un eventuale incendio. Il sistema é stato sviluppato per essere il più possibile riadattabile in circostanze simili, e cioè tutti quei tipi di situazione in cui si richiede l'utilizzo di sensori per la rilevazione di dati per studiare e prevenire fenomeni potenzialmente pericolosi. Si é scelto di eseguire lo studio in particolare sulle rilevazioni dei dati in ambito boschivo per prevenire gli incendi, o per lo meno per rendere gli interventi di messa in sicurezza il più tempestivi possibile, grazie al monitoraggio costante di aree per mezzo di periferiche munite di diversi tipi di sensori. Il monitoraggio avviene tramite l'impiego di diversi sensori, di vario genere, assemblati insieme in un'unica unità di rilevamento, in grado di segnalare anche la minima variazione nei livelli di grandezze che possono indicare il reale rischio d'incendio o addirittura la presenza. Dette unità sono poi collegate tra loro in una Wireless Sensor Network, capace di trasmettere i dati forniti da ciascun device a un server sul quale é presente un demone che ha il compito di raccogliere i dati ricevuti e inviarli al database. La WSN non sarà intrusiva essendo basata su dispositivi mimetizzati, per avere un basso impatto sull'ambiente monitorato. Attraverso la pagina web é inoltre possibile inviare dei comandi a ogni periferica, per modificarne aspetti del comportamento. L'obiettivo finale della tesi é pertanto dimostrare la possibilità di realizzare un sistema di monitoraggio basato su una Wireless Sensor Network, capace di inviare dati a un server, che permetta di poter intervenire tempestivamente, nell'ambito specifico in caso d'incendio, in quelle situazioni potenzialmente rischiose per l'ambiente e per le attività umane.

Introduction

The main objective of the work is to design and develop a forest monitoring system can provide early warnings of any changes in those values that may indicate the presence or risk of a fire. The system was developed to be readapted in similar circumstances, namely, all those types of situation where it's required the use of sensors to collect data to investigate and prevent potentially dangerous phenomena. Some examples include the detection of levels of a river to prevent damage caused by flooding, detections of dangerous levels of gas in enclosed areas, surveys of structural failure of infrastructure, etc.. The project is therefore able to adapt to different needs and types of situations.
It is chosen to carry out the study on survey data, in particular in the field of forest fire prevention, or at least to make the actions of securing the faster possible, thanks to constant monitoring of areas by means of devices equipped with different types of sensors. The study of the dynamics of development and spread of fire allows to distinguish some necessary elements and different moments in its development phase. To take place a fire three elements are required:

- Fuel
- Oxidizing
- Heat

All three of them have to be present for developing a burning, without one of these fire is not possible. In most cases, source of fire is known and can be detected by monitoring the presence of physical quantities such as temperature, humidity, CO2 levels, etc.. ... The

monitoring is effective in the principles of fire detection, but it should be done before it is propagated.
There are different stages of development of a fire:

- Phase of ignition: main phase of the fire, combustible substances begins the process of combustion;
- Propagation phase: characterized by low temperature and low fuel quantity involved;
- Flash Point: gives rise to the flash-over sharp rise in temperature and massive increase in the amount of material that participates in the combustion. Is considered the point of "no return" in the outbreak of fires;
- Generalized fire: most of the material involved in the combustion;
- Extinction and / or cooling: the end of combustion. Cooling is the final post-fire and that involves cooling the affected area.

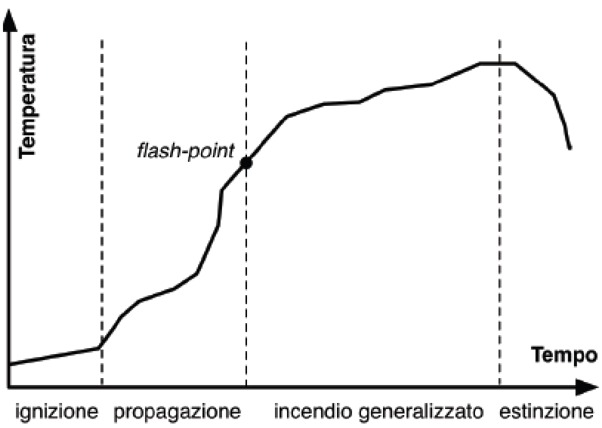

Figure 1: Different stages of development of a fire

In the dynamics of a fire is supposed to be able to intervene with active measures to prevent the first phase of flashover (or within minutes of it), after which massive structural approaches should be applied to extinguish large areas.

The choice of places to observe has been made thanks to the interest of the mountain community of Avacelli, in which inspections were made in order to choose which area best suited to studies, also based on the level of risk of fire. The monitoring is done through the use of various sensors of various kinds, assembled together in a single detection unit, able to report even the slightest variation in levels of temperature, CO2 in the air and other variables that may indicate the real risk of fire or even presence. These units are then linked together in a wireless sensor network capable of transmitting data from each device to a server on which there is a demon who has the task of collecting the data received and sent them to the database. The WSN will not be intrusive, because is based on camouflaged devices, to have a low impact on the environment monitored and enable precise interventions in case of occurrence of fire. For effective implementation of the network need to:

- integrate the wireless sensor node in charge of measuring temperature, humidity, and carbon dioxide;
- develop a low power node with long battery life (at least two years);
- efficiency from the point of view of the operation (minimizing false alarms);
- Establish a low cost solution adopted by the agencies (and certified by agencies).

The prototype includes the software needed for monitoring WSN deployed on the territory, allowing the recovery of data and sending commands to the sensor nodes. The data collected are then analyzed to generate real-time alerts for operators, this can then alert the authorities with response times less than the normal sightings or reports. Through a simple web interface, the data are displayed to the user indicating the presence of abnormalities or significant changes in values obtained. The choice of sensors that communicate via wireless connections is dictated by the fact that any device with onboard sensors will be installed in a natural environment, such as wood, besides the obvious reasons of convenience, there have been attempts to minimize the impact of the latter with the environment itself. The sensor network is designed so that each device was referring to a central receptionist device sending data to the server on which the daemon is installed. Through the web page you may also send commands to each device, to change aspects of behavior, such as the frequency with it sends updates of the data collected. For the development of software that manages the flow of data from each device, we chose to use the development kit from Microsoft, Visual Studio 2010, using the environment. Net 4.0 for the development of the Demon place on the server, and ASP . NET 4.0 for the creation of web pages that provide an interface for users. For the realization of the database has been chosen MySQL updated to version5.5 and using a connector has been possible to interact with Visual Studio. The ultimate goal of the thesis is therefore to demonstrate the feasibility of a monitoring system based on wireless sensor network capable of sending data to a server, allowing to intervene promptly, in particular in case of fire, in potentially hazardous situations for the environment and human activities. Also wanted to show that it is possible to realize a system that is adaptable to many situations involving the collection of data through sensors to avoid hazards or to perform statistical studies of nature.

The thesis is structured into n sections, the first will be devoted to the introduction of the problems resulting from fires and the consequences that they cause, both in human activity on the environment. The second part focuses on the actual description of the system developed, explaining the architecture of the same, the shape of the sensor network, the database structure in which we collect data, and the illustration of the web interface through which interact and observe the behavior of the network.

Chapter 1

Fires in Italy

In Italy the forests cover more than 9,800,000hectares of land, approximately 32% of land area. Italian forests are among the most important in Europe for its breadth and variety of species. Is a valuable asset in the environment and the economy, the balance of the land for the conservation of biodiversity, habitats for animals and the landscape.

1.1 Statistics

Over the last twenty years, according to data from the Civil Protection, forest fires have destroyed more than 1,100,000 hectares of woodland: an extension greater than that of Abruzzo. [2]

From January 1 to December 31, 2010 throughout the country there have been 4,884 forest fires that have covered a total area of 46,537 hectares, of which 19 357 wooded and 27 180 unwooded. Compared to 2009 there were 538 fires in less, the total area affected is less than 26,818hectares and 11,703 hectares of the woodland. In percentage terms the reduction of the number is approximately 10%, the total area is decreased by about 37% and wooded one is decreased about 38%.

Following are summary graphs of the number of fires that have developed in the year 2010 in Italy, by region. There are acres of land that have been affected by fires, areas covered by forests and those without forest.

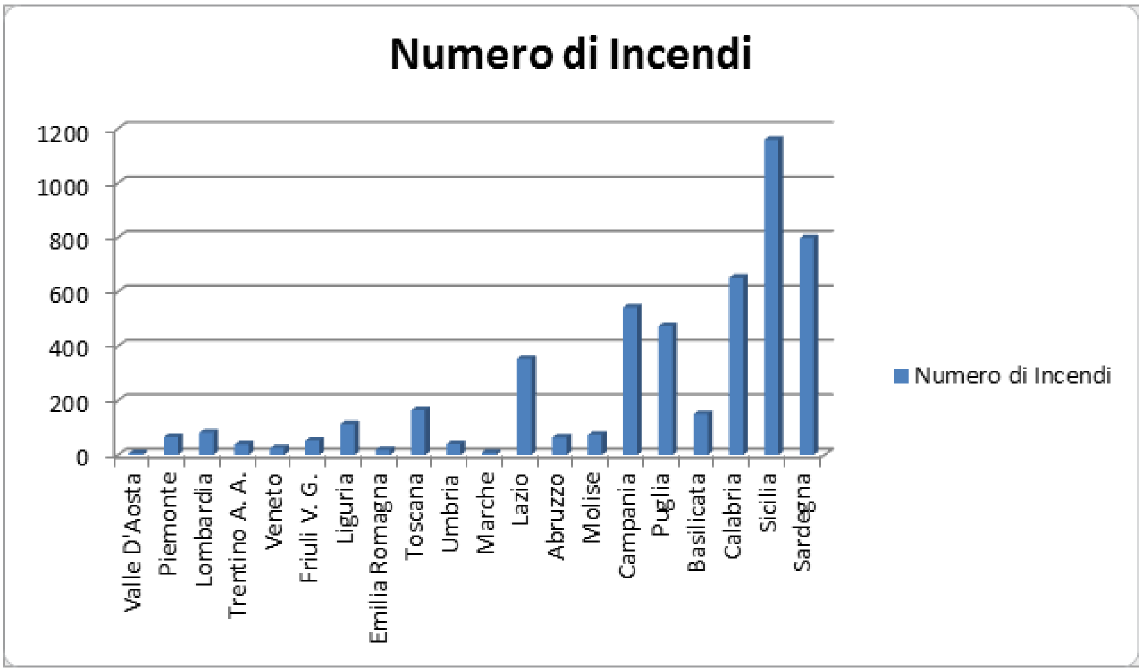

Figure 1.1: Number of Fires in Italy

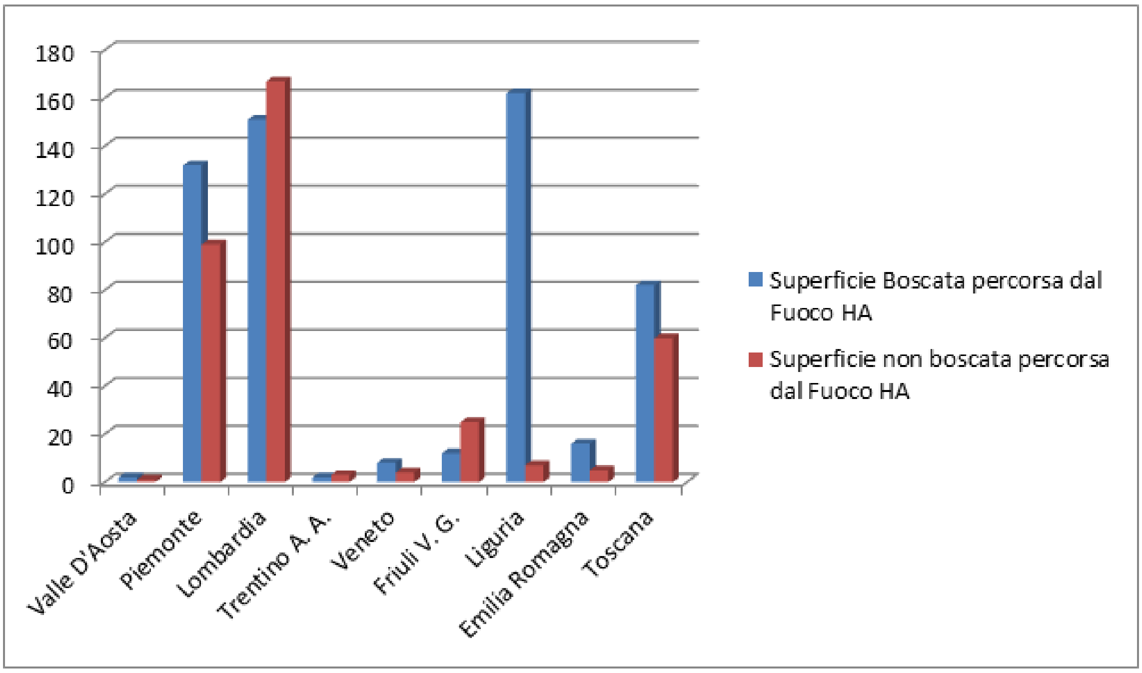

Figure 1.2: Acres of land affected by fires by region

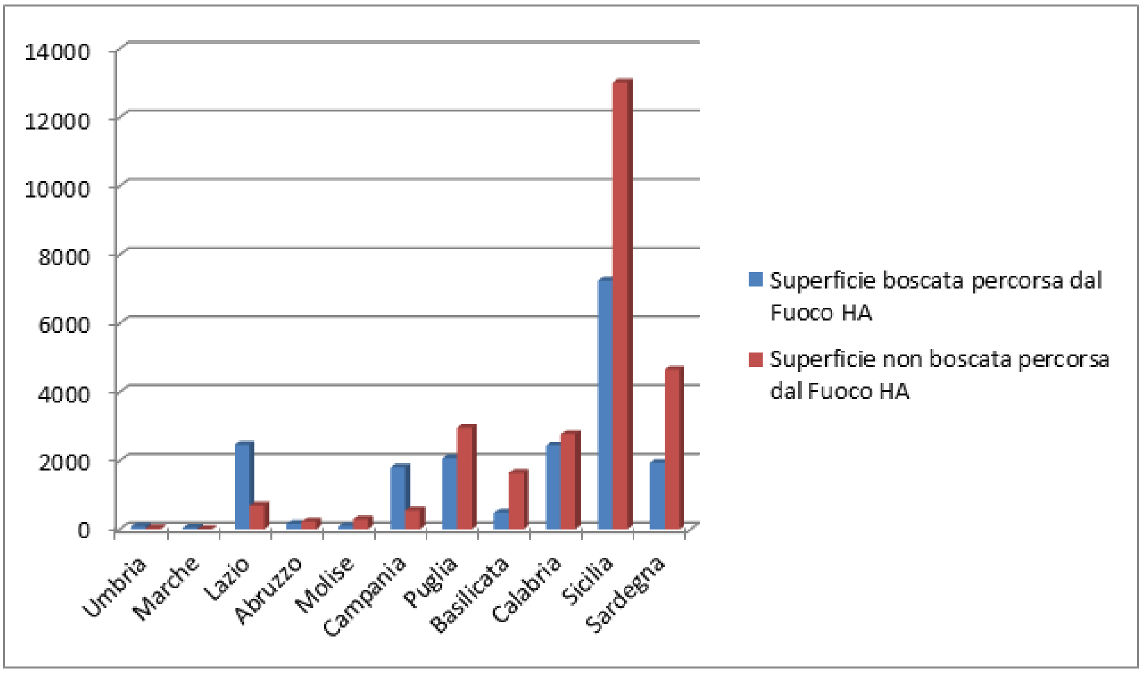

Figure 1.3: Acres of land affected by fires by region

According to static calculations has been made of the following map that shows which of the Italian regions are considered most at risk of fire. The darker areas show a higher risk than lighter areas.

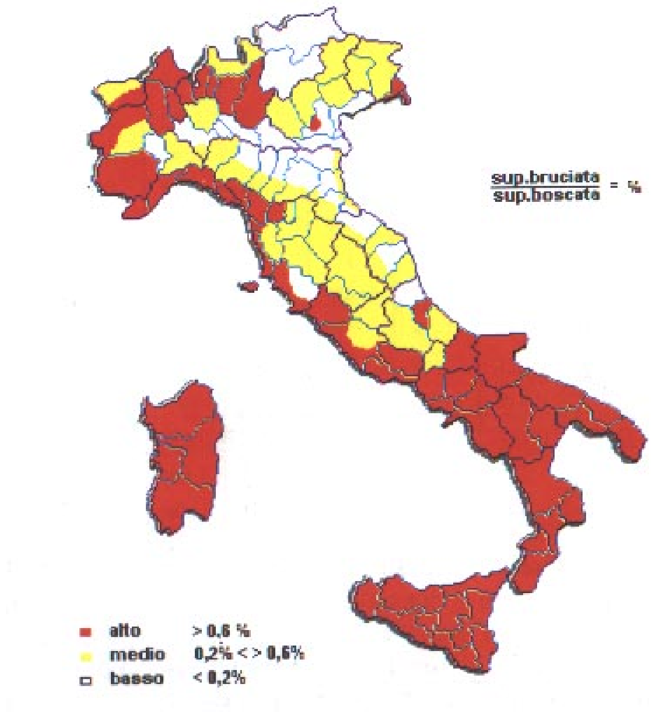

Figure 1.4: Region risk level

Provinces at high risk: where the permanent or cyclical risk of fire seriously threatens the ecological balance, the safety of persons and goods, and contributes to the acceleration of the processes of desertification of rural areas.
Provinces in the medium risk: where the risk of fire, is not permanent or cyclical but may be a significant threat to forest ecosystems.
Low-risk: provinces are all other areas not previously considered.

1.2 Causes

Often we tend to attribute the cause of the fire to natural or accidental. But the only cause of forest fires due to the nature is lightning. The ignition is an accidental event and can be caused by shards of glass that act as a lens through sunlight; this phenomenon, even if technically possible, it is highly unlikely. The same applies to the spontaneous combustion, phenomenon in which the forest gets fired by itself in a period of extreme heat: a similar event is impossible in practice with the Mediterranean climate. From this it follows that the real causes of forest fires are due to man: to his carelessness and recklessness, or its voluntary action of a speculative nature. The action of the man who causes a bushfire can be:

- FRAUDULENT: that is, with the intention of causing the fire (pyromania, revenge, destructive intent)
- WRONGFUL: that is involuntary, because due to inattention or insufficient security measures. Negligent actions are the most frequent causes of fire, among them we must unfortunately note:
  - The lighting of fires in the vicinity or in the woods without the necessary precautions.
  - The storage and burning of waste or no carelessness, neglect, abandonment which is subject to large parts of forests that make it even more vulnerable.

1.3 Consequences

The fires are not only a factor of destruction and alteration of forest ecosystems and environmental but also, as often occurs, emergency situations that interfere with the conduct of human activities on the territory. Alterations of the natural soil conditions produced by the passage of fire, promote, moreover, the phenomena causing instability of the slopes, in the event of heavy rain, slipping and the removal of the layer of topsoil. The months at higher risk are in summer, when drought, high temperature and strong wind evaporate the water retained by plants, leading to favorable natural conditions for fire ignition and development. The timeliness of intervention and the operating conditions are therefore, in the emergency phase, means of fundamental importance to avoid large areas of forest are destroyed in a few hours. In addition to environmental damage should be considered the costs and expenses arising from fires per year that lead to a total spending of 500 million euros.

1.4 Systems (management) in the field of forest fire prevention

1.4.1 Existing implementations

FireLess II [3]

FireLess II is a wireless sensor network developed by EnvEve SA and WSL in collaboration with the University of Turin allowing the acquisition of real time data for monitoring fuel parameters such as litter and humus moisture in specific forest stands. Fireless 2 Wireless consist of two sensors types for monitoring the litter and the humus layer respectively, where the LITTER sentry roughly corresponds to the FFMC of the on FWI-System) and the HUMUS sentry roughly corresponds to the DMC or the DC codes of the FWI according to the depth of positioning in the organic part of the upper soil. When provided of a Meteostation, the FireLess II system can be additionally combined with the implementation of fire weather indices.

System components

The system is composed by 4 main components:

- **Gateway**
- **Meteostation**(facultative)
  - Air Temperature
  - Air Humidity
  - Wind Speed
  - Wind Direction

  - Rain (mm/h)

- **LITTER Sentry**
  - Litter Humidity
  - Litter Temperature

- **HUMUS Sentry**
  - Soil Moisture

Main technical characteristics

Communication

Communication system uses two main radio bands: 433 MHz, communication between Sentries-MeteoGateway and GSM communication between MeteoGateway - OC (Operative Central). The transmission architecture uses a Collision Detect Algorithm to identify and manage data error during transmission. The OC Operative Central updates every sentry status using the transmission of heart-beat pack periodically sent by every device.

Power Supply

The gateway is provided by a solar panel for internal battery charge and a memory for 30 days data storage. A second battery is used to ensure efficiency in critical situations. Each sentry is equipped with a single battery. The operating cycle is optimized to reduce power consumption and increase the device life time (up to 5 years sentries' autonomy).

Security

The system is provided with a protection against rain and high humidity, encrypted data transmission and GPS sensor mapping.

Libelium Waspmote

Libelium designs and manufactures **hardware technology** for the implementation of **wireless sensor networks** so that system integrators, engineering and consultancy companies can implement reliable Smart Cities solutions within the minimum time. Main lines of research and development are:

- Waspmote: low consumption sensor device for the creation of wireless sensor networks that integrates more than 50 different sensors.
- Meshlium: the only multi-tech router integrating Wi-Fi mesh (2.4GHz - 5GHz), ZigBee, GPRS, GPS and Bluetooth technologies in a single unit.

Through a wireless sensor network deployed using Waspmote it is possible detect and prevent forest fires. To achieve this, it is possible to use infrared or ultraviolet sensors to detect flames, heat and gases that help to identify the molecules of chemical compounds generated during combustion (CO and CO2). The Waspmote GPS module allows the exact geolocation of the nodes to be included when sending the alarm to security forces or to the selected incidents system. After installing the WSN, the network can also acquire the daily values for temperature and relative humidity in order to determine the likelihood of a fire in each zone under surveillance. After analyzing this data it is possible to send an alarm to a mobile indicating the probability level and the area, helping to focus the efforts of reserve forces more productively.
It is possible to create a ZigBee wireless communication network in any environment that needs protecting. Waspmote is low-consumption and can be solar-powered meaning that it can operate outdoors for years.

Power Consumption

- Ibernate mode: 0,7 uA
- Deep Sleep mode: 62 uA
- Async sensor interruptions

- Sync timers interruptions

Maximum Range

- 2.4GHz - 7 km
- 900MHz - 24 km
- 868MHz - 40 km
- GPRS Quadband Module
- Bluetooth Module

Sensors Boards

- Gases (CO, CO2, CH4.)
- Temperature, liquid level
- Weight, pressure, humidity
- Luminosity, accelerometer
- Soil moisture, solar radiation

Open Source

- Open source API
- Open source Compiler
- Complete Documentation
- Source Code Samples

Chapter 2

Wireless Sensor Network

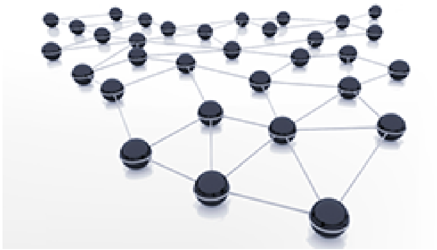

Figure 2.1: Number of Fires in Italy

The term "Wireless Sensor Network" (or WSN) indicates a particular type of network, characterized by a distributed architecture is realized by a set of autonomous electronic device scan pick up information from their surroundings and communicate with each other. Recent technological advances in microelectromechanical systems (MEMS, micro electro mechanical systems), wireless communications and digital electronics have enabled the development of small low-power devices affordable, multifunctional and able to communicate with each other via wireless technology to beam limited. These small devices, called sensor nodes, sensor nodes (in English) or mote (mainly in North America), consist of components that can detect physical quantities (position sensors, temperature, humidity, etc...), process data and communicate with each other. A sensor is commonly defined as a particular transducer is in direct interaction with the system being measured.

A network of sensors (also known as sensor networks) is a set of sensors placed near or inside the phenomenon to be observed. These small devices are mass produced and distributed, have a negligible cost of production and are characterized by very small size and weight. Each sensor has a limited reserve of energy and non-renewable and, once implemented, must work independently, which is why these devices have to constantly maintain very low power consumption, in order to have a greater life cycle. To obtain as much data as possible should be a massive deployment of sensors (in the thousands or tens of thousands) in order to have a high density (up to 20 nodes/m3) and ensure that all nodes are neighbors between them, a necessary condition so that they can communicate. One of the most common applications where you can make use of a sensor network is to monitor physical environments such as trafficking in a big city or a disaster area from data collected by an earthquake. The sensor nodes in a network have the opportunity to collaborate with each other since they are equipped with a processor on-board, with the latter, each node, instead of sending raw data to the nodes responsible for data collection, can perform simple editing and transmitting only the data requested and already processed. The communications made through short-range wireless technology, are usually asymmetrical in that the common nodes send collected information to one or more special nodes of the network nodes called sinks, which are designed to collect data and transmit it typically to a server or a computer. A communication may take place independently from the node when an event occurs, or can be induced by the sink node by sending a query to the nodes concerned. The sensor networks can be used in many applications, but the realization of the latter requires the use of techniques also used in ad hoc wireless networks. Unfortunately, many of the algorithms used in ad hoc networks are not compatible with the requirements of this type of networks. The main reasons stem from the fact that:

- The number of nodes that make up a network of sensors can be several orders of magnitude greater than the number of nodes in an ad hoc network;
- The nodes are arranged with high density;
- The nodes are subject to failure;

- The topology of a sensor network may change frequently due to failures of nodes or their mobility.
- The nodes use a broadcast communication paradigm whereas most ad hoc networks are based on a point-to-point communication;
- The nodes are limited to power supply, processing power and memory;
- The nodes do not usually have a global identifier (such as IP address in the computer);
- The nodes need tight integration with the activities of detection.

For this reason, this type of network requires algorithms designed and built specifically to manage the communication and routing data. The sensor networks can significantly improve the quality of information: for example, can provide high fidelity, can provide information in real time from hostile environments and can reduce the cost of transmitting the same information. Therefore, we assume that sensor networks can be used on a wide array of applications ranging from military to scientific, industrial, medical and domestic. The basic purpose of a sensor network is to produce over an extended period of time, a significant global information obtained from a series of local data from the individual sensors. It is important to note that the network must be built so as to ensure their integrity for a period of time that is as long as possible, in order to obtain accurate information in the event of network attack by external bodies or hardware failures. The fact that a single sensor is equipped with a small amount of energy should not prevent him to send the processed information, which will be collected and combined with information from other sensors. An important way forward is to detect the largest possible amount of local data, avoiding the inefficient transmission of data over the network. There are several possible techniques that can be used to connect the network with the outside world, in particular to convey the information it collects. In our sensory networks assume that there are special entities called sink nodes, which acts as gateway nodes, long-range and distributed near the sensors. A user specifies what information to sink node needs. The sink generates a query which is then fed into the grid. One or more answers are returned back to the sink which gathers and processes these responses before providing the final user. The sensor networks are modeled as distributed databases. The information retrieval is performed through a language similar to SQL. The flow of information depends on the mechanism of internal processing to the database. The simplest thing would be to leave the sink node, the task of collecting all information from each sensor node and the data processing. This thing is impractical for several reasons: The ack that the sink is forced to send to the various sensor nodes constitute a bottleneck when the number of sensor nodes were set to increase; The sensor nodes use to interact with each other, a wireless communication. Some nodes positioned away from the sink may not be able to communicate directly with this because of limited transmission power available. Therefore, we must model a sensor network as a distributed database where all nodes are passive, but as a distributed set of nodes that work together and where everyone has the capacity programmable active. This allows all nodes to coordinate with one another to perform a task assigned. In this way the sensor nodes become active and autonomous. The sensor nodes are scattered in an area called the sensing area. Each node within this area, has the ability to collect and route data to the sink node and finally to the end user. The sink node consists of an antenna that can light up the whole domain occupied by the sensor nodes or the sensing area. The position of the nodes within the network should not be predetermined as this allows you to use this technology in places difficult to access or rescue operations in areas devastated by a provision requiring mandatory random nodes. This means that the algorithms and protocols used in sensor networks, self-organization skills are required. The sensors, knowing the transmission characteristics of the sink, and using its signal light (which may or may not feel), can make a self-study position, thus allowing the distribution of these random in the network. The self-study position of the sensors is therefore one of the most important of these and, given the limited amount of energy that comes with a sensor, you should try to optimize the algorithms that allow a sensor to know its position, lowering his time learning, or trying to minimize the time in which a sensor must stay awake and the number of times it has to wake up.

2.0.2 Area monitoring

Area monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. A military example is the use of sensors to detect enemy intrusion; a civilian example is the geo-fencing of gas or oil pipelines. When the sensors detect the event being monitored (heat, pressure), the event is reported to one of the base stations, which then takes appropriate action (e.g., send a message on the internet or to a satellite). Similarly, wireless sensor networks can use a range of sensors to detect the presence of vehicles ranging from motorcycles to train cars.

2.0.3 Air pollution monitoring

Wireless sensor networks have been deployed in several cities (Stockholm, London or Brisbane) to monitor the concentration of dangerous gases for citizens.

2.0.4 Forest fires detection

A network of Sensor Nodes can be installed in a forest to control when a fire has started. The nodes will be equipped with sensors to control temperature, humidity and gases which are produced by fire in the trees or vegetation. The early detection is crucial for a successful action of the firefighters; thanks to Wireless Sensor Networks, the fire brigade will be able to know when a fire is started and how it is spreading.

2.0.5 Greenhouse monitoring

Wireless sensor networks are also used to control the temperature and humidity levels inside commercial greenhouses. When the temperature and humidity drops below specific levels, the greenhouse manager must be notified via e-mail or cell phone text message, or host systems can trigger misting systems, open vents, turn on fans, or control a wide variety of system responses.

2.0.6 Landslide detection

A landslide detection system makes use of a wireless sensor network to detect the slight movements of soil and changes in various parameters that may occur before or during a landslide. And through the data gathered it may be possible to know the occurrence of landslides long before it actually happens.

2.0.7 Industrial monitoring

Machine health monitoring

Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionalities. In wired systems, the installation of enough sensors is often limited by the cost of wiring. Previously inaccessible locations, rotating machinery, hazardous or restricted areas, and mobile assets can now be reached with wireless sensors.

Structural monitoring

Wireless sensors can be used to monitor the movement within buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc... enabling Engineering practices to monitor assets remotely without the need for costly site visits, as well as having the advantage of daily data, whereas traditionally this data was collected weekly or monthly, using physical site visits, involving either road or rail closure in some cases. It is also far more accurate than any visual inspection that would be carried out.

2.1 Designing a network of sensors

The design of a sensor network is influenced by many factors that are not only necessary for the design of the network, but in turn affect the choice of algorithms used in the network. These factors are:

2.1.1 Designing a network of sensors

In the sensor network is the possibility that some nodes are affected by the malfunction or failure of which may be due to physical damage, interference or low battery. Fault tolerance is the ability to operate a network of sensors in the event of failure by some nodes. Fault tolerance (or reliability) of a node k is modeled by a Poisson distribution and is seen as the probability of not having failures in (0, t): Rk (t) = e - λ kt λ k Where is the failure rate of sensor node k. The protocols and algorithms can be designed to ensure the level of fault tolerance required by the network. The tolerance level depends strongly on the application that uses the network of sensors (military, domestic, commercial, etc. ...).

2.1.2 Scalability

The system must be able to function even as the number of nodes (which can range from a low number of units, up to a few million sensors). The scale can also be achieved by exploiting the nature of dense sensor networks. The density of nodes in a sensor network is application dependent and can vary from a few to hundreds of nodes in a region with a diameter of less than 10m. The density, the number of nodes within the transmission range of a node, can be calculated as follows:
Where N is the number of nodes scattered in a region of area A and R is the transmission range of a node.

2.1.3 Production costs

Since a sensor network consists of a large number of nodes, the cost of a single node is very important. If the network cost is greater than using conventional sensors then the use of a sensor network is not justifiable. The cost of a sensor node should therefore be quite low (less than 1 euro). This is not a very easy goal to reach as currently only the price of a Bluetooth radio (which is considered a cheap), is at least 10 times higher than the price indicated. A node has other units such as a processor or a sampler, some applications may also be necessary to equip a node with a global positioning system (GPS). All these things lead to an increase in the cost of a sensor.

2.1.4 Environment

The sensors are placed very near or even inside the phenomenon to be observed. So, often find themselves working in remote geographical areas (eg inside of a machine at the bottom of the ocean, the ocean's surface during a tornado in a biologically or chemically contaminated, in a field of battle etc...) without human supervision. This gives an idea of the conditions under which the sensors must be able to work (if they work must withstand high pressures at the bottom of the ocean, high or low temperatures, etc.).

2.1.5 Network topology

A large number of nodes are arranged side by side, sometimes with a high density. This requires careful work to maintain the topology. Maintaining and changing the topology can be divided into three phases: Pre-deployment and deployment phase: Sensors can be either thrown or placed one by one into the environment, it can be thrown from a plane from a catapult, one by one placed by a robot or a human person. Post-deployment phase: the network topology changes are due to change in the position of the nodes, or the change of reachability of a node, the available energy in the presence of malfunction, etc. Re-deployment of additional nodes phase: additional sensor nodes can be rearranged at any time to replace malfunctioning nodes or due to the dynamic task. The addition of new nodes involves the need to reorganize the network. The high frequency of topological changes and the stringent constraint of energy-saving routing protocols require very special.

2.1.6 Hardware Constraints

A sensor node consists of four basic units: Sensing Unit: usually consists of two subunits: sensors and analog-digital converter; Computational units: it is usually associated with a small storage unit and manages the procedures that allow the cooperation of the node with other nodes of the network to complete the task assigned; Transceiver Unit: is the drive that connects the node to the network. It can be an optical device or a device to radio frequency (RF). You can also use the radio with a low duty-cycle, although at present there are problems in using these devices as much energy is consumed in firefighting operations; Energy unit: it is perhaps the most important component of a sensor node, sometimes may be supported by a unit for energy recovery (eg from solar). These are the four basic components for a sensor node.

2.1.7 Transmission medium

In a multi hop sensor network nodes can interact with each other via a wireless communication medium. You can then use radio waves. One possibility is to use the ISM bands (Industrial, Scientific and Medical), is a predefined group of bands that can be used freely in many countries. Most sensors currently on the market make use of an RF circuit. Another possible way to communicate between nodes is via infrared. The communication with the infrared does not require permits or licenses and protected from interference. The infrared-based transceivers are cheaper and easier to build. The biggest problem we have in the use of IR is the necessity of having a direct interface between the transmitter and receiver. This makes it difficult to use in infrared sensor networks where nodes are usually randomly arranged. The choice of transmission medium is set by the application. For example, marine applications require use water as the transmission medium. It is therefore necessary to use waves that can penetrate the water.

2.1.8 Energy Consumption

A sensor has a limited source of energy. The lifetime of a sensor node depends very much on the lifetime of the battery. In a sensor network, each node has the role of both generating and receiving data, so the disappearance of a few nodes can lead to significant topological changes that may require a reorganization of the network and routing. It is for these reasons that many studies are focusing on the creation of power-aware algorithms and protocols, ie protocols that optimize energy consumption. While in mobile networks and ad hoc networks energy consumption is an important factor but not the main (which is rather the fulfillment of QoS, ie the quality of service), in sensory networks energy consumption is the primary metric for evaluate the performance: this is because the other networks you can reload or change the batteries of the nodes in the networks and sensory once the battery is low the node is considered dead. Energy consumption in a sensor node is essentially due to three main activities of the node:

- Sensing: the power to carry out the sampling depends on the application;
- Data processing: the energy spent in processing the data is very small compared to that spent on communication;
- Communication: the three factors are what require the greatest amount of energy. The communication includes both the receiving and transmitting data where the energy costs can be considered equal.

2.2 Network Topologies

A variety of network topologies are possible with IEEE 802.15.4. A network must consist of a minimum of two devices, of which one device must act as the network co-ordinator, referred to as the PAN Co-ordinator.
The possible network topologies are:

- Star topology
- Tree topology
- Mesh topology

2.2.1 Star topology

The basic type of network topology is the Star topology. A Star topology consists of a central PAN Co-ordinator surrounded by the other nodes of the network, often referred to as End Devices. Each of these nodes can communicate only with the PAN Co-ordinator. Therefore, to send a message from one node to another, the message must be sent via the Co-ordinator, which relays the message to the destination node. The application program in the Co-ordinator is responsible for relaying messages.

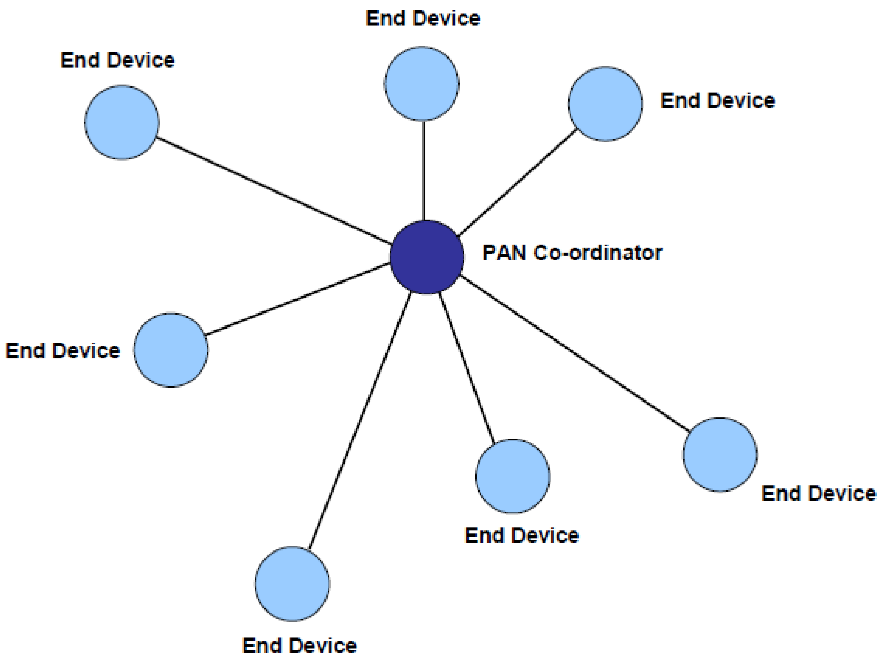

Figure 2.2: Star topology

A disadvantage of this topology is that there is no alternative route if the RF link fails between the PAN Co-ordinator and the source or target node. In addition, the PAN Co-ordinator can be a bottleneck and cause congestion.

2.2.2 Tree Topology

The Tree network topology has an implicit structure based on parent-child relationships. Each node (except the PAN Co-ordinator) has a parent. The node (including the PAN Co-ordinator) may also (but not necessarily) have one or more children. Each node can communicate only with its parent and its children (if any). Any node which is a parent acts as a local Co-ordinator for its children. The network can be visualized as a tree-like structure with the PAN Co-ordinator at the root (at the top).

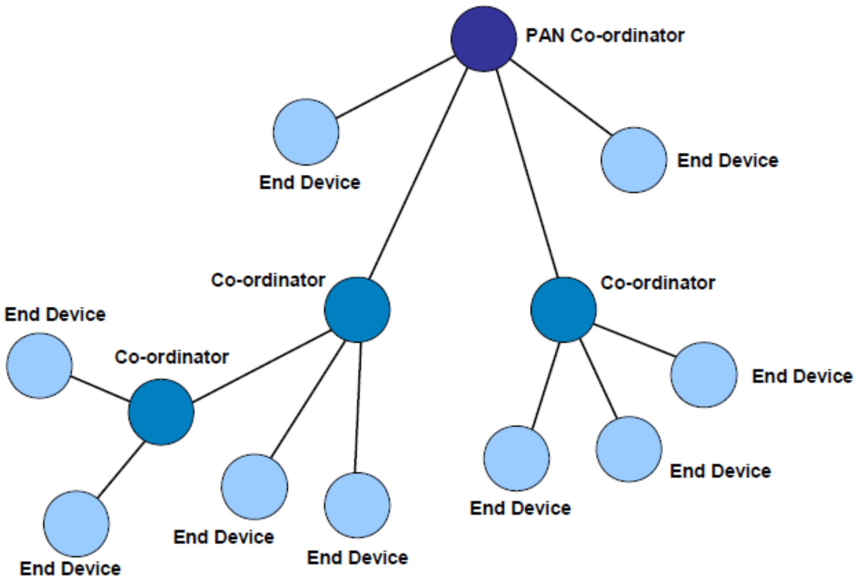

Figure 2.3: Tree topology

In a ZigBee network, which uses IEEE 802.15.4 to transport data, the local Co-ordinators are termed Routers. A special case of the Tree topology is the Cluster Tree topology, in which a given parent-children group is regarded as a cluster, each with its own cluster ID.

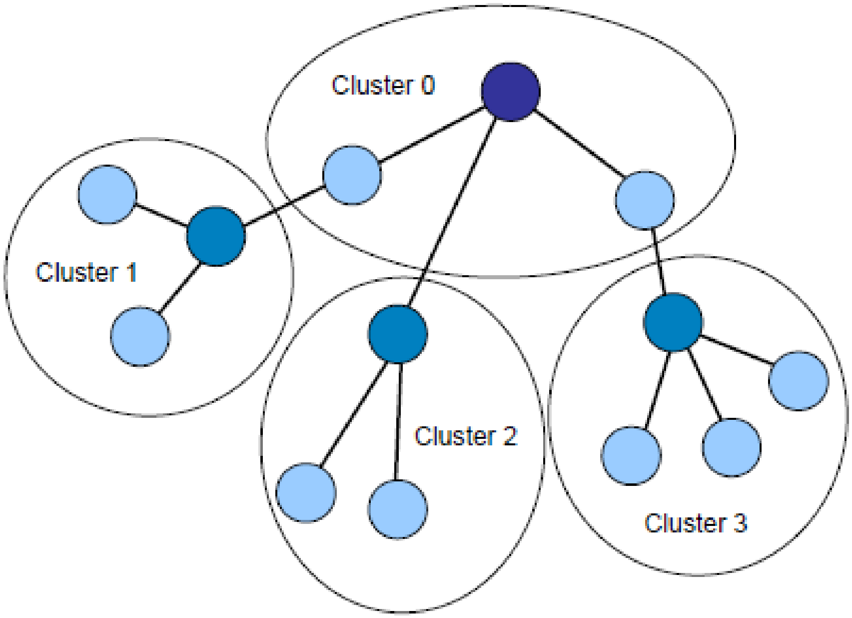

Figure 2.4: Cluster topology

2.2.3 Mesh Topology

In the Mesh network topology, all devices can be identical (except one must have the capability to act as the PAN Co-ordinator) and are deployed in an ad hoc arrangement (with no particular network structure). Some (if not all) nodes can communicate directly. Not all nodes may be within range of each other, but a message can be passed from one node to another until it reaches its final destination.

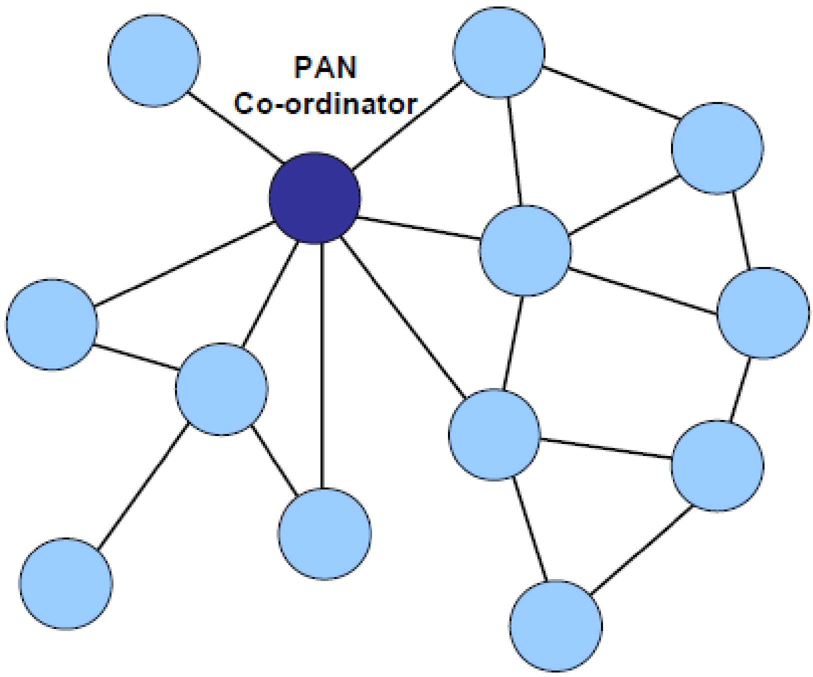

Figure 2.5: Mesh topology

Alternative routes may be available to some destinations, allowing message delivery to be maintained in the case of an RF link failure.

2.3 Hardware

A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. For example, a mercury-in-glass thermometer converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tube. A thermocoupleconverts temperature to an output voltage which can be read by a voltmeter. For accuracy, most sensors are calibrated against known standards. Sensors are used in everyday objects such as touch-sensitive elevator buttons (tactile sensor) and lamps which dim or brighten by touching the base. There are also innumerable applications for sensors of which most people are never aware. Applications include cars, machines, aerospace, medicine, manufacturing and robotics. A sensor is a device which receives and responds to a signal. A sensor's sensitivity indicates how much the sensor's output changes when the measured quantity changes. For instance, if the mercury in a thermometer moves 1 cm when the temperature changes by 1 °C, the sensitivity is 1 cm/°C (it is basically the slope Dy/Dx assuming a linear characteristic). Sensors that measure very small changes must have very high sensitivities. Sensors also have an impact on what they measure; for instance, a room temperature thermometer inserted into a hot cup of liquid cools the liquid while the liquid heats the thermometer. Sensors need to be designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages. Technological progress allows more and more sensors to be manufactured on a microscopic scale as micro sensors using MEMS technology. In most cases, a micro sensor reaches a significantly higher speed and sensitivity compared with macroscopic approaches.

2.3.1 Device Types

The nodes of an IEEE 802.15.4 based network are of the following general types, which depend on their roles in the network:

- PAN Co-ordinator: There must be one and only one PAN Co-ordinator. Its roles include:
  - Assigning a PAN ID to the network
  - Finding a suitable radio frequency for network operation
  - Assigning a short address to itself
  - Handling requests from other devices to join the network
  - Relaying messages from one node to another (but not in all topologies)
- (Local) Co-ordinator: A Tree network can have one or more local Co-ordinators (as well as a PAN Co-ordinator). Each of these Co-ordinators serves its own children and its roles include:
  - Handling requests from other devices to join the network
  - Relaying messages from one node to another
- End Device: This is a node which has an input/output function but no co-ordinating functionality. The term “End Device” is not used in the IEEE 802.15.4 standard, but is commonly used in the field.

Within the project has been chosen to use a network topology in the variant type Tree Cluster based on the need to build a large WSN, with the presence of about 500 sensors in total, capable of controlling several hectares of land under observations. It must also be adapted to the shape of the land and vegetation. This structure is best suited to the needs than others design and allows communications only between routers and nodes, and vice versa. This structure also makes easier to manage and control malfunctions of coordination devices in a manner that does not allow the loss of communication with end devices.

2.3.2 Devices used

Devices that we selected to produce although committed to address the particular needs of their areas of application, can be schematically represented by a common "core" base represented by the ability to acquire, process, store and transmit data and information. This feature plays a key role and a great help in developing a single model that can be applied and replicated on a number of areas, leading to a breakdown of the cost, easy maintenance and resolution of problems that this can also be beneficial on other production lines.

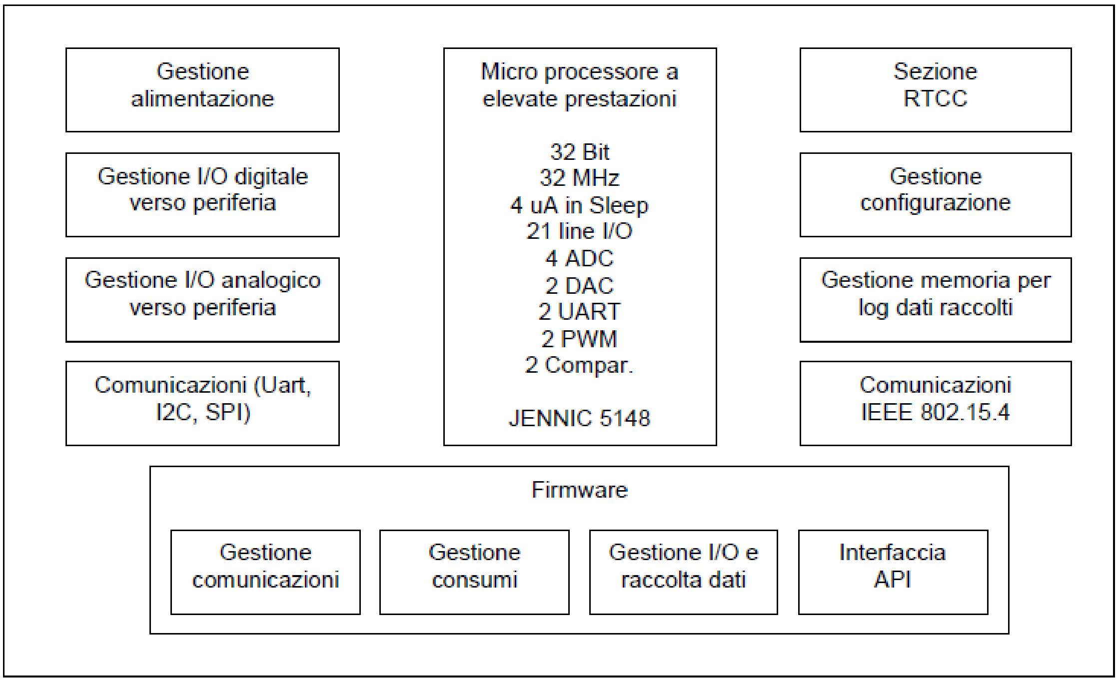

Figure 2.6: Production scheme

The study has been done in recent months has led to clarification of the powers that were in any case can be found as part of the common systems, the generation of the above scheme allows to understand the potential that can be found on a well-developed product. The choice of the micro JENNIC JN5148 [4],rests on the fact that it is the only one found so far, to be able to provide the ability to count pulses (2 to 16 1-bit or 32-bit) in the sleep state with the oscillator stopped, thus ensuring the possibility of measurements in low power state without additional external components. This feature is particularly interesting and important in solutions where required the ability to perform the measurements from the measuring devices (Energy, Gas, Water etc.). in which there is a pulse output. In contrast, the microJN5148 has a cost of about 2 / 3 times to the same capacity as a processor could be a Micro Chip or Atmel, due to the fact that also incorporates a section of IEEE 802.15.4transceiver in low power.

Another feature that has led to the choice of thatmicro is that development environment andlanguage, which unlike other brands, such as Texas Instruments, Freescale, Atmel, etc. MicroChip, the JENNIC releases in free-form usingEclipse " and language C ", therefore it is clearthat the costs of license fees related to" IDE ", and the time development of learning particular languages are completely destroyed, by contrast, the pay for such instruments, involves the lack of specialized technical support , if not in the presence large quantities handled. This setting is found in the case of the product "Arduino", that although mounts “Atmel” processors, is characterized by the possibility to be programmed via free environment and free, thanks to the work done by engineers who have made available a free low-level firmware management. At the time, anyway, the development of processors JENNIC is considered good quality and although you can’t still find that instead all those libraries can be found for the Arduino project.

Another interesting factor is that the British company JENNIC was absorbed by Electronic NXP multinational group which in turn is controlled by the multinational Philips. This represents a guarantee of continuity and development, the fact JENNIC is one of the companies, along with the Dust Network and a few others that have undertaken joint development of IP on IEEE 802.15.4 for the start of the new paradigm "of the Internet things", which will in the short term the possibility of making many devices, even in common use, to be integrated into the network.

At this point it is possible to integrate the core, with the sensors and related components necessary to perform what is necessary to meet the needs of various sectors in a way definitely easier. The optional modules are then represented by systems that can adapt communications with others instruments (Modbus, Wi-Fi, etc. Eterneth.) And interact with sensors and actuators of various types.

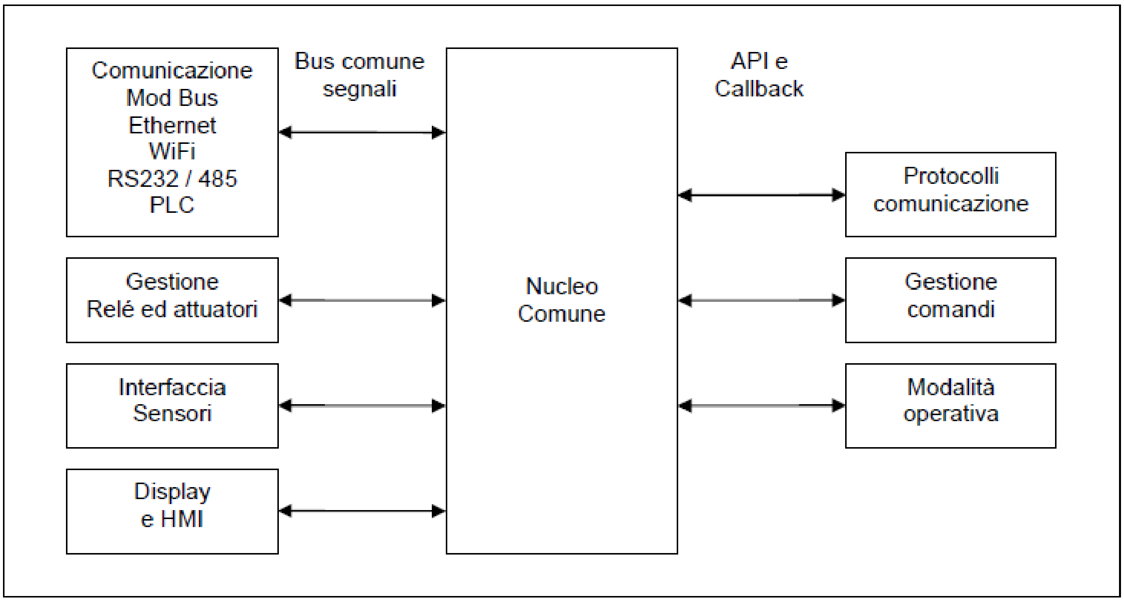

Figure 2.7: Optional modules

The easy to affix hardware extensions, will be guaranteed by the connection interface that allows a kind of layered growth throughcommon bus signals. It will be possible to create software extensions, through the development of a management firmware that through an API allowing the attachment of special features and designed for the needs of its modules. These devices are characterized by the need to operate in open places and often difficult to reach, such as forests and tracts of land, which are characterized by a high risk of fire or which are, by their shape and location, important and for which direct and indirect damages are serious entity for the community. This means that devices designed to perform this task, they must have uncommon characteristics such as:

- Reduced energy consumption
- Ease of Installation
- Reduced environmental and visual impact
- High construction features
- Easy remote management

For the first point, that relating to energy consumption, one must start from the analysis of the operational environment, in a wooded area or at least devoid of connections to power supply lines. The devices must be able to operate battery powered or alternative support systems that take advantage of natural physical conditions such as light, wind, temperature, etc... In this context, then take on particular importance is the selection of passive components and can provide ultra-low duty cycles, as well as the design and implementation of a firmware that allows for efficient and effective management of the energy available to fulfill all the tasks required. It is necessary to design the devices so that they have the ability, once placed in the desired place and activated, to be able to configure second established parameters and / or modified by remote according to the mode of operation and association within the network monitoring that takes place. The reduced environmental and visual impact are important factors in order to find the consent of the community for use of such devices. The ability and operating methods makes the device virtually no harmful emissions. The reduced consumption achieved, allow operation with non-polluting batteries and systems. The small diameters, obtained thanks to new technologies, both electronic sensors, allow placing in cases that may represent the natural elements and therefore invisible to the eye. This feature becomes particularly important in the case of applications of these systems in nature trails and / or natural parks. The construction technology and materials must be selected to be able to operate without problems in a hostile environment, where temperature, humidity, wind and other natural factors may change the operating conditions in a sudden and even violent way. At the same time the materials must meet the requirements, listed above, low environmental and visual impact. From the design point of view we find basically three types of devices:

- Central Coordinator or gateway
- Local coordinator or router
- End Node

This segmentation reflects that given by the the tree (tree topology) IEEE 802.15.4 networks, with the cluster variation. In the particular case we will not use a mesh network because is not suitable to support a large number of nodes.

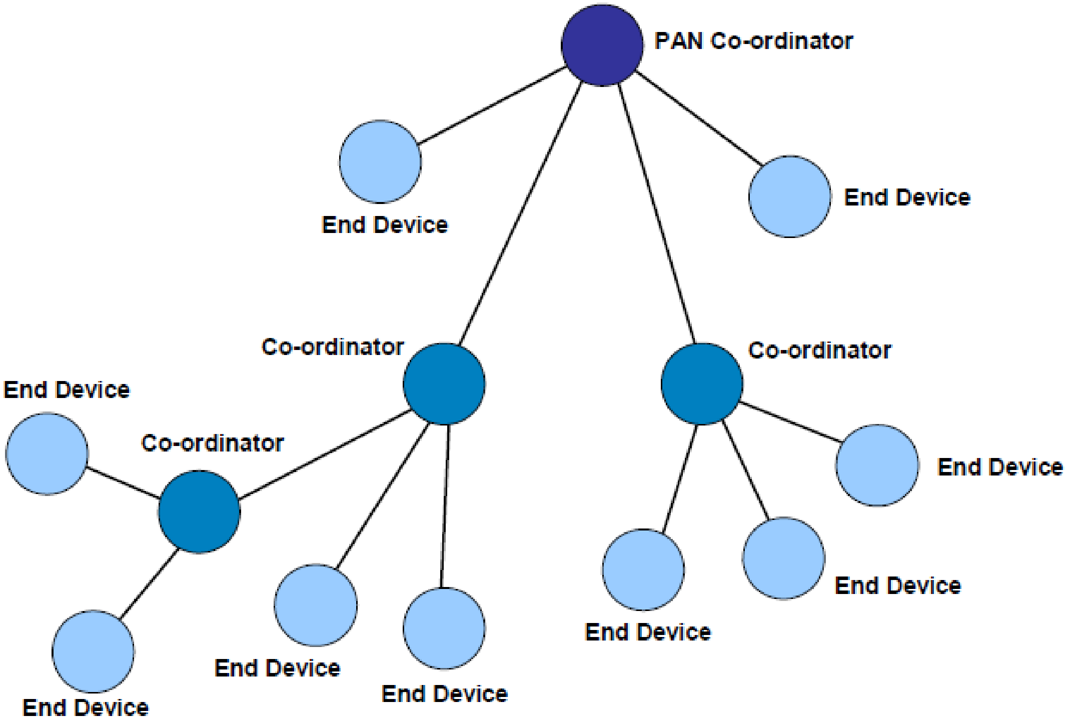

Figure 2.8: Tree Structure

As said, the choice of using a tree structure lies in some operational considerations for which there is a need to communicate in a clear direction and that is only from the gateway to nodes and vice-versa, the network must support a large number of nodes, including over 500 in order to ensure coverage on several acres of land and / or adapt to the particular form and vegetation, so we need low latency of information propagation. The variation of the cluster tree system design and installation phase, make it possible to manage automatically and /or controlled way any malfunctioning devices on local coordination, so that the nodes associated with them are not to be ousted from communications.

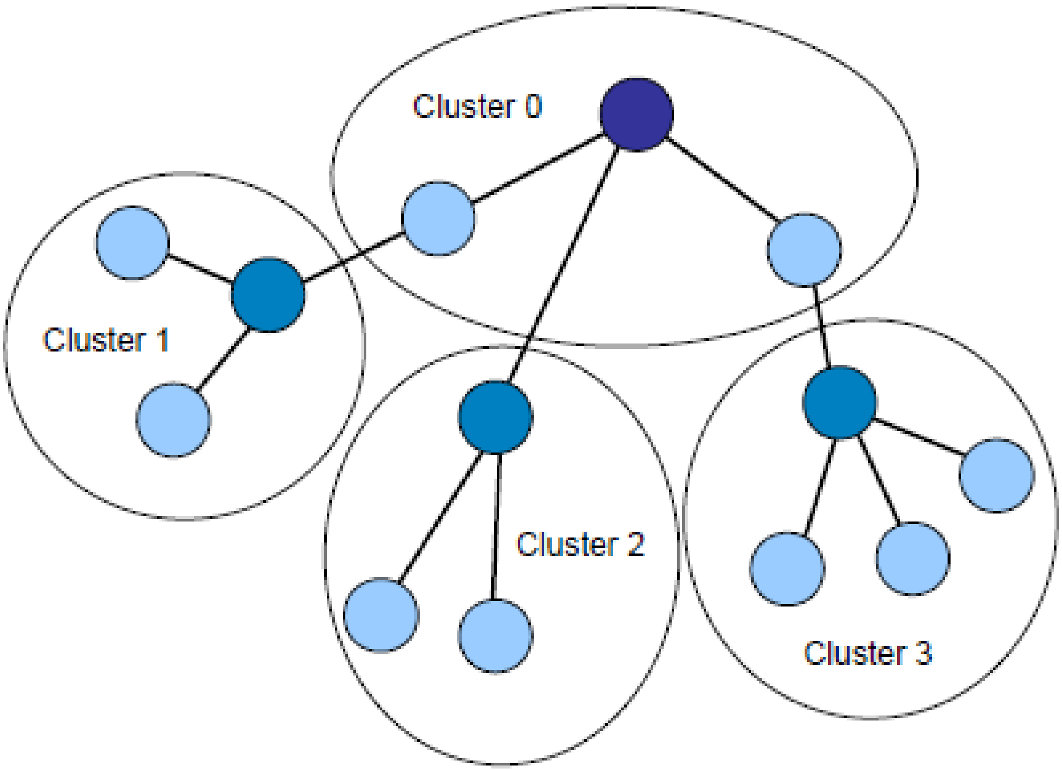

Figure 2.9: Tree Cluster Structure

No less relevant is the characteristic, given by the choice of protocol used. You can run a network of communication between groups, thus giving the possibility to extend in a continuous network of sensors and / or transfer parts of networks for improve performance in terms of transmission capacity with automatic procedures and / or manual balancing between branches.

2.3.3 Structure of communications

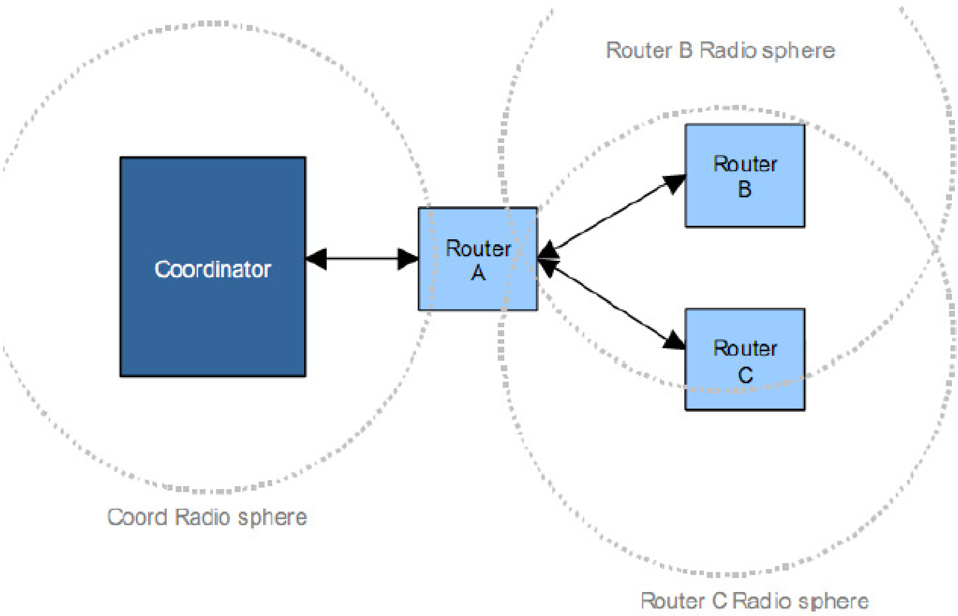

Figure 2.10: Communication Structure

As you can see from the figure, between the various local router is established a connection, in order to ensure the passage of information to and from the central coordinator, status messages allow you to determine the proper functioning of the network and the structure of the same, as shown detailed below.

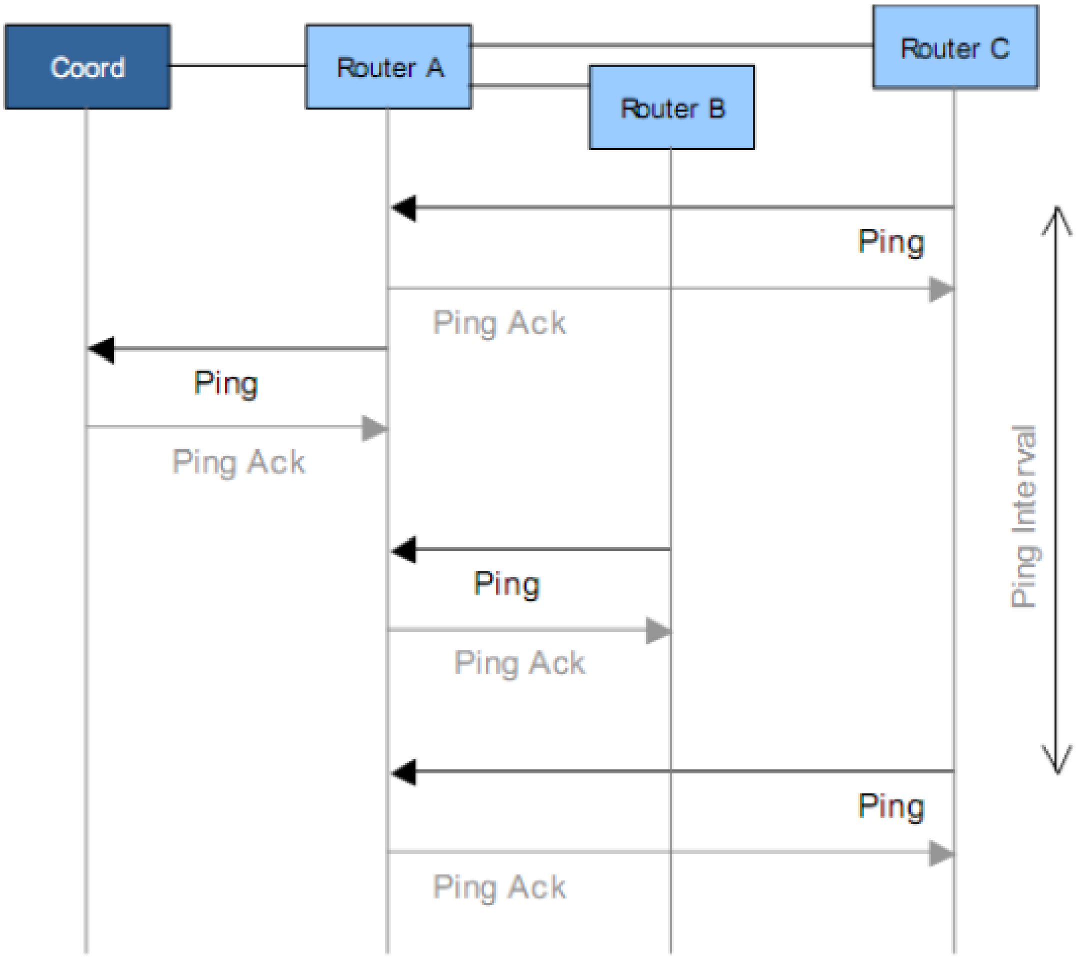

Figure 2.11: Communication Structure (2)

At the level of layer PHY and MAC, communications are characterized by confirmation of reception (ACK), that allow devices to perform variations on connection in case of loss of communication with the "father" device. The communications related to configuration and management are made with the explicit request confirmation of receipt at the application level to ensure the correct reception of the material submitted.

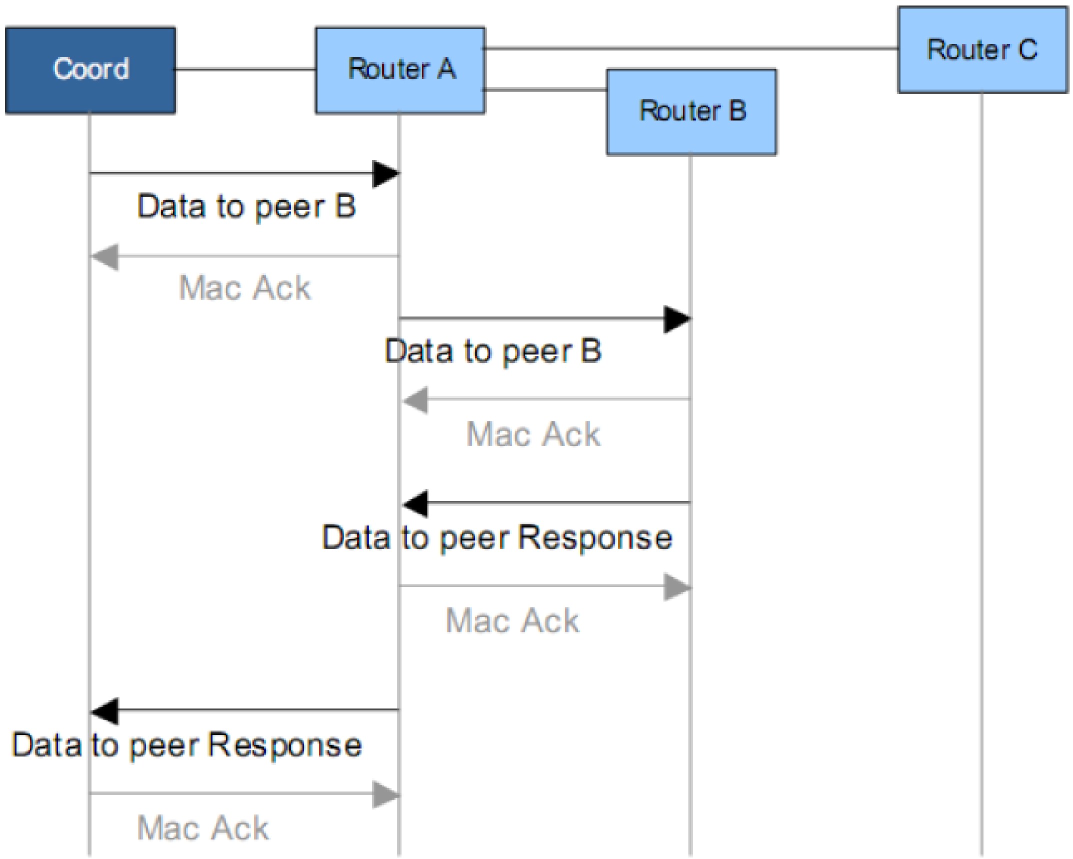

Figure 2.12: Communication Structure (3)

While the periodic messages and non-priority status, such as alarms, are sent without receiving a reply request to ensure low fuel consumption and reduced collisions in the programs that are liable, given the large number of devices, generating an increase in consumption on the nodes.

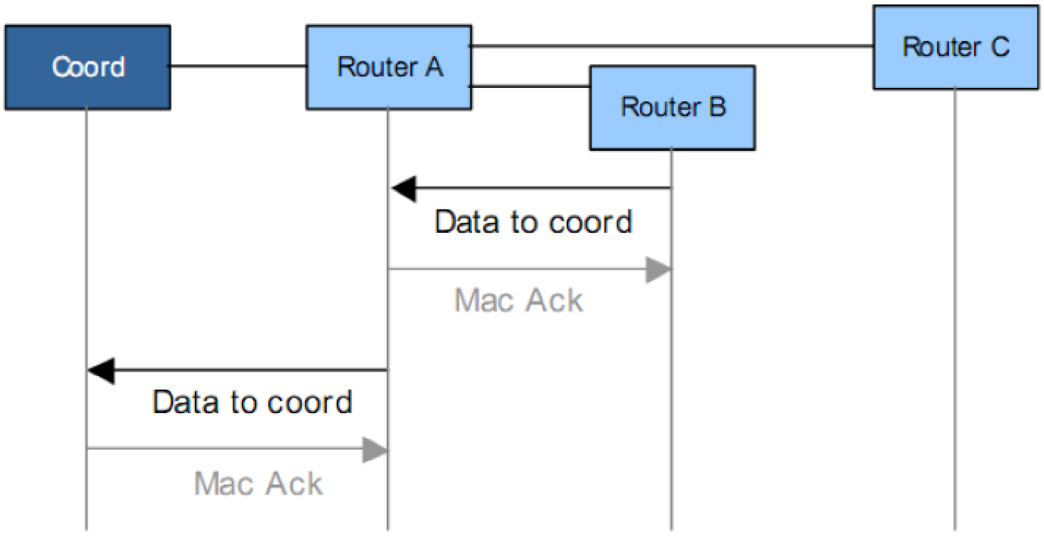

Figure 2.13: Communication Structure (4)

Finally there is the possibility to send messages to specific groups of nodes belonging to the same cluster or in broadcast mode to all nodes in the network.

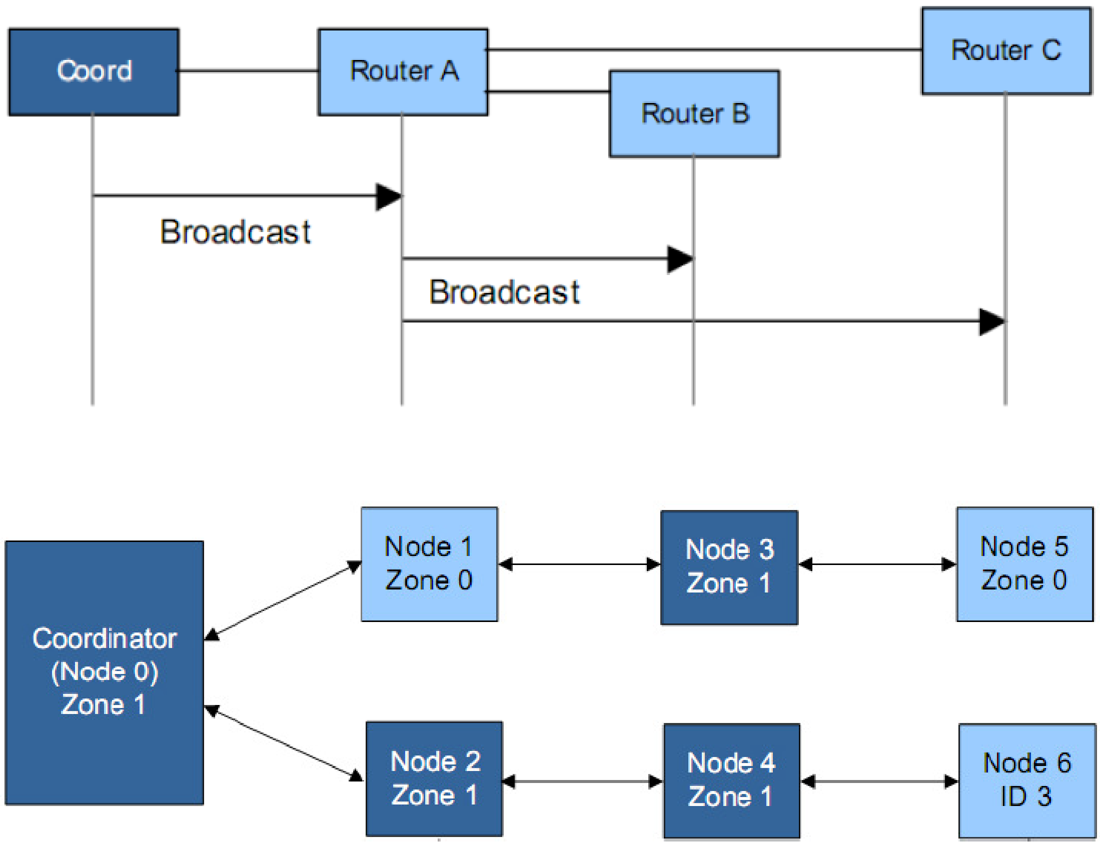

Figure 2.14: Communication Structure (5)

As shown, the system management and messaging control, are designed to take corrective automatic or controlled actions to achieve a proper balance of the branches of the network in order to effectively manage communications, this balance can be made by inserting one or more additional routers within the same network as shown later.

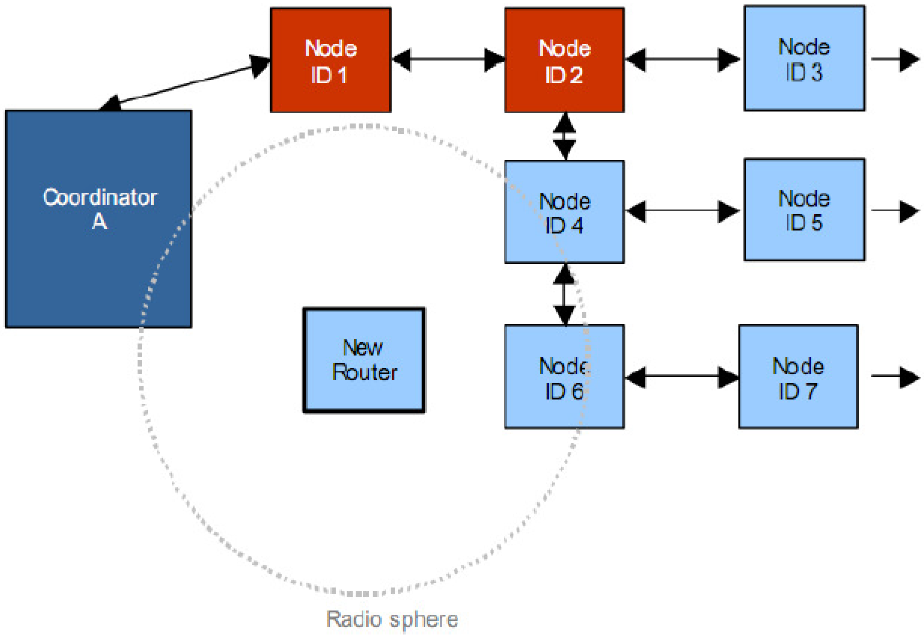

Figure 2.15: Communication Structure (6)

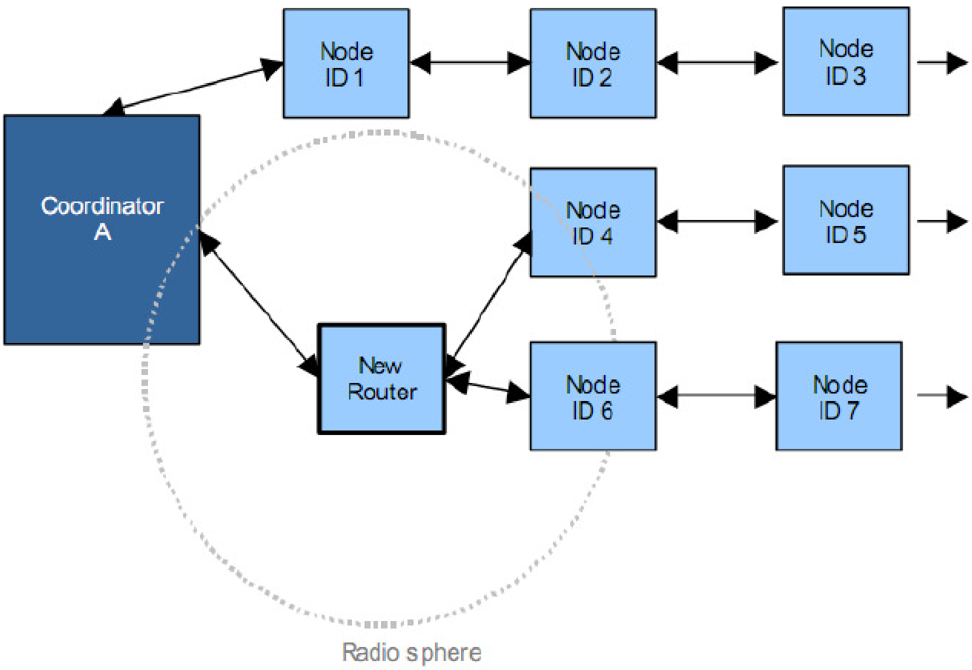

Figure 2.16: Communication Structure (7)

Similarly in case of fall of the main features of the co-ordinator or the need to move part of the network is possible determine the commissioning of a new co-ordinator and the creation of a new network, as shown below.

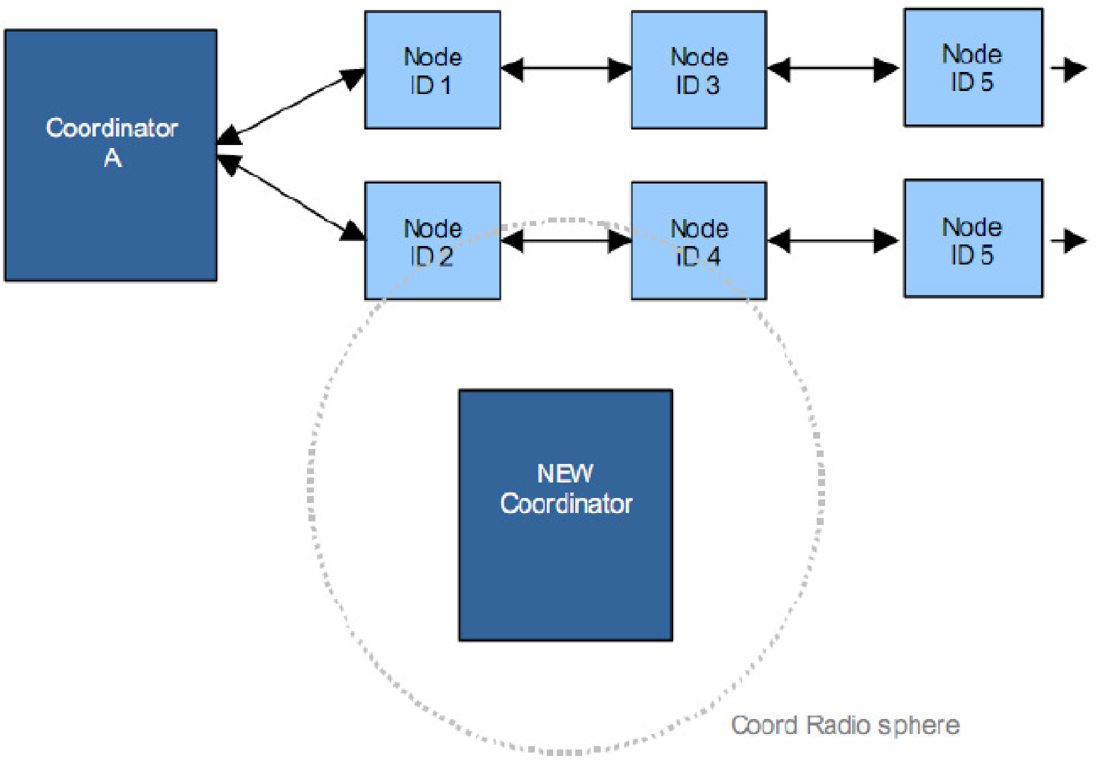

Figure 2.17: New Coordinator

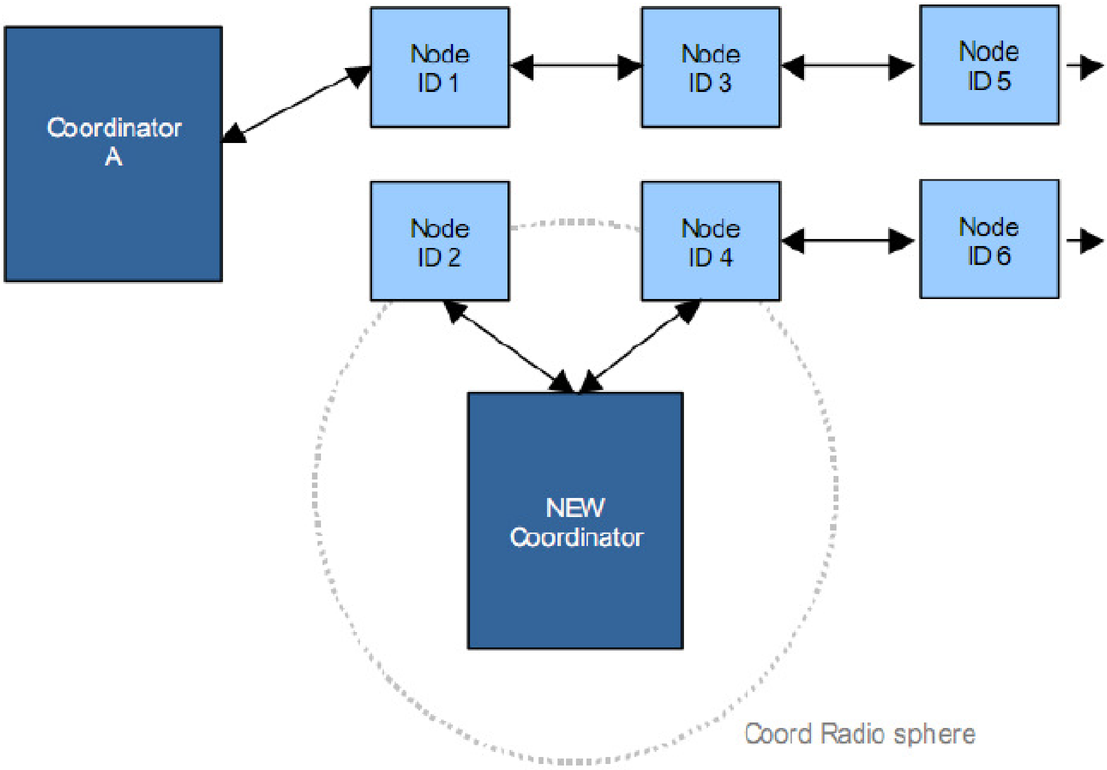

Figure 2.18: New Coordinator Association

Structure of devices

The devices to be implemented are therefore three types:

- Coordinator or central gateway
- Coordinator or the local router
- Final Node

Central coordinator

The coordinator central represents the point of birth of the sensor network. To this device is assigned the task of establishing the network and provide permission to access of the

next node in the case of network protected by authentication keys.
The Central coordinator is to all intents and purposes a router and in the specific case is also the gateway to the outside world for the passage of data acquired from the peripheral nodes and commands sent by the system of supervision and remote control.
The Central coordinator is therefore characterized by the need to be constantly maintained in a manner that the section of the transceiver and processing can handle the constant flow of traffic.
For this reason, this device will be powered by power supply system or subjected to battery charging system with energy harvesting with a photovoltaic panel , wind, heat, etc..
In order to allow the passage of the received data, the coordinator is provided with the ability to communicate with system monitoring via RS-232 serial port, WiFi, Ethernet or GSM data network, the choice of which one or more solution adopted depends on the location and the presence or absence of energy sources to ensure the operation of the modules electronic data transfer, which by their nature have very high power consumption.
The modes of communication with the central system in any case is discontinuous, in order to maintain an optimal use of energy in this situation the coordinator will always and only one that perform and establish the conditions of access to the superior system that will act as a slave is waiting for a connection.
The Central coordinator will if necessary incorporate additional sensors for environmental monitoring, although the already heavy power consumption use against doing so.
A block diagram macros and functions is the following.

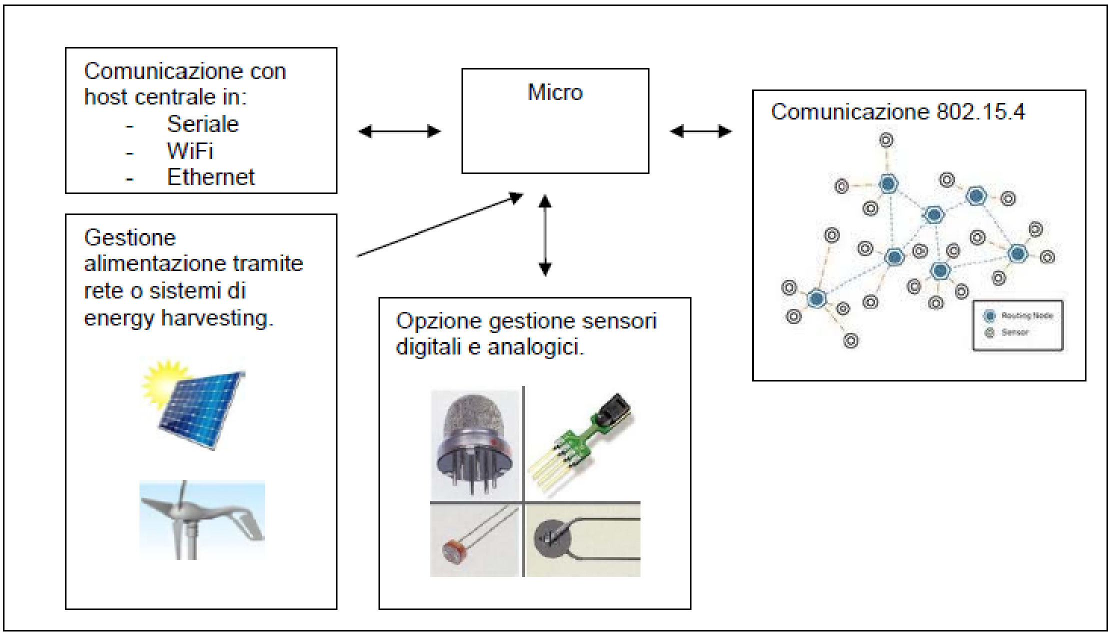

Figure 2.19: Block diagram macros and functions

The energy consumption is reported to be about 20 mAh, including sensor temperature / humidity and reading nutrition value, for this reason, the device must be equipped with at least 8 Ah battery, which ensures about 15 days of operation under normal conditions with a sufficient margin to allow charging systems to restore the appropriate level of charge. The presence of optional communication systems such as Wi-Fi or Ethernet or GSM involve a substantial increase because the type of electronics used, the activation time and duration of intervals between activations. The data available indicate an increase in consumption of about 70/80 mAh values with the activation module Wi-Fi 802.11 b / g as shown in the graph.

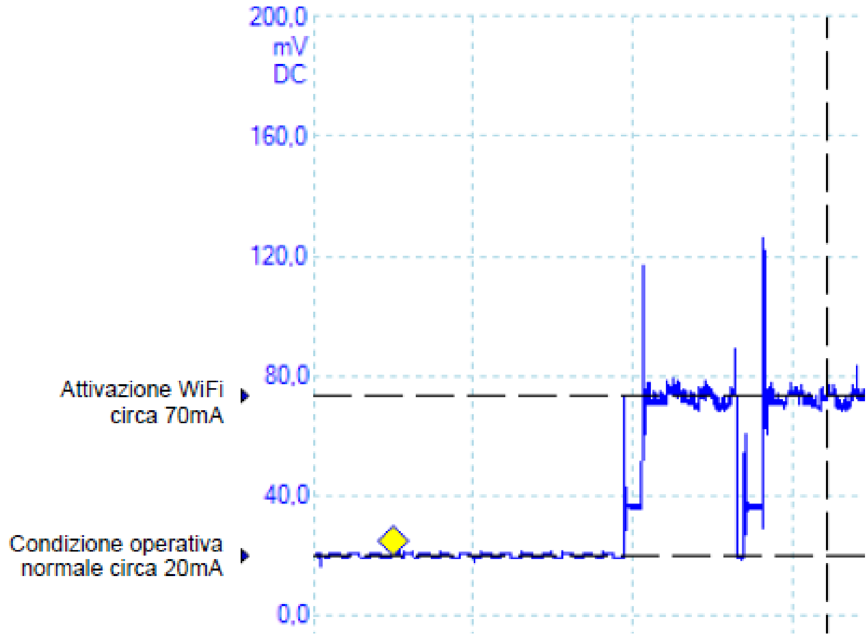

Figure 2.20: Consumption with the wi-fi active

Local Coordinator or Router

The router is responsible for managing the routing of communications to and from end nodes connected to it, basically has a similar function to coordinator with the difference that is not able to start a network and manage permissions and policies of protection. The router is characterized by the need for continuous power required for the transceiver and processing module for the management of communications. Therefore the same considerations set out for the coordinator are valid for the router except for the possible systems accessories necessary to pass data to the central host. The lack of this prerogative provides the opportunity to set up the router to handle detection sensors without affecting significantly the general consumption which remains substantially close to the value of 20 mAh.

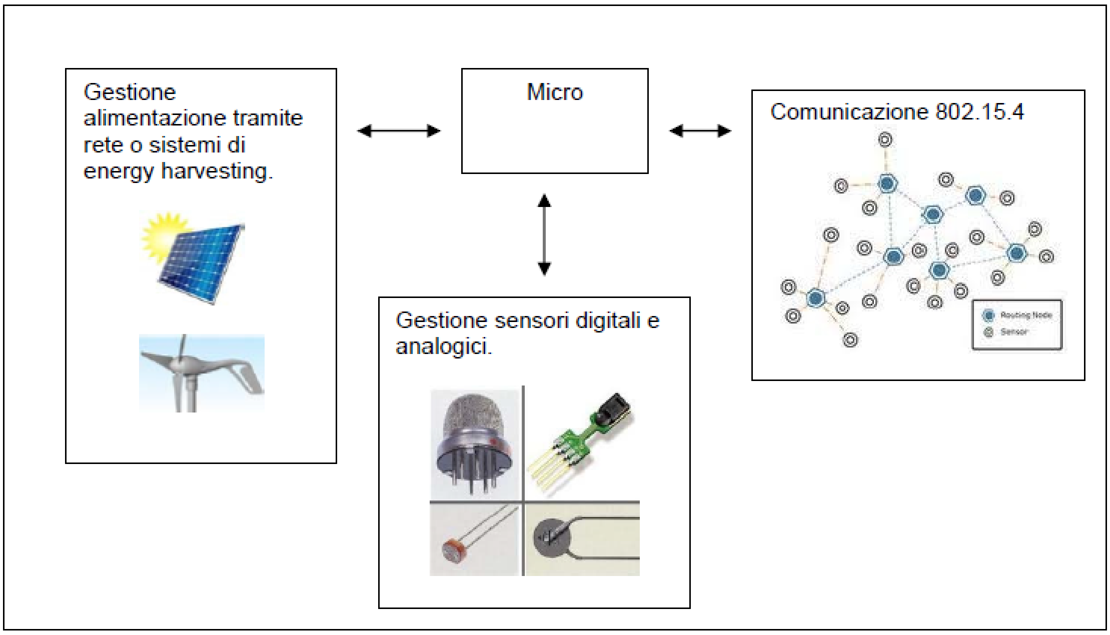

Figure 2.21: System structure of Router

End Node

The final node is the last device characterized by the ability to be supplied entirely by battery, its consumption as can be seen from the graph is basically given by the time of data acquisition and communication of information that is around 10/12 ms with an average consumption about 7 / 9 mA during operation is not all the electronics is placed in a state of "sleep" with a consumer does not cross the 50 uA including 4 uA data from the processor and the rest of the sensors and related components in use.

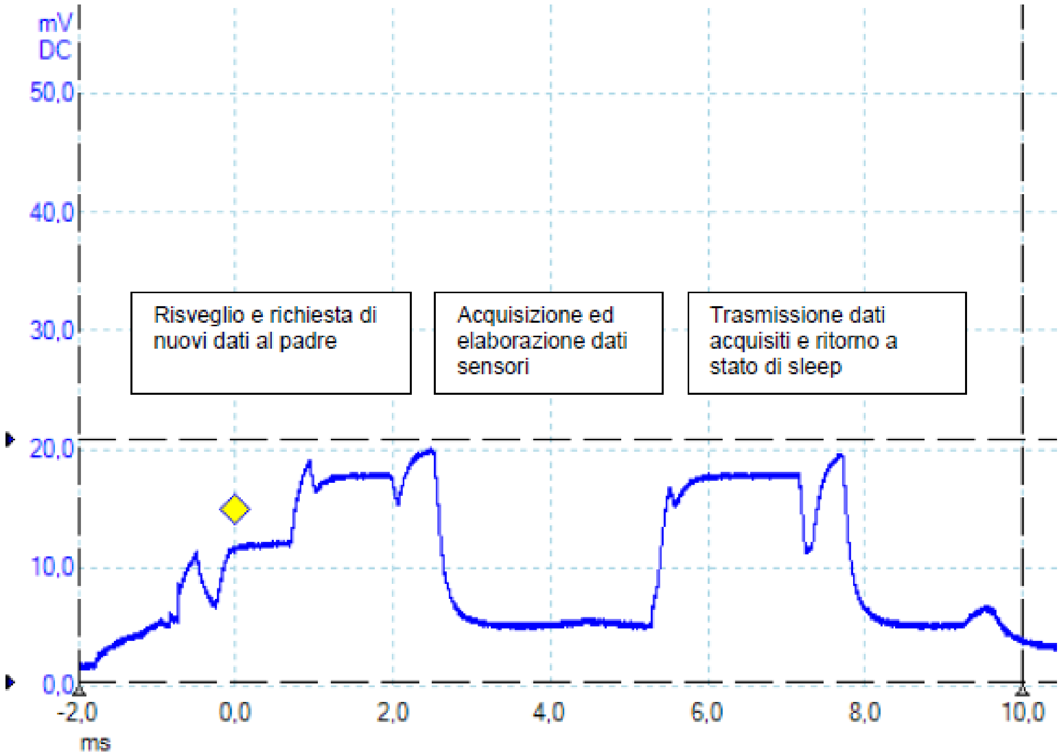

Figure 2.22: System consumption

The maximum period in which the device can remain in the sleep state can be varied from a few seconds to several minutes, during this period, the device may not receive communications from the control system and therefore it follows that the increase in this period, although it increases the life of work, penalizes the speed of communications with an increase in latency. In any case, setting a sleep time of 10s and a relative period of activities of 11 ms can be established that with an AA battery with 2.5 Ah, is possible achieve a long life of over 4 years. As the power given by section in one battery with contained capacity, it follows that the total cost appears to be contained as well as its size. The system can be schematically summarized as follows.

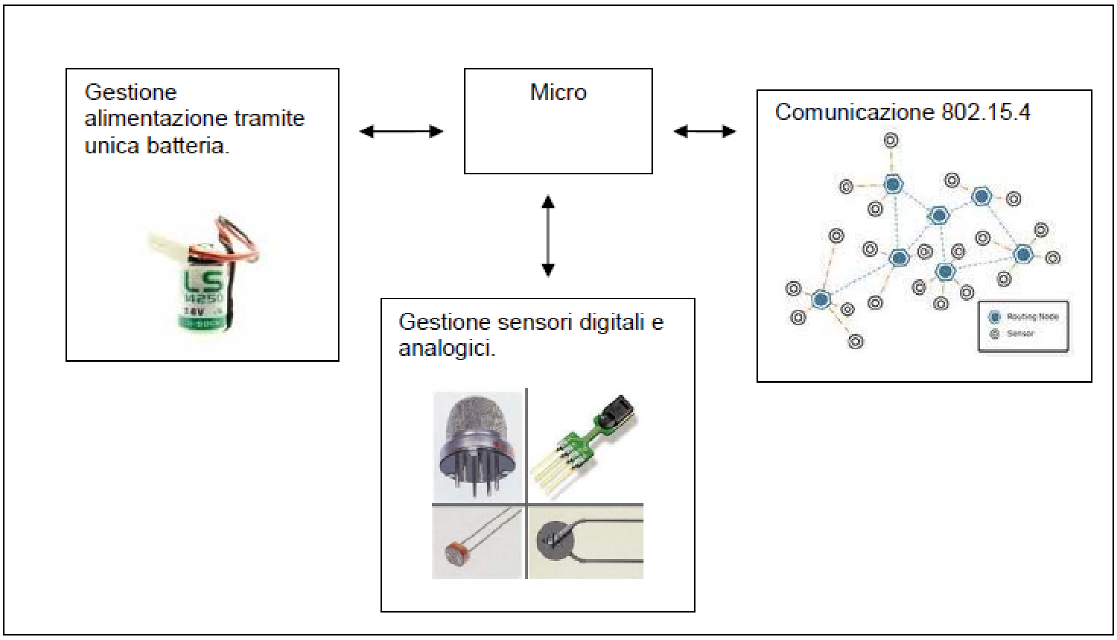

Figure 2.23: System summarization

The life of this device is not only given by the relationship between work and rest, but also by the choice of suitable sensors for this purpose. The sensors must be characterized by reduced response times to read, they can operate in digital mode, which have low operating power consumption, but in particular allow to be placed in a resting state with a consequent increase in energy savings.

Sensors

It now remains to understand which sensors can be applied in order to easily obtain data thatallow identification and assessment of the riskof fire. Given that the duration in terms ofoperating devices is characterized by the need for extremely low consumption levels, then playsan important role the proper evaluation andimplementation of such elements. Basically a sensor is a component of various kinds and forms, which has the characteristic of being able to convert a physical quantity such as temperature, humidity, light etc.. into an electrical analog or digital time-varying. For the particular application we need data collection, we come to define the variables of interest were as follows:

- Ambient temperature in air
- Relative humidity
- CO2 Concentration
- Wind speed and direction

The data collected from these measured values, set in correlation with each other may give an estimation of the hazard and / or the presenceof a fire in a given place. These measures, you can add those relating to:

- Temperature measured on the ground
- Relative humidity in the soil
- Analysis of thermal zones
- Analysis of movements with infrared

These measures can increase the degree of precision in the estimation of danger as well as being used for purposes in connection with security and control for easy identification of responsibility in case of malicious events. As mentioned the choice of sensors plays an important role in the overall system, for each family of sensors listed above there are hundreds if not thousands of possible alternatives now on the market. The choices made in this direction are addressed to achieve common type sensors do not show the ability to not have to consume more than 10 at rest and no greater than 10 mA in a state of work, these choices are also due to cost that these features behave during the purchase.

Chapter 3

Developed Software

3.1 System description and purpose

The system is made up essentially of four parts, the web pages from which the user can monitor the behavior of the devices installed on the territory under observation, the hardware part, composed of the various devices which are installed on various types of sensors, third party formed by the daemon installed on a server that acts as a conduit for communications between sensors and web pages required and the fourth is the database.
The web interface through which the user can monitor and manage the sensor data made up of five web pages created in ASP. Net, using the programming language C.
The pages show in tables the data taken directly from the database, to which the devices send the data gathered by installed sensors .
The hardware is made up of small devices that are used to collect data in an attempt to prevent and combat forest fires.
The third part of the project is represented by a "demon" that will be installed on a server listening for communications from the devices and will put them in the DB. In addition to this the "Demon" establishes a connection with the described above web pages making an intermediary between the web and the entire sensor network in case the user wants to send commands to one or more devices in the wireless sensor networks.
The attempt is to show that it is possible to implement a systematic monitoring of a wood fire in an attempt to prevent or at least to intervene quickly and prevent the spread of fire

and to avoid damage.
An attempt was made to use the best possible solutions to fit the large number of devices to be installed, however, containing costs, for purchase of devices and sensors to be installed and in terms of battery consumption of each device. This has been designed in a manner that to reduce the maintenance work during years, for lowering the overall cost of the system.

3.2 Software Design

3.2.1 Requirements

Here are the requirements for each component of the project. The requirements are expressed in FURPS + format (funcionality, Usability, Reliability, Performance and Supportability), a model for the classification of software features.

3.2.2 Requirements

Components Requierements

Server Application
Tipology: Application **Requirement:** Realization of a daemon application which manages the communication between hardware devices and software system. Usage of supplier software development kit. **Objective:** Make software and hardware to communicate **Priority:** High **State:** Done

Table 3.1: Requirements - Server Application

Asp .NET website
Tipology: Application **Requirement:**Realization of ASP .NET website providing monitoring and administration capabilities. **Objective:** Event monitoring and data administration.. **Priority:** High **State:** Done

Table 3.2: Requirements - Asp .NET website

Database
Tipology: Application **Requirement:** Realization of database for data storage. **Objective:** Data storage. **Priority:** High **State:** Done

Table 3.3: Requirements - Database

Reporting Service
Tipology: Service **Requirement:** Configuration of a reporting service. **Objective:** Generation of Reporting documentation. **Priority:** Normal **State:** Done

Table 3.4: Reporting Service

Server Application Requirements

Event Generation
Tipology: Functionality **Requirement:** Event generation at the receipt of a signal by an activator. **Objective:** Event Generation **Priority:** High **State:** Done

Table 3.5: Requirements - Event Generation

Send command to the devices
Tipology: Functionality **Requirement:** Sending the read command to the physical devices using supplied APIs. **Objective:** Communicate with the hardware. **Priority:** High **State:** Done

Table 3.6: Requirements - Send command to the devices

Event Insertion
Tipology: Functionality **Requirement:**Insertion of the generated event into the database. **Objective:** Event logging. **Priority:** High **State:** Done

Table 3.7: Requirements - Event Insertion

Event creation Response
Tipology: Functionality **Requirement:** Send back a message after the event creation. **Objective:** Give a response message to notify the success of the operation or to communicate with other devices such ligHs or alarms. **Priority:** High **State:** Done

Table 3.8: Requirements - Event creation Response

Asp. NET website Requirements

Membership Login
Tipology: Functionality **Requirement:** Relization of ASP.NET Membership Login form and components **Objective:** Site access security **Priority:** High **State:** Done

Table 3.9: Requirements - Membership Login

Real-Time Event Monitoring
Tipology: Functionality **Requirement:** Event monitoring with a dedicated page. Ajax Automatic update. **Objective:** Event monitoring. **Priority:** High **State:** Done

Table 3.10: Requirements - Real-Time Event Monitoring

Devices Administration
Tipology: Functionality **Requirement:** Devices administration page. Possibility to insert, delete or modify personnel data. **Objective:** Possibility to manage and visualize personnel information. **Priority:** Normal **State:** Done

Table 3.11: Requirements - Event Reporting

Device Administration - Quick Insert
Tipology: Functionality **Requirement:**Realize a quick device information insert, i.e. import from a different source. **Objective:** Possibility to insert device information in an easy way. **Priority:** Normal **State:** Done

Table 3.12: Requirements - Device Administration - Quick Insert

Device Amministration - Delete
Tipology: Functionality **Requirement:** Deleting an unuseful or broken device from the database. **Objective:** Delete no more useful devices. **Priority:** High **State:** Done

Table 3.13: Requirements - Device Amministration - Delete

Event Reporting
Tipology: Functionality **Requirement:** Event report generation. Microsoft reporting services. **Objective:** Need of documentation in different formats (PDF, oce etc.) **Priority:** Normal **State:** Done

Table 3.14: Requirements - Event Reporting

Report Visualization and exportation
Tipology: Functionality **Requirement:** Possibility to visualize the chosen report directly in the webpage or download it in different formats. **Objective:** Quick display of report. **Priority:** Normal **State:** Done

Table 3.15: Requirements - Report Visualization and exportation

3.2.3 Use Cases

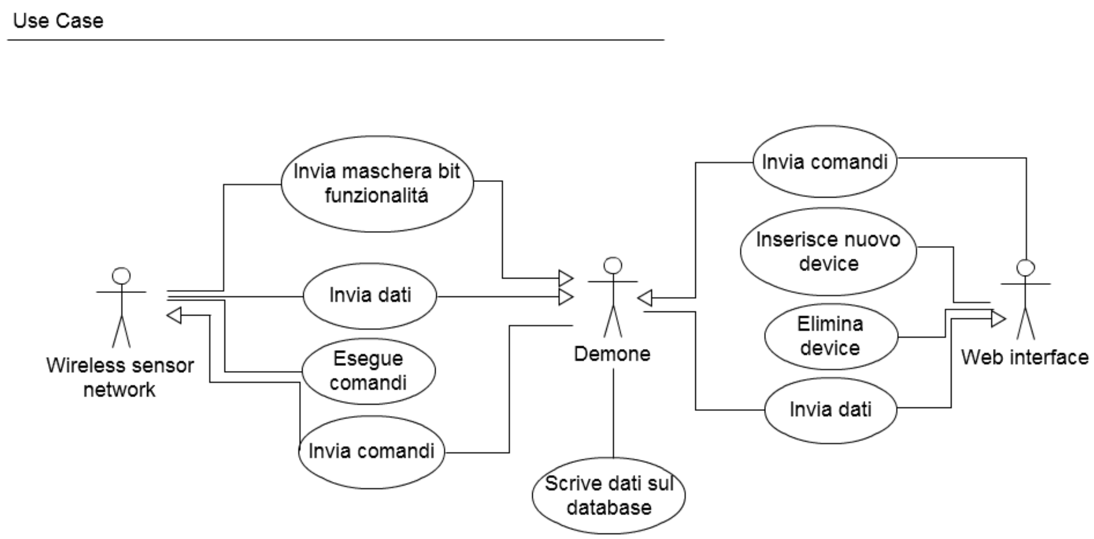

Figure 3.1: System Usecase

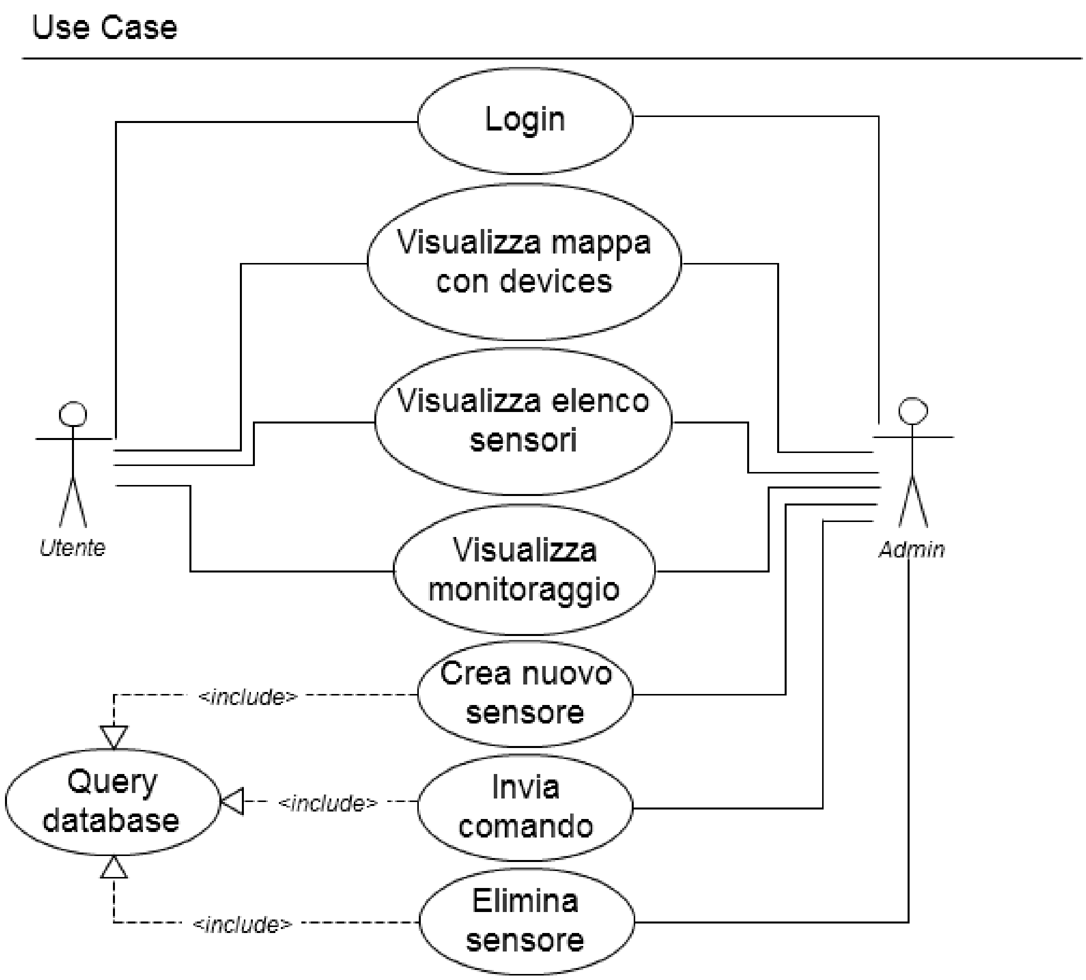

Figure 3.2: Web Site Usecase

3.2.4 Class Diagram

Web Page Class Diagram

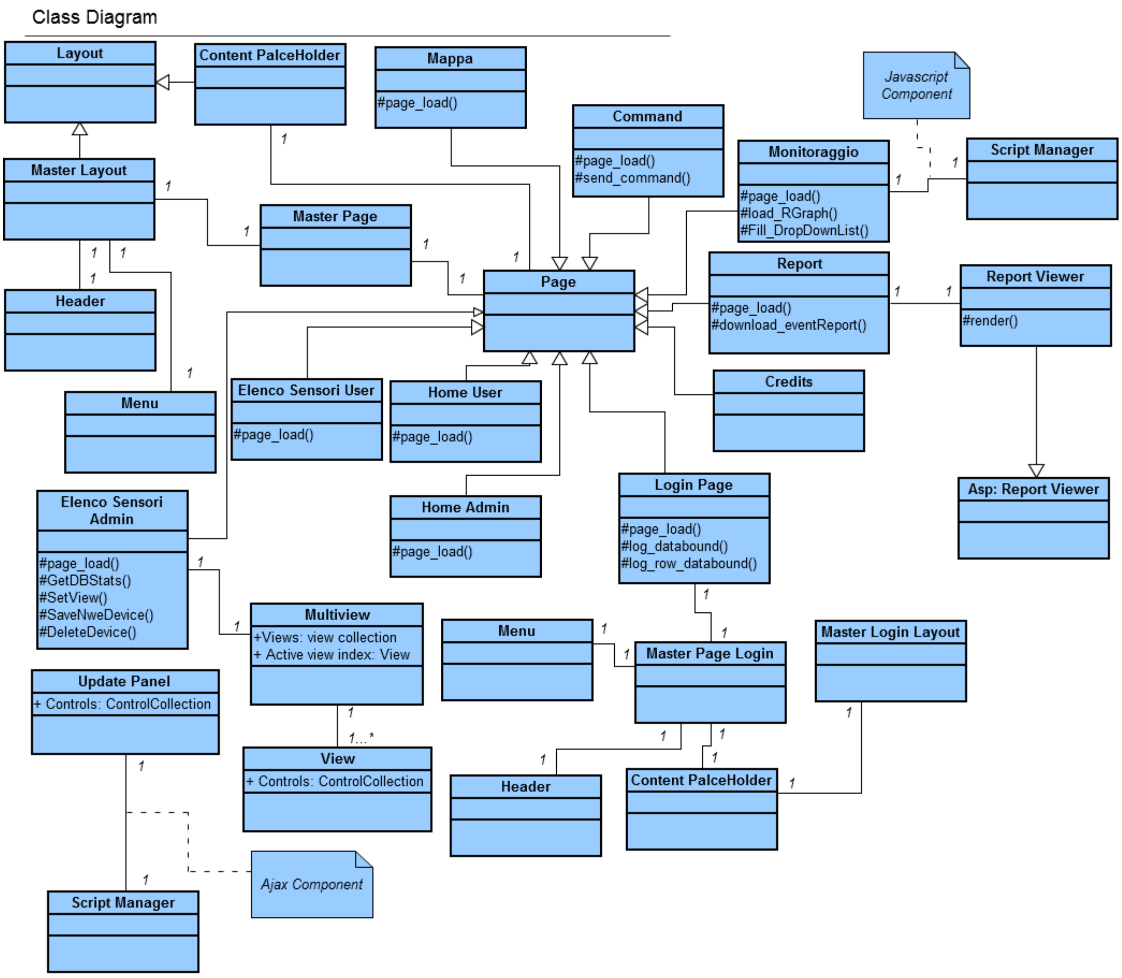

Figure 3.3: Web Site Class Diagram

Server Application Class Diagram

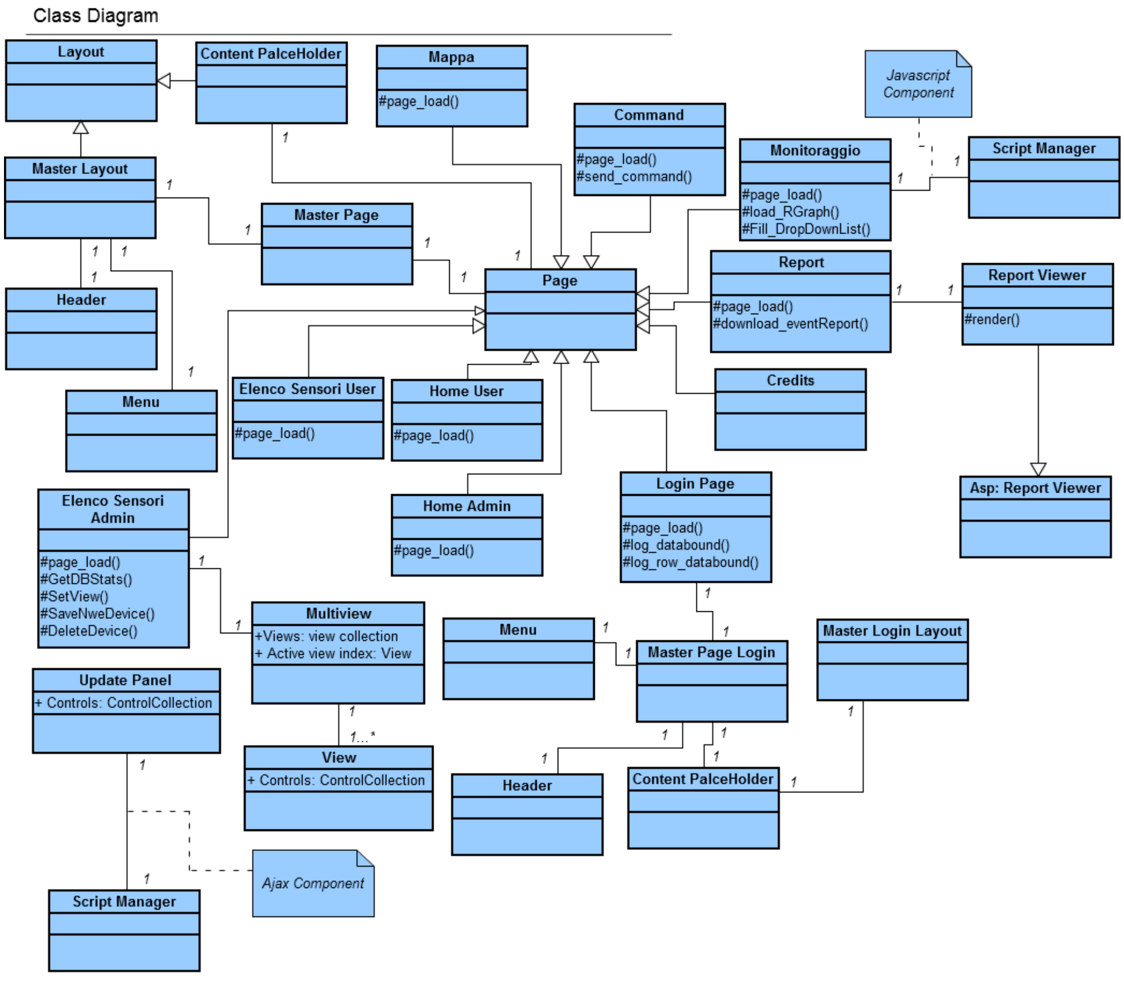

Figure 3.4: Server Application Class Diagram

3.3 System Architecture

As mentioned previously, the system consists mainly of four parts:

- The Wireless Sensor Network
- Server Application
- Database
- Web interface

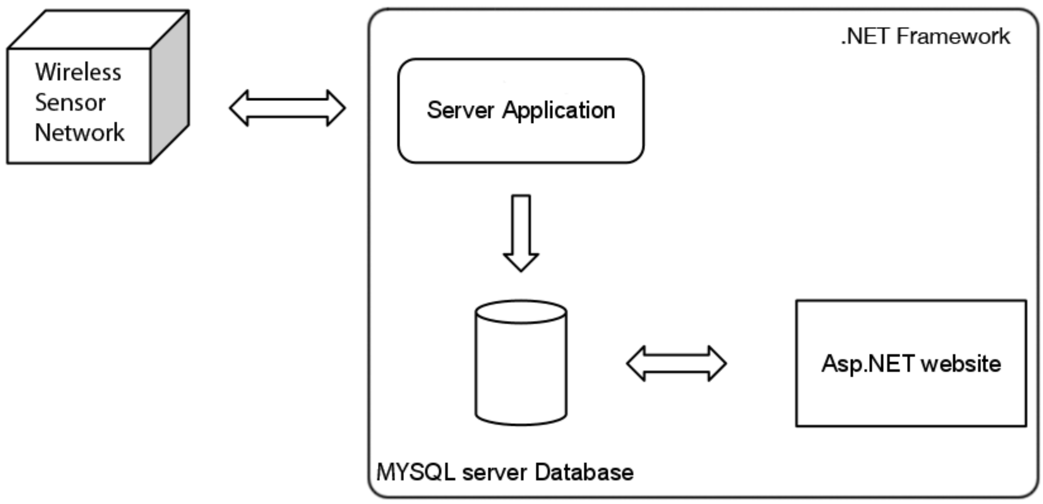

Figure 3.5: System Architecture

The sensor network is the hardware part of the whole system, and allows the effective detection of giving the field. The cross-sectional study of data collected makes it possible to determine whether there is any risk or the presence of a fire.
The daemon is a program installed on the server and acts as an intermediary between the hardware side and software side, ie between the sensor network and the graphical interface

of the system. The daemon is also responsible for writing the data in the database on the same server or remotely.
The database so represents the central part of the whole project, storing all information received from the network.
The third part is the web interface is built in a more intuitive as possible to allow the end user to monitor the condition of the forested area, and timely notice of significant changes to normal, and possibly contact the authorities used to fight fires. Obviously we have to play in real time in order to intervene as soon as possible.

Figure 3.6: System representation

3.4 Developed Software: web application

3.4.1 Technologies employed

Asp .Net / C♯ Environment

ASP.NET is a Web application framework developed and marketed by Microsoft to allow programmers to build dynamic Web sites, Web applications and Web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages. ASP.NET Web pages, known officially as Web Forms, are the main building block for application development. Web forms are contained in files with an ".aspx" extension; these files typically contain static (X)HTML markup, as well as markup defining server-side Web Controls and User Controls where the developers place all the required static and dynamic content for the Web page. Additionally, dynamic code which runs on the server can be placed in a page within a block <% – dynamic code – %>, which is similar to other Web development technologies such as PHP, JSP, and ASP. With ASP.NET Framework 2.0, Microsoft introduced a new code-behind model which allows static text to remain on the .aspx page, while dynamic code remains in an .aspx.vb or .aspx.cs or .aspx.fs file (depending on the programming language used).

Google maps

Google Maps (formerly Google Local) is a web mapping service application and technology provided by Google, free (for non-commercial use), that powers many map-based services, including the Google Maps website, Google Ride Finder, Google Transit, and maps embedded on third-party websites via the Google MapsAPI. It offers street maps, a route planner for traveling by foot, car, bike (beta) or public transport and an urban business locator for numerous countries around the world. Google Maps satellite images are not updated in real time; they are several months or years old.

Google Maps uses a close variant of the Mercator projection, so it cannot show areas around the poles. A related product is Google Earth, a stand-alone program which offers more globe-viewing features, including showing polar areas.

Google Maps API

After the success of reverse-engineered mashups such as chicagocrime.org and housingmaps.com, Google launched the Google Maps API in June 2005 to allow developers to integrate Google Maps into their websites. It is a free service, and currently does not contain ads, but Google states in their terms of use that they reserve the right to display ads in the future. By using the Google Maps API, it is possible to embed Google Maps site into an external website, on to which site specific data can be overlaid. Although initially only a JavaScript API, the Maps API has since expanded to include an API for Adobe Flash applications, a service for retrieving static map images, and web services for performing geocoding, generating driving directions, and obtaining elevation profiles. Over 350,000 web sites use the Google Maps API, making it the most heavily used web application development API. The Google Maps API is free for commercial use providing that the site on which it is being used is publicly accessible and does not charge for access, and is not generating more than 25 000 map accesses a day. Sites that do not meet these requirements can purchase Google Maps API Premier. The success of the Google Maps API has spawned a number of competing alternatives, including the Yahoo! Maps API, Bing Maps Platform, MapQuest Development Platform, and OpenLayers. In September 2011, Google announced it would discontinue a number of its products, including Google Maps API for Flash.

RGraph API

RGraph is a graphing/charting software component for websites. It uses HTML5 technologies (the new Canvas tag) which is already present in most modern web browsers to create and show graphs/charts. The graphs are created inside the web browser, removing the burden from the webserver (just as rendering HTML works) and so it can alleviate some of the burden from a web server. It supports a wide variety of graph types – from Bar charts, Line charts and Pie charts to Scatter, Rose and Radar charts. In total there's roughly 15 base chart types available with further variations possible too. It has interactive features (for example zoom, tooltips and annotations) which can improve the usability of a website/web application. It's very easy to work with and extend.

MySQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after developer Michael Widenius' daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietaryagreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, MyBB, phpBB, Drupal and other software built on theLAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches),Facebook and Twitter.

3.5 Database Structure

The database is composed of seven tables as shown in the following diagram:

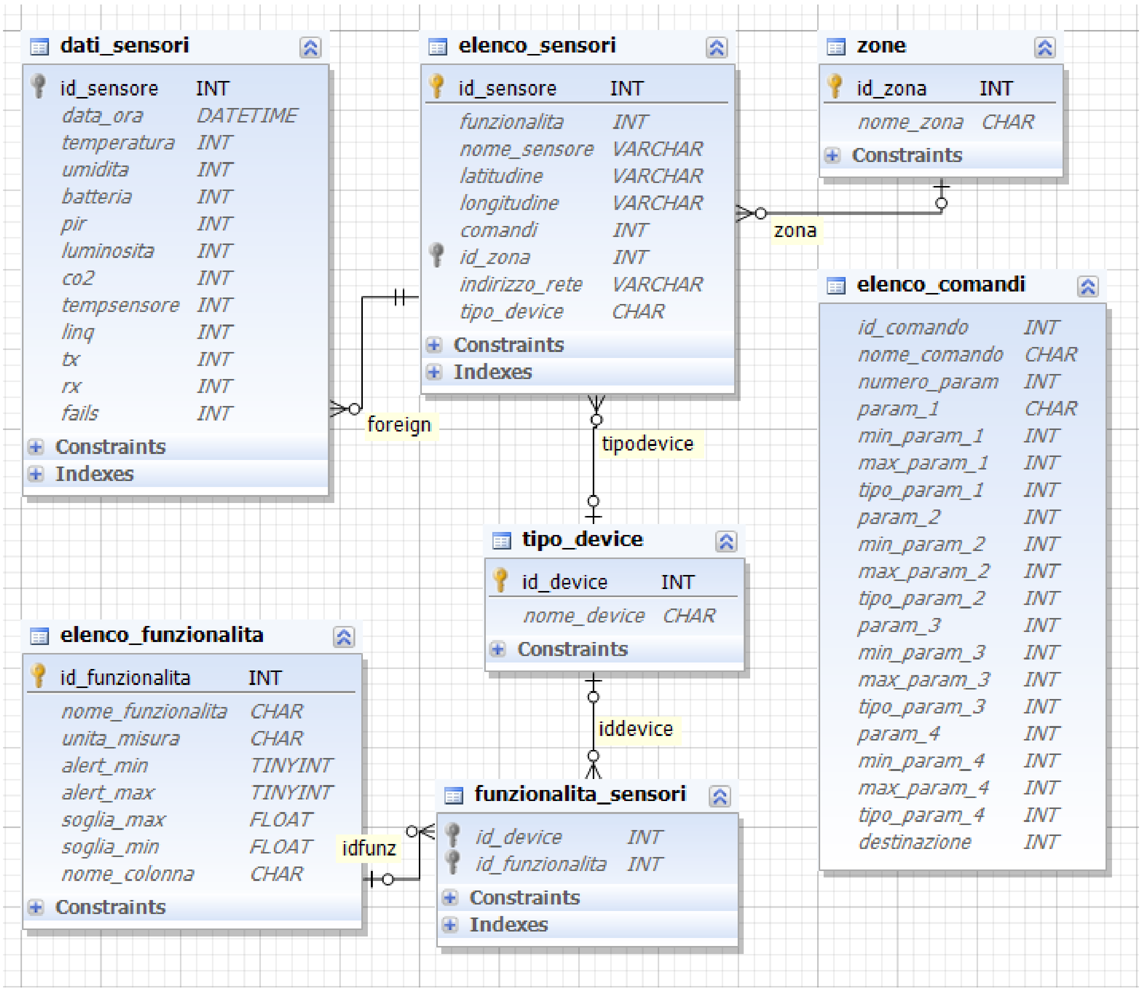

Figure 3.7: Database structure

The tables that compose the database are:

- "elenco_sensori"
- "dati_sensori"

- "tipo_device"
- "funzionalita_sensori"
- "elenco_funzionalita"
- "zone"
- "elenco_comandi"

elenco_sensori

"Elenco_sensori" is the main table in which there is a list of installed devices. identified by "id_sensore" of type INT and "nome_sensore"of type VARCHAR. The "id_sensore" is also the primary key of the table.
There are also geographic coordinates to which the device has been installed, represented by the fields "latitude" and "longitude". the field "functionality" of INT indicates which sensors are installed on the device.
"Command" is the field that indicates which commands you can issue the device. This field is INT, but the number that represents the sum of powers of two that correspond to commands in the table "elenco_comandi". For example if the table "elenco_sensori", the voice command is present the number 3 means that the devices are allowed commands "id_comando", 1 and 2 in the table present "elenco_comandi" The "id_zona" through an INT to indicate which area belongs to the device.
The zones are defined by the area where the sensor network is installed and according to the needs that arise.
"Tipo_device" is an INT that identifies the type of device on the network, including node, router, and coordinator.
the field "indirizzo_rete" is the unique MAC address of each device, useful for communication within the network.

dati_sensori

The table "dati_sensori" is connected to "elenco_sensori" by a foreign key that links the fields "id_sensori" on the two in tables"elenco_sensori" and "dati_sensori".
In addition to the "date_time" of type DATE_TIME are present all fieldscorresponding to all the sensors that were installed on several devices, such as"temperature" "humidity" "co2" "pir""brightness".
There are also the "battery" indicating the numerical value representing the battery charge installed remaining.
"Temp_sensore" is the field that indicates the temperature reached inside the device, useful to find out if there are problems of overheating of the various sensors installed.
linq tx, rx fails are all fields that indicate the general quality of communications within the network. "Linq" numerical value represents the quality of the connection between the two communicating devices. "Tx" is the total number of messages transmitted by the device while "rx" represents the number of messages received. "Fails" is the number of failures that have occurred in communications.

tipo_device

Within the table "tipo_device" there are only two fields:

- id_device
- nome_device

The first is an identifier INT that represents the primary key of table, linked by a foreign key to the "tipo_device", the second is the name given to the type of device. In this case, node, router or coordinator.

funzionalita_sensori

In this table are:

- id_device

- id_funzionalita

The first is a foreign key linked to primary key "id_device" of table "tipo_device", while the second is a foreign key linked to primary key "id_funzionalita" of table "funzionalita_sensori". With "funzionalita" we mean all types of sensors that were installed on the device such as temperature humidity, CO2 etc ...

elenco_funzionalita

The "id_funzionalita" in this table is represented by powers of two from 1. The reason for this choice is dictated by the fact that every devices just entered the network sends a 32-bit mask with some bits set to 1.These bits represent sensors that are installed. For example, in this case we have that temperature has id 1, 2 humidity, pir 4, 8brightness and co2 16.

id_funzionalita	nome_funzionalita	unita_misura	alert_min	alert_max	soglia_max	soglia_min	nome_colonna
1	temperatura	c	0	1	35	0	temperatura
2	umidita	hr	1	0	0	30	umidita
4	pir	0/1	0	0	0	0	pir
8	luminosita	lux	0	0	0	0	luminosita
16	co2	ppm	0	1	10000	0	co2

Figure 3.8: "elenco_funzionalita" table

So if the sensor will send a bitmask of this form: 00000011 you will find that the first 1 in the position corresponding to the power of two, 2^0 is equivalent to 1, while the second 1 in the position corresponding to 2^1, will be equal to 2, obtaining a value of 3 which is can see in the table "elenco_sensori".

funzionalita	id_sensore	nome_sensore	latitudine	longitudine
3	1	sensore_1	42.86212473434316	13.501424789428
3	2	sensore_2	42.85655665884564	13.501424789428
23	3	sensore_3	42.85130270352123	13.506231307983
23	4	sensore_4	42.85699708988391	13.511767387390
23	5	sensore_5	42.853221864766425	13.509149551391
23	6	sensore_6	42.85123977919203	13.515114784240

Figure 3.9: "funzionalita" field in "elenco_sensori" table

Through this mask is so possible to trace through the table "elenco_funzionalita" to the actual sensors installed on the device by performing a double query. The following is an example of the double query.

```
select s.id_sensore,
concat('select', (group_concat(f.nome_colonna)),
'from␣dati_sensori␣d␣inner␣join␣elenco_sensori␣s
on␣(s.id_sensore␣=␣d.id_sensore)
where␣s.id_sensore␣=␣',s.id_sensore) as query_string
from elenco_sensori s inner join elenco_funzionalita f
on (s.funzionalita & f.id_funzionalita > 0)
where s.id_sensore = 1
group by s.nome_sensore
```

Listing 3.1: Sample Query

in this case the sensor you want to have data is1. Within the project the number of the sensor is selected from the web page. This SQL query will likely result in a table with two columns, the first will contain the id of the selected sensor, the second will contain a second query:

```
select temperatura,umidita from dati_sensori d inner join elenco_sensori s
on (s.id_sensore = d.id_sensore)
where s.id_sensore = 1
```

Listing 3.2: Sample generated Query

This select is generated dynamically based on the functionality present on the device and stored in the table "elenco_sensori" from the "functionality", which specifies the sum of thetwo corresponding to the bit mask sent from thedevice itself.

nome_funzionalita indicates the name offunctionality such as temperature, humidity, CO2...

unita_misura is the field that contains the unit ofmeasure of the magnitude measured.
alert_min is a binary field, if set to 1 indicatesthe presence of an alarm, 0 otherwise. In this case if the minimum threshold is exceeded.
alert_max is a binary field, if set to 1 indicates the presence of an alarm, 0 otherwise. In this case the maximum threshold is exceeded.
soglia_max indicates the upper limit reached.
soglia_min is the field that represents the minimum threshold that one of thesemagnitudes can reach before triggering an alarm.

nome_colonna is the field where you enter thename of the column you want to display within a Web page.

zone

Within this table are the identifiers of the various zones of the area and the name of the same.

elenco_comandi

In addition to the id and name of the commandsthat can be sent to the devices are present-fields that have been incorporated to improvements and additions to the project, to allow the user to send different parameters for each command. This project will use a single parameter for control.

3.6 Web Pages Structure

As mentioned previously, the system is made up primarily of eight web pages to which you can access in quality of two different types of users: administrator and normal user . The obvious distinction provides different rights for each type of user: the administrator as well as monitoring the data shown in the tables will also be able to create new devices, change some information on them and send commands to individual devices. Unlike the administrator the simple user does not have the permission to act in any way in the behavior and information devices, and then only to observe the flow of data from the WSN.

3.6.1 Login

To access the system, there are two types of authorization:

- User
- Administrator

The user is enabled to the display-only pages and data from the sensor network, and the administrator in addition to all the possibilities offered to the user is also able to interact with the WSN, by sending the command to each devices, creating, through a special box, a new sensor, inputting id, name, geographical coordinates in which it is installed the device and finally the zone to which you want to attach the device. As regards the login page is necessary to insert your username and password.

Figure 3.10: Login Page

The correct data entry will allow access to the system while the incorrect entry will show an error, not allowing you to continue browsing. for the realization of the login page were made rules included in the web.config file that define the access permissions to the various pages.

```
<location path="User">
  <system.web>
      <authorization>
             <allow users="User" />
      </authorization>
  </system.web>
</location>

<location path="Admin">
 <system.web>
      <authorization>
             <allow users="Admin" />
      </authorization>
 </system.web>
</location>
```

Listing 3.3: Sample web.config Rule

You can see clearly the restrictions to each folder containing the web pages created. There are some pages in common to both types of users who can access the page without differences as "Monitoring", the "Home" page and "Credits".

3.6.2 Home Page

Once logged in, regardless of role membership, you will come to the front page of the actual system, Home.

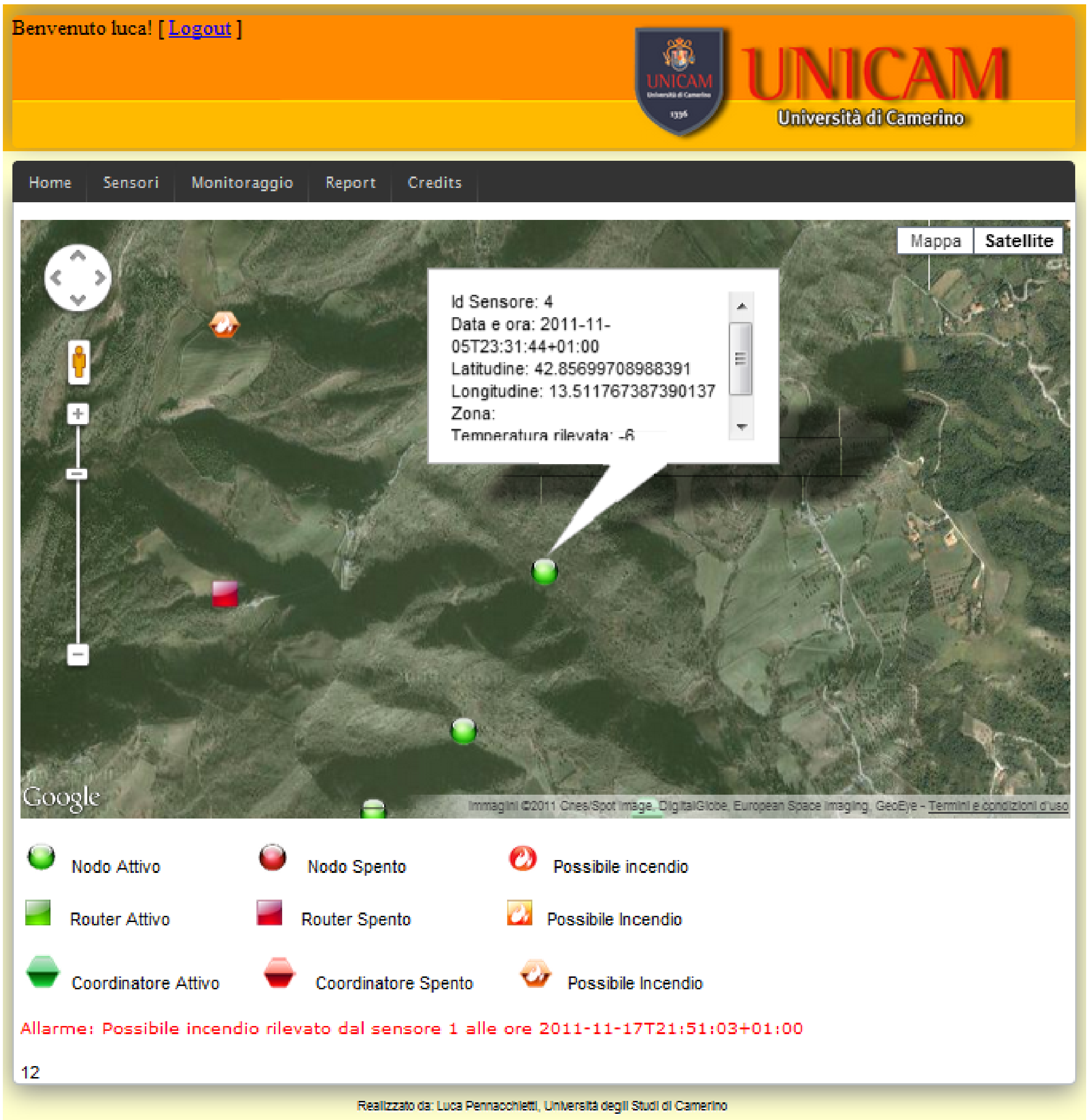

Figure 3.11: Home Page

The page contains a representation of the wireless sensor network installed in the forest monitored. has been achieved through the use of the Google Api for the construction of the map indicating the location of each node in the network. Each type of devices installed is shown by different representations, including also the state that are the nodes is represented differently.

Figure 3.12: Home Page legend

As described in the chapters devoted to the network of sensors, there are three types of device.
The node is the device that is lower in the hierarchy of the network, forming the smallest unit of data collection which then circulate to a router. the latter will have a maximum of 8 nodes will be children and the device designed to control and manage communications with the nodes.
the coordinator is the unit that will be tasked to coordinate communications between the various clusters of the network and installed on the server daemon.
It has tried to make the first page in a more intuitive as possible, which would give the user the possibility to understand immediately whether or not there is the presence of a possible fire detected by the device.
In addition to the various representations for each type of device is indicated, as mentioned before, the state where there are the various nodes. The green symbol indicates that there are no conditions of particular concern and that the data from the sensors are normal and therefore within the maximum and minimum thresholds established.
The node represented by the red symbol indicates that the battery is low or the node is already off. Where the symbol of the node should appear orange and the stylized image of

a flame in front of a detection is worthy of attention. The sensors have found that cross the values of temperature, humidity and CO2 are beyond the limits, and therefore the possible presence of a fire, and there is a need to intervene on the spot with the most timely manner possible to limit the damage and before the fire from spreading and causing damage to the environment human activities.

To click on any marker present on the map will show a popup described the selected device from where you can also access the "monitoring" the Cartesian graph loaded with data on the device.

At the bottom of the page in case you encounter a fire detections are shown a list of alarms triggered by exceeding the minimum and maximum levels of control, indicating the date and the device that detected the change in normal conditions.

3.6.3 "Elenco Sensori" Page

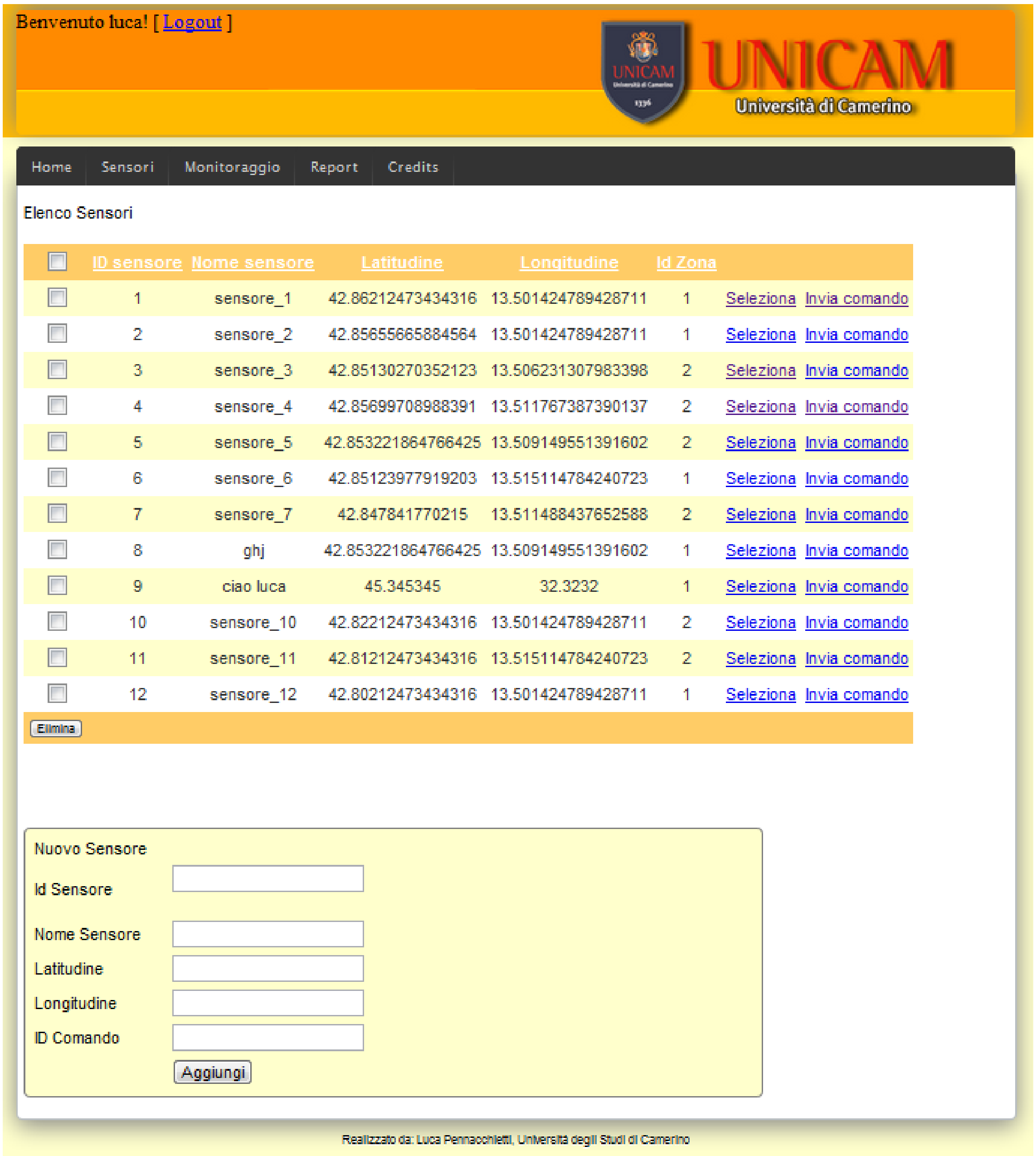

Figure 3.13: Elenco Sensori Page

"Elenco_Sensori" is the second page that is shown, it consists mainly of a gridview filled by the corresponding database table elenco_sensori. As the name of the table it includes the list of sensors sorted in increasing value according to their number. The gridview contains the following fields:

- Sensor ID
- Sensor name
- Latitude
- Longitude
- ID Zone

Sensor ID is the unique identifier of each device.
The sensor name is the name that the system administrator can assign to each device and can either numeric or not, it is useful to make the individual more easily recognizable device.
The latitude and longitude expressed the geographical coordinates of each device, expressed in the format required by the Google API, and are using them, from the pages to indicate on a map the exact location of each node in the network.
The Area ID indicates the type of area to which the device belongs.
The last two columns correspond to the Gridview command "Select" and "Send command". As regards the first it is a link which provides access to a new page, "Sensor Data" which displays additional information about the device name.
The second of two columns provided a link is called instead of "send" command, and a column is visible only if you enter the system through an authorization from an administrator, otherwise remains invisible. This link provides access to a page where you can send commands to one or more devices, a practice therefore possible to only to administrative users.
The Gridview in question has another peculiarity reserved only for administrators, that is the possibility to select via checkbox in the first column of the table, and then delete the device selected by the button "Delete".

Another feature of the page, only present if you are an administrator of the system, is displayed at the bottom. Inside a special box and are using different textbox where one can define the main characteristics of a new sensor, which will be precisely set at the database level, and which must then be effectively positioned in the forest monitored.

3.6.4 "Dati sensori" Page

The third page, accessible through links placed within the gridview in Elenco_sensori.aspx page, called "Select", allows the visualization and monitoring of data actually sent by each device in each time interval. The data are obviously sorted by date and time in order to have the most recent entries in the highest part of the gridview.

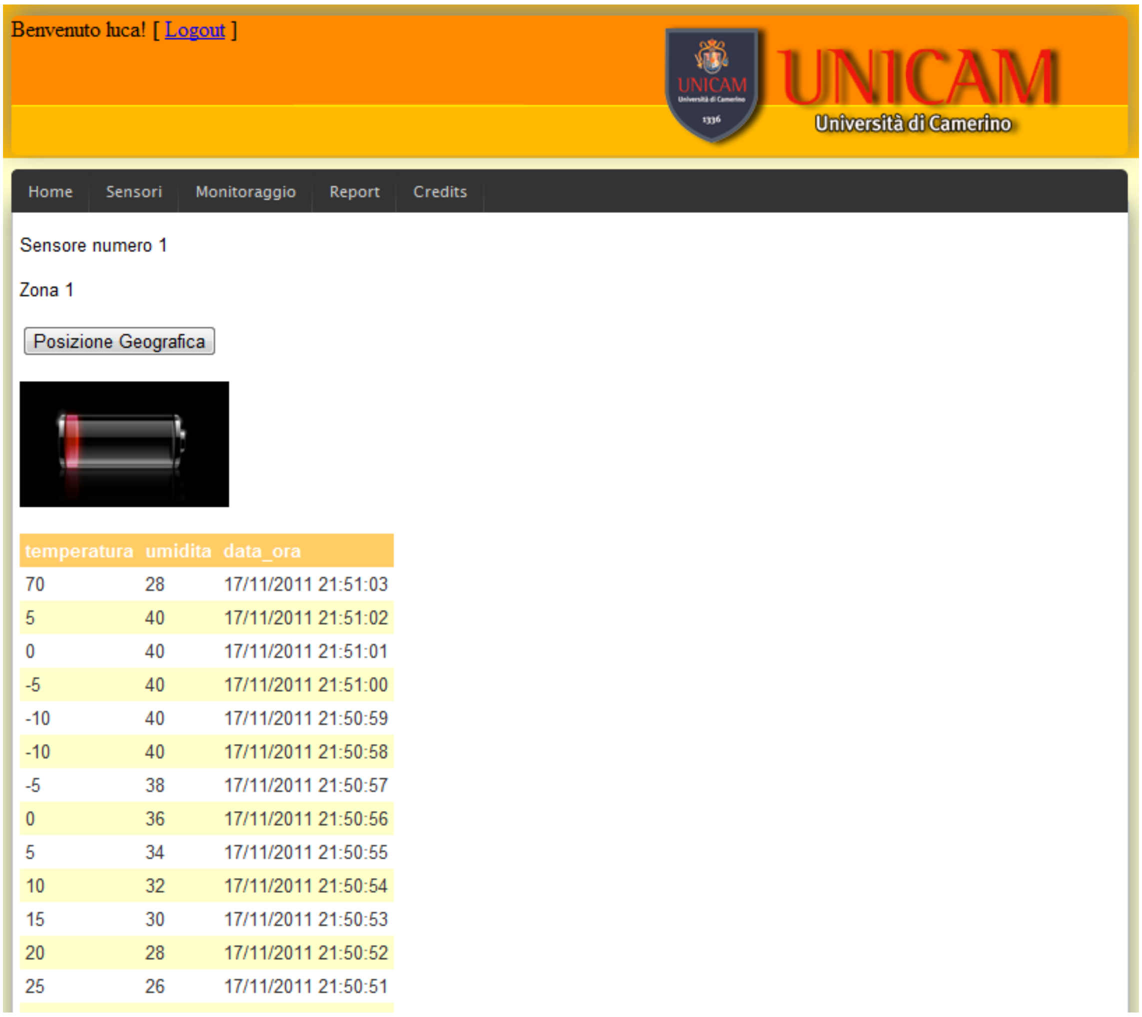

Figure 3.14: Dati Sensori Page

Each column represents the data from each sensor mounted on the device selected.
The sensors available to detect relevant variables to determine whether the presence of fire

or possibility of fire in the area under observation. These quantities range from ambient temperature to moisture detected, the brightness, the presence of more or less high levels of CO2, more information purely technical conditions of the device itself as a quality signal, between device and coordinator, the charge battery expressed in numerical value, and also expressed by a special image that is in a more intuitive level, charging the battery installed on the device.
These values depend on which of these sensors were installed on each device, this composition also characterizes the property that is defined as a functionality, which is then identified by a number.
In the upper table on the "dati_sensori.aspx". This is another link called "Geographical Location". It provides access through to the next page and query string data is sent to the next page. these data are the id of the sensor that was previously selected page "Home.aspx" coordinates, ie latitude and longitude, the numerical value of the battery charge remaining on the device, and the id of the area belongs to the device in question.

3.6.5 "Mappa" Page

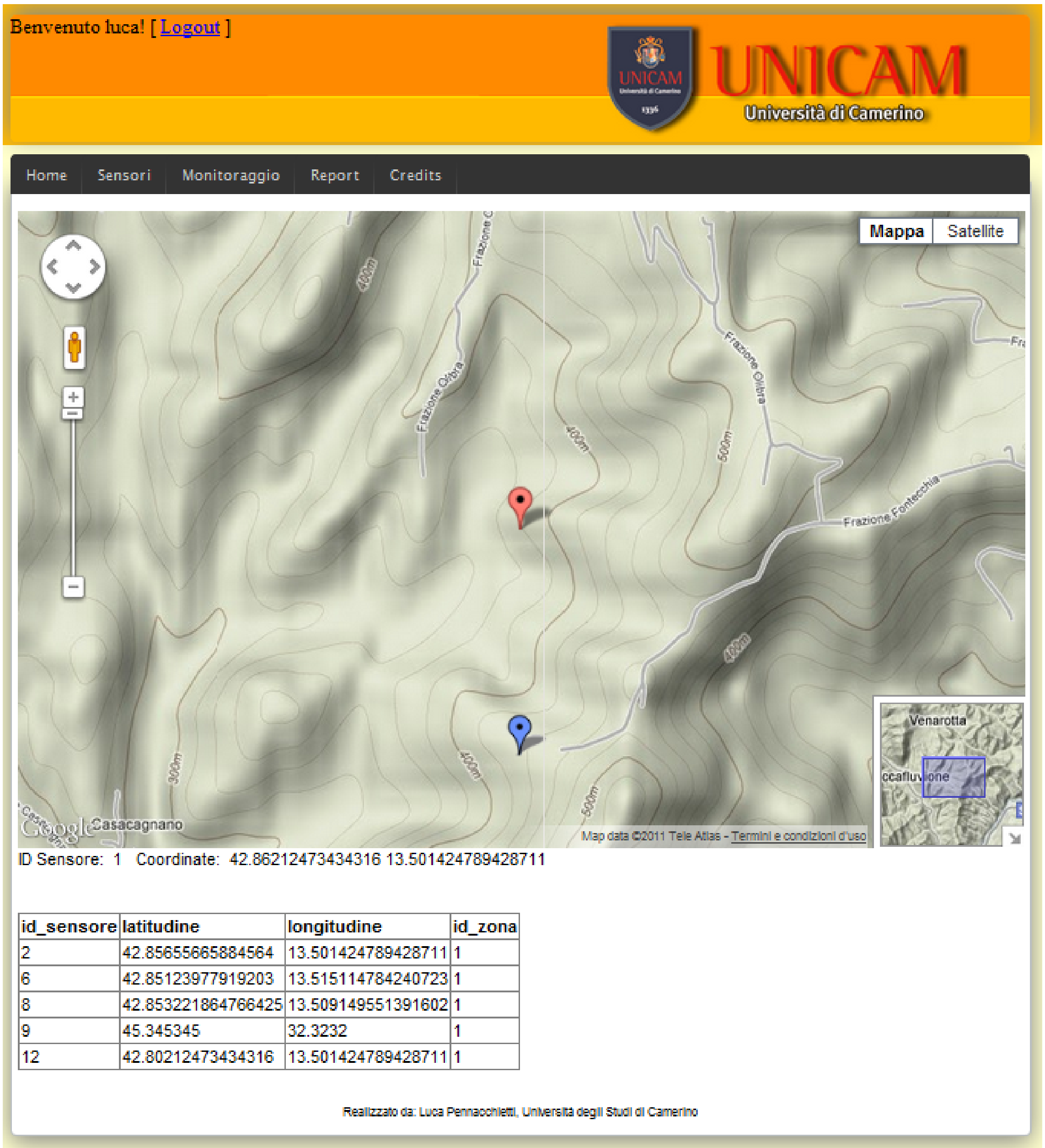

Figure 3.15: Mappa Page

The page that is accessed via the link "position" in the table on page "dati_sensori" shows a map generated by using the Google API, on which you place the red marker at the coordinates of the previously selected device, and blue markers in the location of all devices that belong to the same area of the selected one.
The latter data are also visible at the bottom of the page, just below the restricted area of the map, within a gridview that shows IDs of each device and the data concerning the location and area of origin. The map has been set as the default mode, the display mode "map" that allows you to see the various levels of elevation in which the devices were placed, so as to give a better idea of the structure of the sensor network. It also added the possibility to choose the view by satellite to see more realistically monitored area.

```
<script type="text/javascript"
      src="http://maps.googleapis.com/maps/api/js?sensor=true">
</script>
```

Listing 3.4: Including Google Maps Api

With these few lines of code in the project we include libraries of Google maps, which are loaded from the remote site.
From here on begins the script itself.

```
<script type="text/javascript">
```

Listing 3.5: Beginning of the script

In the next line of code is declared a variable set to null for the map, which is then loaded into theactual map.

```
var map = null;
```

Listing 3.6: variable "map" declaration

Inside the function "initialize" are defined before all the necessary variables, some of them as "Latitude" and "Longitude" take the values from the label created in the ASP page. net. The variables "testopopup" and "infowindow" define the content of the popup generated at the click of the mouse on the selected sensor.

```
function initialize() {
      var Latitude = document.getElementById("content_LabelLat").innerHTML;
      var Longitude = document.getElementById("content_LabelLng").innerHTML;
```

```
    var IdSensore = document.getElementById("content_Id").innerHTML;
    var myLatlng = new google.maps.LatLng(Latitude, Longitude);
    var numeroimmagine = document.getElementById("content_Label3").innerHTML;
    var image = 'Immagini/gmap_blue_icon.png';
    var myLatLng2 = new google.maps.LatLng(42.85130270352123, 13.506231307983398);
    var testopopup = '<div␣id="content">' +
                    '<h1␣id="firstHeading"␣class="firstHeading">Sensore␣' + IdSensore + '</h1>
                        ' +
                         '<div␣id="bodyContent">' +
                               '<p>Latitudine:' + Latitude +
                               '<p>Longitutdine:' + Longitude +
                               '<p>'+
                               '<img␣width=120␣height=80␣src="/Immagini/'+ numeroimmagine +
                                   '.png"␣>'
                               '</div>'
                     '</div>';


    var infowindow = new google.maps.InfoWindow({
                    content: testopopup
                    });

    var latlng = new google.maps.LatLng(Latitude, Longitude);
```

Listing 3.7: Function "Initialize"

Within the variable "myOptions" can be defined features which will take the map once initialized. For example, the zoom level is defined and the map is centered at the coordinates indicated by "latlng". Another peculiarity of this variable is the possibility to determine the look of the map, and that is whether it must be shown in political mode, physical or mixed. In this case the type is specified "TERRAIN" that allows you to see the various differences in height of the territory in question. In addition to this, and introduces a small window in the lower right corner of the map that shows the same area of interest but from a highest zoom level, so as to allow to better identify the area where you are. It is set to "opened" and that is immediately visible and not minimized.

```
var myOptions = {
                    zoom: 16,
                    center: latlng,
                    mapTypeId: google.maps.MapTypeId.TERRAIN,
                    overviewMapControl: true,
                    overviewMapControlOptions: {
```

```
                    position: google.maps.ControlPosition.BOTTOM_RIGHT,
                    opened: true
                      }
                      };
       map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
```

Listing 3.8: "myOption" variable declaration

The variable markers defines thecharacteristics of the markers to be displayedin the position indicated by the variables"Latitude" and "Longitude".

```
var marker = new google.maps.Marker({
              position: myLatlng,
              title: "Sensore"
              });
```

Listing 3.9: Marker definition

The method "setMap ()" is that who takes careto place the marker on the map indicating the location of the sensor.

```
// To add the marker to the map, call setMap();
marker.setMap(map);
```

Listing 3.10: Adding marker to map

This component handles events that occur during the use of the map. this case the event that is "captured" by the listener is the click of the mouse over a marker. In case this event will trigger the display of the variable defined within a pop-up "infowindow".

```
google.maps.event.addListener(marker, 'click', function () {
                                                            infowindow.open(map, marker);
                                                            });
```

Listing 3.11: Google maps event handler

The "Read_Data" is used to read inside a gridview, created in the ASP .Net page. , the data of sensors that are part of the same area that you selected. This table is loaded from the Id of the sensors, the latitude and longitude. these data will then be used to bring the sensor of the same area on the map, shown in contrast to the one selected with the color blue in a way that can be distinguished.

```

function Read\_Data() {
```

```
                    var idsensore = '';
                    var lati = '';
                    var longi = '';
                    var latlng = new google.maps.LatLng(Latitude, Longitude);
                    var posizionemarker = new google.maps.LatLng(lati, longi);
                    var image = 'Immagini/gmap_blue_icon.png';
                    var Latitude = document.getElementById("content_LabelLat").innerHTML;
                    var Longitude = document.getElementById("content_LabelLng").innerHTML;
                    var IdSensore = document.getElementById("content_Id").innerHTML;
                    var myLatlng = new google.maps.LatLng(Latitude, Longitude);
                    var numeroimmagine = document.getElementById("content_Label3").innerHTML;
                    var latlng = new google.maps.LatLng(Latitude, Longitude);


                    var Grid_Table = document.getElementById('<%= GridView1.ClientID %>');

                    for (var row = 1; row < Grid_Table.rows.length; row++)
                    {

                        for (var col = 0; col < Grid_Table.rows[row].cells.length; col++)
                        {

                            if (col == 0) {
                                            idsensore = Grid_Table.rows[row].cells[
                                                col].innerText;
                                        }

                            if (col == 1) {
                                            lati = Grid_Table.rows[row].cells[col].
                                                innerText;
                                        }

                            if (col == 2) {
                                            longi = Grid_Table.rows[row].cells[col].
                                                innerText;


                                        }
                        }

                    createMarker(lati, longi);
                    }
                }
}
```

Listing 3.12: "Read_Data" function

The "createMarker" function takes care to define all the characteristics that must have once placed markers on the map.
in this case as well as being specified by the position of the marker variables "sides" and "longitudinal", you specify the map on which they must be placed, the text displayed by the tooltip that appears to hover over each device, and the 'image must have the marker, in this case is assigned an icon in blue, in addition to the image of the shadow.

```
function createMarker(lati, longi) {

var shadow = 'Immagini/shadow2.png'
var myLatlng3 = new google.maps.LatLng(lati, longi);
var marker2 = new google.maps.Marker({
                                                position: myLatlng3,
                                                map: map,
                                                title: "Sensore",
                                                icon: "Immagini/gmap_blue_icon.png",
                                                shadow: shadow
                                          });

marker2.setMap(map);

}

</script>
```

Listing 3.13: Sample web.config Rule

The script ends here and within the definition of the body of the page, which must be displayed in the map we call the function "initialize" that just initializes the map with the rest of the features described above.

```
<body onload="initialize()">
      <div id="map_canvas" style="width:790px;␣height:500px"></div>
</body>
```

Listing 3.14: Calling "initialize" function

3.6.6 "Monitoraggio" Page

Using Cartesian graph to represent the values measured by sensors on each device has been chosen to use the libraries RGraph. The page has two DropDownList Monitoraggio.aspx, the second of which is filled by selection of the first.
In fact, the DropDownList "Select Area" allows you to select the area of membership of the devices that you want to monitor. The second DropoDownList instead shows the list of the selected device.
Once you choose which device you want to view statistics, will appear in the area below the two selections, a Cartesian graph representing the values measured in the intervals of time recorded in the database.
It is also chosen to make the chart interactive. Were in fact included the possibility of seeing the actual value of each interval detected, and the option to exclude, by the legend, all the measured variables, except the one of interest.
The declaration of the variable "line1" defines the characteristics that you want to have completed within the graph. In this case you want to see the lines representing the temperature, humidity, the value of the pir, and CO2.

```
var line1 = new RGraph.Line("myLine", temperatura, umidita, pir, luminosita, co2);
```

Listing 3.15: Variable "line1" declaration

The graph is created in order to have the x-axis exactly at the center of the axis of ordinates. Defines the background color, the color of the background grid at intervals and the amount that must be present in this case the number is10. Below is shown the code that defines the main features of the Cartesian graph inside the page:

```
line1.Set('chart.xaxispos', 'center')
line1.Set('chart.xticks', 10);
line1.Set('chart.background.barcolor1', 'rgba(255,255,255,1)');
line1.Set('chart.background.barcolor2', 'rgba(255,255,255,1)');
line1.Set('chart.background.grid.color', 'rgba(238,238,238,1)');
```

Listing 3.16: Graph appearence definition

In the next few lines of code is called the aspect that should take the lines representing the values of the quantities studied.

In addition is also described in the legend explaining the structure located on a corner of the graph, given also to allow interactive viewing of a quantity at a time, simply selecting the desired measurement on the legend.

```
line1.Set('chart.colors', ['#FF0000', '#FFCC00',
                           '#FFFF33', '#33FF00',
                           '#0000CC']);
line1.Set('chart.key', ['Temperatura', 'Umidit[U+FFFD]', 'Pir', 'Luminosit[U+FFFD]', Co2 ]);
line1.Set('chart.key.rounded', true);
line1.Set('chart.key.shadow', true);
line1.Set('chart.key.shadow.blur', 15);
line1.Set('chart.key.background', 'white');
line1.Set('chart.key.position.x', line1.Get('chart.gutter.left') + 650);
line1.Set('chart.key.position.y', line1.Get('chart.gutter.top') + 1);
line1.Set('chart.key.interactive', true);
line1.Set('chart.tickmarks', null)
line1.Set('chart.shadow', true);
line1.Set('chart.shadow.blur', 15);
line1.Set('chart.shadow.color', ['#faa', '#aaa']);
line1.Set('chart.linewidth', 1);
line1.Set('chart.filled', false);
line1.Set('chart.hmargin', 5);
line1.Set('chart.labels', [ u[0], u[1], u[2], u[3], u[4], u[5], u[6], u[7], u[8], u[9], u[10]]);
line1.Set('chart.gutter.left', 75);
line1.Set('chart.gutter.top', 110);
line1.Set('chart.gutter.bottom', 110);
line1.Set('chart.text.angle', 50);
line1.Set('chart.text.size', 7);
line1.Set('chart.tooltips', temperatura, umidita, pir, luminosita, co2);
line1.Set('chart.tooltips.effect', 'fade');
line1.Set('chart.tooltips.highlight', true);
```

Listing 3.17: Graph features definition

the method "Draw" does the real construction of the lines on the graph. line1.Draw ();

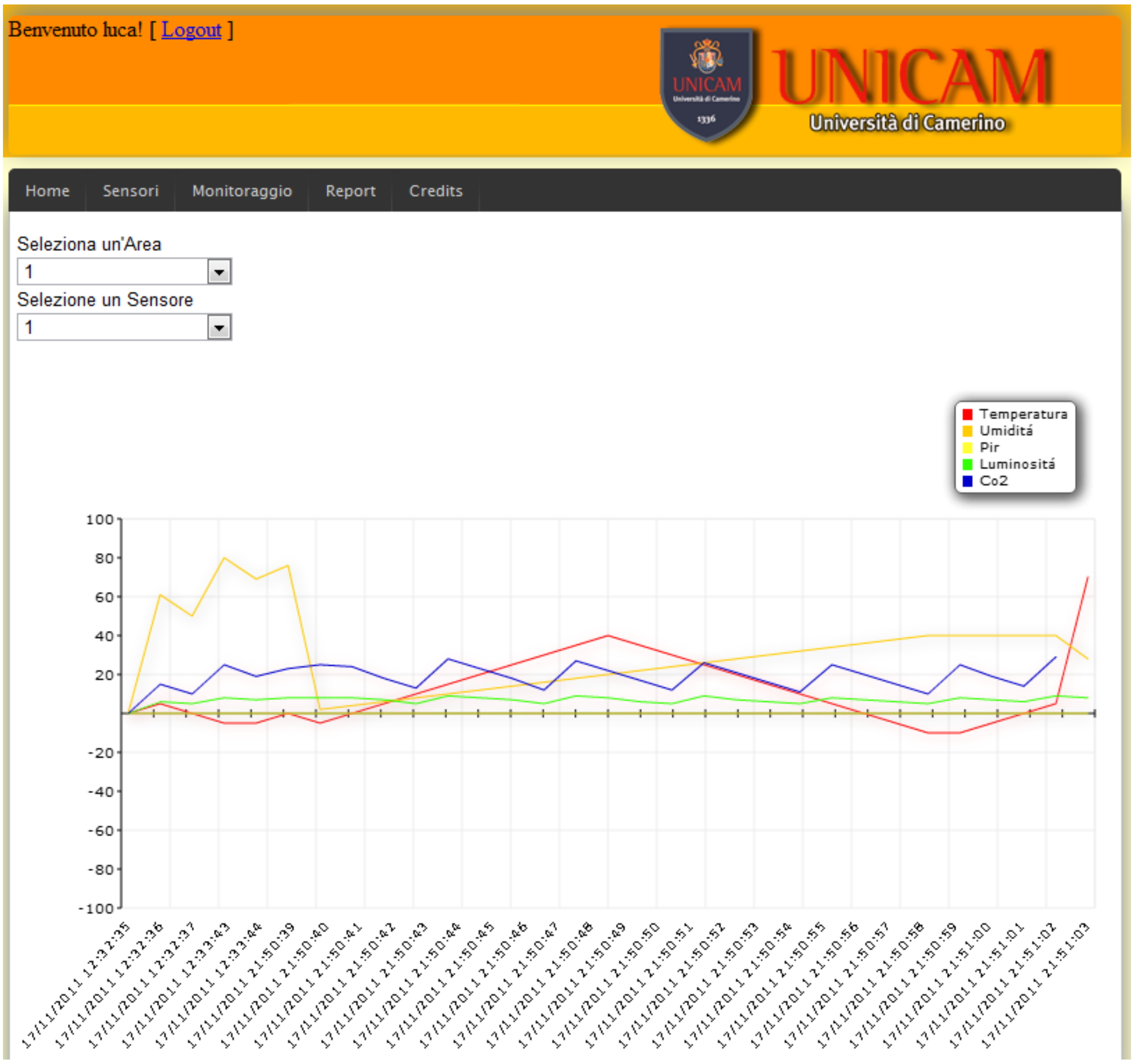

Figure 3.16: Monitoraggio Page

3.6.7 "Invia commando" Page

Figure 3.17: Invia commando Page

Through a link placed within the gridview on the page you reach Home.apsx "Command.aspx".This new page, using the same technique as the query string, do a select in the database according to the id of the device selected in the gridview mentioned previously, showing you the various commands that can be sent to the daemon which then forward it to the affected device . The Select uses the logical operator used to derive from a database table "commands" the actual order given to the various devices. The query in question is as follows:

```
Select elenco_comandi.nome_comando From elenco_sensori
left join elenco_comandi on (elenco_comandi.id_comando & elenco_sensori.comandi ) > 0
where id_sensore="1"
```

Listing 3.18: Select for "elenco$_{c}omandi$”

In particular case the sensor ID is 1, but within the page, the ID is passed through query string, and then inside the gridview are shown the commands for the device.

The commands that have been implemented for now are:

TIME_SLEEP: Command to set the sleep time of the processor parameter from 1 to 5 seconds. TIME_TX: Command to set the update time data coordinator, parameter 15 to 60 seconds. POWER_TX: Command to set the transmit power, a parameter from 0 to 3. CHANNEL: Command to set the transmission channel, parameter 11 to 25. RESET: Command to reset and then the subsequent restart of the device. No parameters. JOIN: Command to enable you to carry associations to a router or coordinator, parameter 0 = no 1 = yes

by opening a socket connection to the daemoninstalled on the server, the command will besent to the latter in the form of XML files. The XML file must be of the form:

```
<mac>9999999</mac>
<cmd>TIME_SLEEP</cmd><par1>[1..5]</par1>
<cmd>TIME_TX</cmd><par1>[15..60]</par1>
<cmd>POWER_TX</cmd><par1>[0..3]</par1>
<cmd>CHANNEL</cmd><par1>[11..25]</par1>
<cmd>RESET</cmd>
<cmd>JOIN</cmd><par1>[0..1]</par1>
```

Listing 3.19: XML format of command for device

The administrator, the only user who is allowed access to this area, can enter a numerical value of the command you wish to apply to the device. Through clicking on the "Send Command" is the sending of the message. A message will appear if the corms data transmission is successful, otherwise error.

3.6.8 Report Page

The Report page allows you to download .pdf, .doc and .xls document that collects data on temperature, humidity and CO2 values of those sensors that have detected measurements of these quantities equal to or higher than the predetermined threshold. The data are sorted by date from most recent to oldest. Below a screen representing the page in question:

Figure 3.18: Report Page

Inside the ReportViewer control is shown a preview of the document can be downloaded in several formats listed before. The Report has been created with SQL Reporting Service contained within the Visual Studio 2010 development tools.

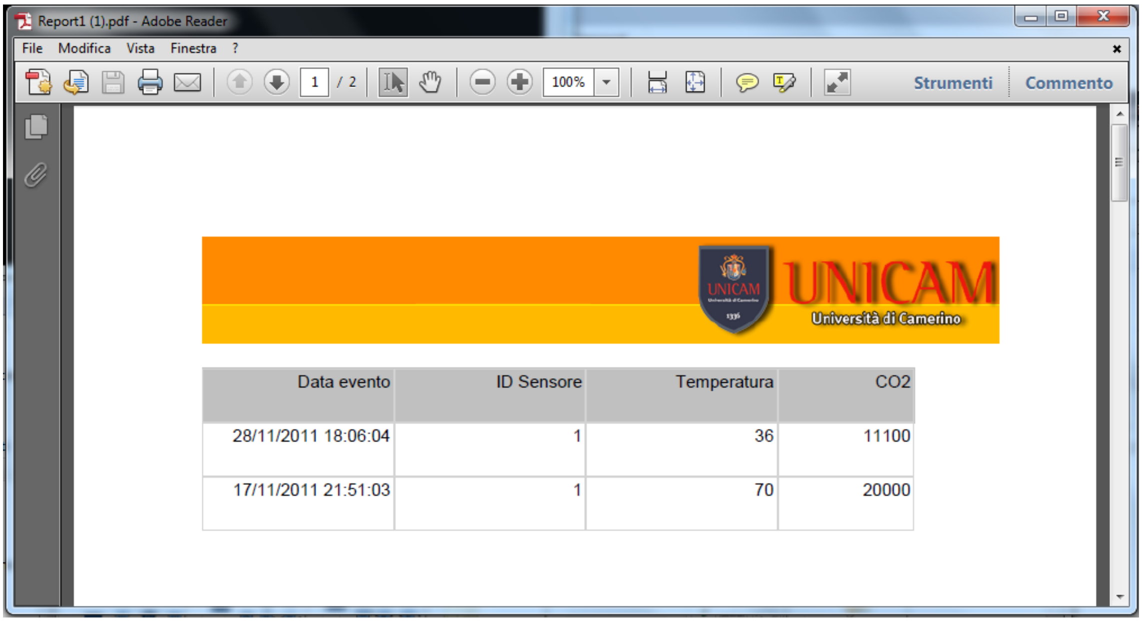

Figure 3.19: Report preview

3.7 Developed Software: daemon interface to the WSN

The application server is designed to allow communications between the wireless sensor network and the system. It uses the development kit provided by the supplier to perform communications with the sensor network. At the current level of achievement was not implemented graphical interface, so everything will be run by console commands.

The application started working as a daemon, listening for communications from the hardware side, just run the software provided by the supplier, and from the web side, listen for commands from the web pages can be sent to the sensor network.

To send commands to each device was inserted into the characteristics of each device a unique network address, a MAC address, through which the manufacturer's software manages the hardware communications with the devices themselves.

The screenshot below shows how the console application is waiting to receive any message.

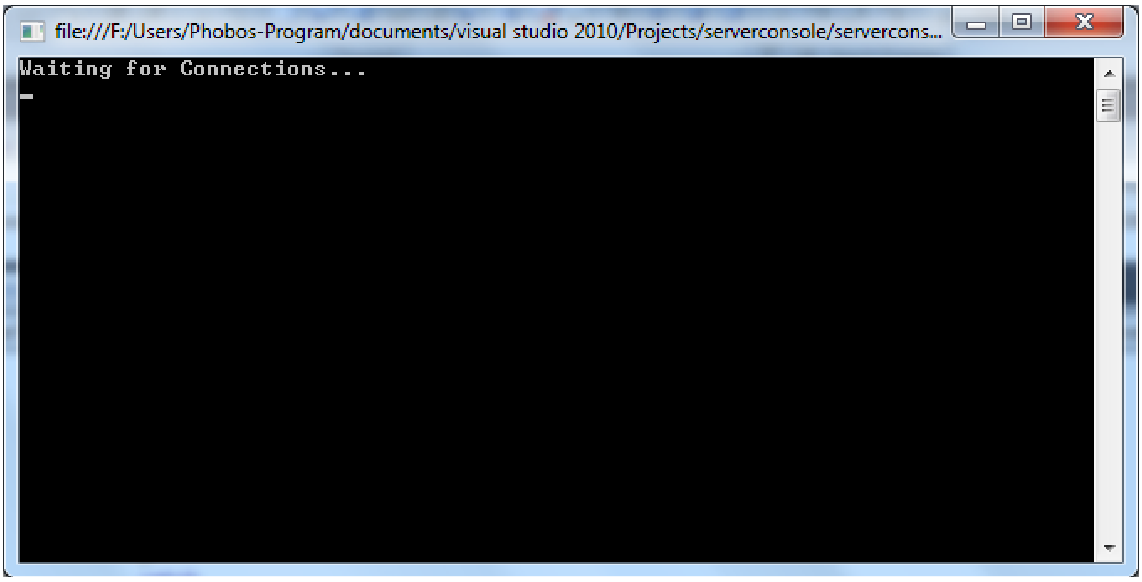

Figure 3.20: Server Application waiting

In the event that should receive a message from the web interface designed to one of the components of the wireless sensor network, the application will display the full content of the message in the form of XML files.

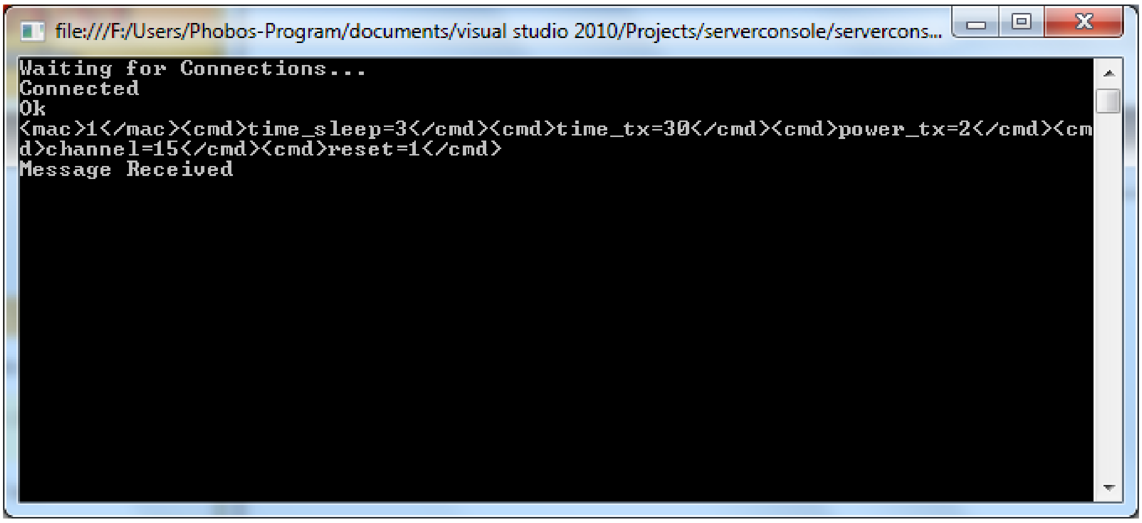

Figure 3.21: Message received by Daemon

In response to that message it will send, in case of receipt, a confirmation of the type "OK" to the web interface, if they had occurred instead, socket connection problems or other the message that will be sent will be "KO ".

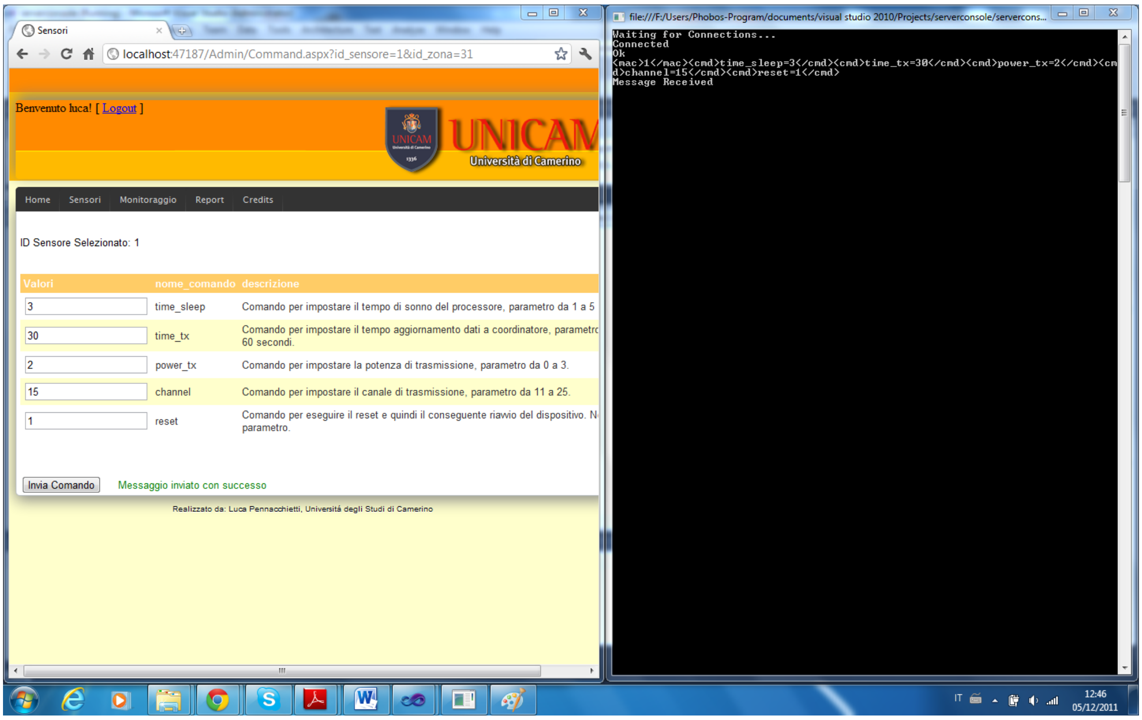

Figure 3.22: Message received successfully

In the event that received messages come from the sensor network, the daemon will query the database to verify the presence or absence of the sensor is communicating with whom. If it does it will conduct an "INSERT" in the database, inserting the values that the device is sending. In the event that in the database does not appear that the device corresponding to the MAC address, with which the daemon is communicating, will be added a new field containing the identification information of the device.

The identification code for each device is just the MAC address in the database but is sequentially added an additional ID.

Future versions of this application will be introduced some features.

The first of these is an event logger, which allows the daemon to create an xml file in

which are entered all the information regarding any alarms detected in the presence of levels of measured variables, outside the established norms.
Following must be made a more user-friendly graphical interface for end users, showing them in real-time exchange of messages with the web interface and that in case of suspected fires detected by the sensor network, the screen shows the log of the event.

Chapter 4

Conclusions

The proposed monitoring system is characterized by a low impact on the environment with a high return in terms of accuracy of the surveys useful not only for the fire prevention but also for more comprehensive statistics on the flora of the parks Italian.
For a real validation of the proposed technology and system software, a prototype system is being deployed in some wooded areas of the Mountain Community Esino-Frasassi (located in Fabriano (AN)). In this case the connection to the Internet is guaranteed by the regional network "Marche Way".
The forest has chosen to deploy the majority of conifers and in this respect, the experiment involves the use of cone-shaped containers camouflage for a minimal environmental and visual impact. The routers, which require solar panel, were made as boxes of or small birds.
The authors wish to extend the environment by implementing detection andrecognition of learning "patterns environmental" so as to adapt the system to an increasing number of cases and facilitate the adoption of broad-spectrum in the local authorities.

Bibliography

[1] Jennic, *IEEE 802.15.4 Wireless Networks User Guide*, 2006.

[2] protezionecivile.gov.it, Incendi Boschivi, *http://www.protezionecivile.gov.it/sistema/incendi.php*, 2010.

[3] wsl.ch, Fireless II, *http://www.wsl.ch/fe/oekosystem/insubrisch/projekte/FireLessII/index_EN*.

[4] zigbee.org, Products, *http://www.zigbee.org/*.

[5] fub.it, Guida alle Wireless Sensor Network, *http://www.fub.it/pubblicazioni/dossier/guidaallewirelesssensornetwork*.

[6] pce-italia.it, Sensori, *http://www.pce-italia.it/html/strumenti-di-misura/misuratori/sensoristica.htm*.

[7] rgraph.net, RGraph: HTML5 Javascript charts library, *http://www.rgraph.net/*.

[8] google.com, Google Maps Javascript API V3 Overlays, *http://code.google.com/intl/it-IT/apis/maps/documentation/javascript/overlays.html#Markers*.

[9] google.com, Google Maps Javascript API V3 Tutorial, *http://code.google.com/intl/it-IT/apis/maps/documentation/javascript/tutorial.html*.

[10] google.com, Google Maps Javascript API V3 Infowindow, *http://code.google.com/intl/it-IT/apis/maps/documentation/javascript/overlays.htmlInfoWindows*.

[11] libelium.com, Environment, *http://www.libelium.com/applications/environment*.