17th Symposium IMEKO TC 4, 3rd Symposium IMEKO TC 19 and 15th IWADC Workshop
Instrumentation for the ICT Era
Sept. 8-10, 2010, Kosice, Slovakia

An RFID Plug-n-Play smart sensors for monitoring forest fires

Fabrizio Ciancetta[1], Giovanni Bucci[1], Edoardo Fiorucci[1], Carmine Landi[2]

[1] *Dipartimento di Ingegneria Elettrica e dell'Informazione - University of L'Aquila, Località campo di Pile, 20 – 67100 L'Aquila (AQ), Italy, {fabrizio.ciancetta, giovanni.bucci, edoadro.fiorucci }@univaq.it*
[2] *Department. of Information Engineering - Seconda Università di Napoli, Via Roma, 29 – 81031 Aversa (CE), Italy, carmine.landi@unina2.it*

Abstract- Several applications require for a distributed measurement system able to measure the same or different parameters at different points in a wide area. Distributed systems based on smart web sensors represent the best solution to many different measurement problems. One of the main disadvantages of monitoring systems based on smart Web sensors is the difficulty to have a complete access to all data because they present a closed approach to interact with them, while Web Services are adopted in order to give a standard approach in developing a Service Oriented Architecture [1]. In this paper the authors present an implementation of a Plug-n-Play wireless sensor network based on Web service and adopting open source software and low cost hardware architecture. The sensor network is presented a framework of hierarchical distributed system for metrological application of monitoring. Measurement results are made available as Web services so that all users can build up their own applications. The particular metrological application chosen to show some preliminary results of distributed monitoring network is the monitoring of forest fire. This solution offers great possibility in terms of fast and easy access to measured data, of integration of large complex Web sensor network, of realization of flexible custom applications and of service reusability developing new measurement application starting with different information also coming from different sensors.

I. Introduction

Any biological, social or technical dynamic process needed of continuous measurements for monitoring system evolution and taking proper actions when needed. In this field of applications, smart sensors have been the best solution to many different metrological problems [1-3]. Adopting smart sensors, the client can read the state of a particular physical phenomenon with an application developed to receive information from the smart sensor which publishes the results on the Web. Unfortunately, at best author's knowledge, the devices developed in literature have many limitations. In fact, most popular implementations are based on system that produces static Web pages, adopting HTML, or data socket connection adopting applet java. The solution adopting the HTML generates static Web page in which both formatting tags and requester information are present. If a data stream and its graphical representation are required, it is necessary to leave HTML and to use a system that allows the creation of socket to exchange data: the traditional alternative to HTML approach is the adoption of applet java [4].
The main drawback of both approaches is that the smart sensor is a close system. In fact, with HTML, user has a passive role of simply static-page reader formatted by the web sensor manufacturer. On the other hand, with the java approach, as the manufacturer develops the protocol used to exchange information between server and client, he is the only one that can use or modify service. This leads to the necessity to use the software system developed by the manufacturer because he only knows the way in which server and client exchange information.
Moreover, for metrological applications, problems can come from long time required to access to information, from the locking of the sensor during client access, from the impossibility to access to the single data sent from smart sensor, from the impossibility to perform further data manipulation, from the impossibility to correlate two measures supplied on-line by two different systems, and so on [5-6].
In order to address all these issues, in this paper, a new architecture for smart sensor based on Web service approach is proposed for publication and sharing of measurements results. In fact, adopting Web service approach, smart sensors aren't simply supplier of internet pages to browsers but they become servers of measurement functions and so they can play a more complex role in monitoring system or in the integration of different measurement networks. This solution offers great possibility in terms of fast and easy access to measured data, of integration of large complex sensor network, of realization of flexible custom applications and of service reusability. The goal is to give the possibility to every client and every developer to obtain only the

needed information or to develop new measurement application starting with different information also coming from different sensors.
All these features come from the new concept of Web services that provides an infrastructure for the exchange of structured data in a distributed environment whatever language and platform is used.
Unfortunately, the most Web service limitation is the impossibility to give services in a dynamic way: when a developer wants to use a Web service, he downloads a file that describes the service, and then creates its application. The developer doesn't know if a new service (or a new smart sensor) is added or if it is deleted or doesn't correctly work. In this way, the entire system developed could be collapse and the application generates an exception that turn off the system. Generally, the removing or down services problem, can be managed with an appropriate control in run time during its use by the application. Moreover, it's impossible know which new services are added on the Web service, when they are connected and which are their functions. That limitation is most important when a sensor network is upgraded to extend the area to control or to change the sensor configuration or placement [7].
The authors present an implementation of a smart sensor based on web services approach utilizing open source software and adopting low cost hardware architecture for monitoring forest fires. To give much more freedom to the whole system the RFID technology has been adopted to allow a deep penetration of the proposed network in the forest. The sensor is presented in the framework of a hierarchical distributed system for metrological application of monitoring.

II. Network architecture of the proposed distributed measurement system

The proposed network architecture is reported in Figure 1. Three are elements present in the network: the Coordinator, the Routers and the End Devices. The Coordinator covers an important rule in the sensors network because it receives the request from the Internet interface and resends it to the sensors network. The Routes, which are smart systems, are been developed to extend the network range receiving and resending the wireless packets on the distributed network. They operate on level 2 in ISO/OSI stack linking the Coordinator with the End Devices. At the end, in the sensors network are present the End Devices that cover the sensing parts of the network. They receive wireless packets from the Coordinator via the Routers and response to the request performing the measurement of the environmental parameter. The request bounces back to the Coordinator which sends the response to the Web service Interface (WsI). The WsI connects the sensors network with the Internet network. It is a Web services that interface the sensors network with a client using SOAP messages. The use of Web service gives much more freedom to the network and in order to avoid the access to broken End Device it is have been used the adoption of dynamic Web service [7].

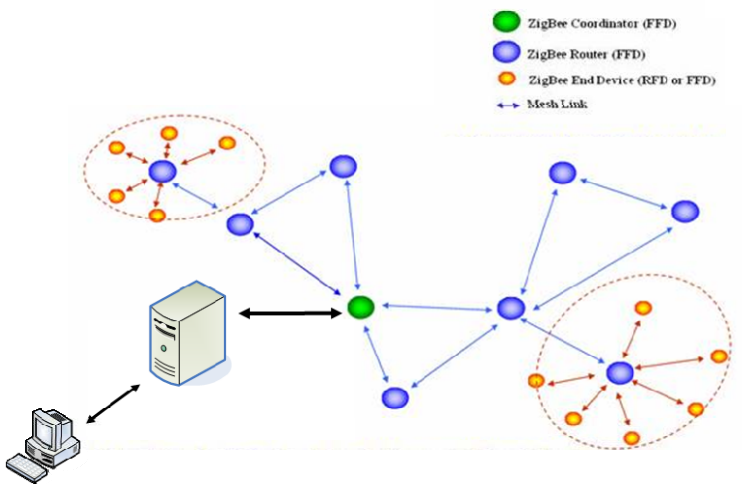

Figure 1. Network architecture

A. Coordinator / Router

The Coordinator and the Router adopt the ZigBee protocol to send wireless packet. Based on IEEE 802.15.4, ZigBee is a standard wireless communication system. In literature, many are the wireless sensors network that presents the limitation of the using of ad hoc communication protocol [8]. The use of ZigBee gives a more degree of freedom of the wireless network. Besides, the ZigBee operative range is about 100m to 1.000m in

function of the obstacles that are enough for the particular application. In fact, the use of other RFID system (active of passive) reduces the operative range giving value from 1m to 10m.
In order to have a dense monitoring system and to increase the details and the resolution of the area under test, it is necessary use many End Devices but, using this strategy, the cost of the whole system increases. In the other hand, the use of away End Point reduces the details of the area and the cost too. So, a good compromise is in the use of middle range system based on ZigBee, which can guarantee a good area resolution and reduce the cost of the monitoring system
The adopted Coordinator and Router hardware is reported in Figure 2, while in Figure 3 the ZigBee stack is reported.

Figure 2. RFID Coordinator/Router based on ZigBee protocol

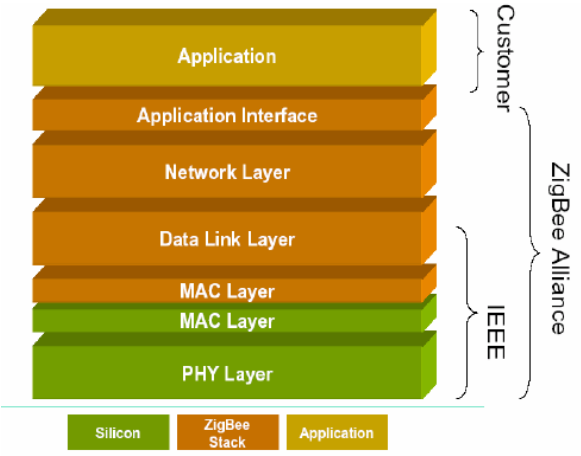

Figure 3. ZigBee stack

B. End Device
The End Device are based on a low cost, low power microcontroller PIC 18LF4620. The microcontroller is a CPU RISC @ 10 MIPs with 32 KB of program memory and 1.5 KB of data memory, 4 time modules, 3-wire SPI, 1 I2C, 1 addressable USART module and 10-bit Analog-to-Digital Converter (A/D) @ 30 kSps with 8 input. The RF front-end is based on MRF24J40MA that is connected with the microcontroller via a SPI interface as reported in Figure 4. On the microcontroller, the acquired data are buffered and pre-processed using a 5-taps moving average system with a sliding window technique. The data are then processed using a 3rd order IIR filter to investigate the monitored signals during fast variations. We tested the maximum operative range for the specific application. The results have shown an operative range of 450 m. It is important underline that the performed test are been executed with no other ZigBee network near the tested on. In fact, the presence of other ZigBee network can reduce the operative range and the network capacity due to the collision of wireless packets.
In Figure 5 a developed End Device placed on a tree has been photographed.

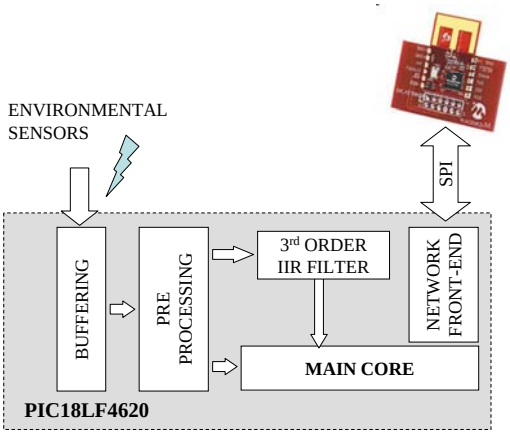

Figure 4. Diagram block of the proposed End Device

Figure 5. Developed End Device placed on a tree

C. Web service Interface

The Web service Interface is the element that interfaces the sensor network with the client. Based on open source software, it realizes a Web service using NuSOAP PHP class, while a network interface links the Web service functionality (called Web methods) with the sensor network as reported in Figure 6. The WsI uses a MySQL server to store all the data provided by all the End Device. In particular, the Web methods exported are:

- *EndDevicePresent:* the Web method pings all End Device connected to the sensor network. The active one which responses to the ping are alive and so the client can access to its functionalities. As results, the Web method provides a list of accessible End Device address.
- *EndDeviceServices(EndDeviceAddress):* from the alive End Device, the Web method access to the specific End Device using the parameter *EndDeviceAddress* and as result the client has a list of service (or sensor) present on it.
- *AccessEndDevice(EndDeviceAddress,Service):* using the *EndDeviceAddress*, the client with this Web method can access to a specific service using the parameter *Service*. As result, the client has a floating point number that represent the value of the sensor.
- *AccessCurrentData(EndDeviceAddress,Service):* using the *EndDeviceAddress* and the *Service* parameters, the client can download the data from MySQL server relatively in the course of the day.
- *AccessHistoricData(EndDeviceAddress,Service,DateStart,DateEnd):* using the *EndDeviceAddress* and the *Service* parameters, the client can download the data from MySQL server from a DateStart to a DateEnd

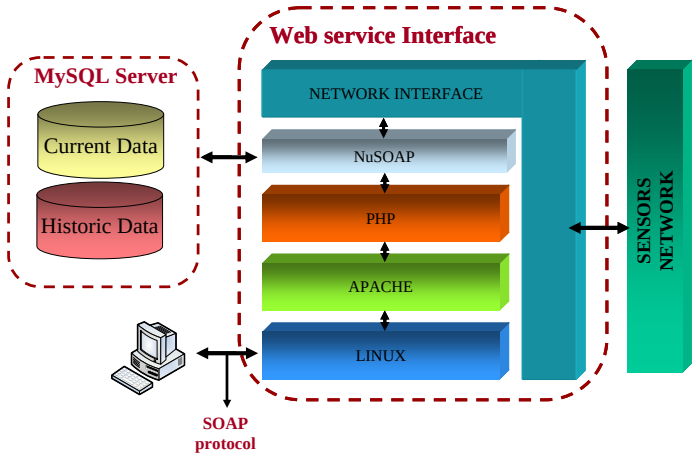

Figure 6. WsI architecture

III. Results

In order to test the proposed sensors network, a user interface to keep all the information about the sensors present to the network, the services exported, the state of the sensors and the access of the measurement data has been developed. The user interface has been implemented using Nation Instruments CVI SDK. In the user interface, reported in Figure 6, the PAN address of generated sensors network has been reported. Besides, for each sensors that join the network its assigned network address and its exported functionalities have been reported. The user interface allows the access to the selected wireless sensor downloading the current measurement data and setting the monitoring parameters for asynchronous access.

Next, to consume the proposed sensors network via Internet, a user Visual Basic .NET 2005 interface based on Microsoft Framework .NET 3.0 has been developed. In particular, the client accesses to the WsI using the SOAP protocol. During the development of the user interface, a proxy class has been created downloading the WSDL XML file. Using the automatic generated class, the client can access to all the Web methods, and so, to all the functionality present in the network. The steps to consume the sensors network have been: i) download the list of sensors actual connected to the Coordinator: in the list is supplied not only the network address of the End Device present in the network, but also other secondary information on the sensors such as the GPS coordinates and other information about the developer; ii) selecting an End Device, the user interface can download the actual situation accessing to all the sensors presented on it, or can download the daily data stored in a database present in the WsI. In Figure 8 is reported a screenshot of the developed client.

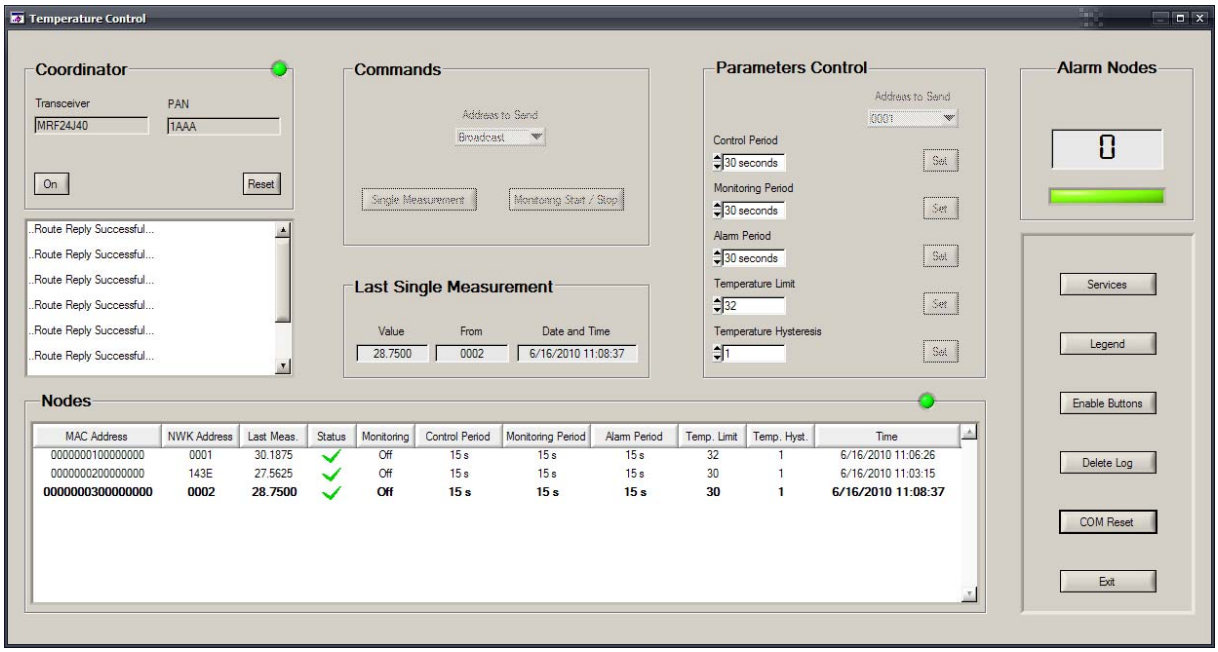

Figure 7. A .NET client that access to the wireless sensors network

Figure 8. A .NET client that access to the wireless sensors network

References

[1] F. Ciancetta, B. D'Apice, D. Gallo, C. Landi, "Architecture for Distributed Monitoring based on Smart Sensor and Web Service", IMTC 2006 – Instrumentation and Measurement Technology Conference Sorrento, Italy, 24-27 April 2006.

[2] F. Ciancetta, E. Fiorucci, B. D'Apice, C. Landi "A Peer-to-Peer Distributed System for Multipoint Measurement Techniques" Proceedings of IEEE Instrumentation and Measurement Technology Conference, Warsaw, Poland, May 1-3, 2007.

[3] J. Pottie and William J. Kaiser: Embedding the internet: wireless integrated network sensors, Communications of the ACM, 43(5), pp. 5 1-58, May 2000.

[4] D. Grimaldi, L. Nigro, F. Pupo, "Java-based Distributed Measurement Systems", IEEE Trans. on Instrumentation and Measurement, vol. 47, no. 1, pp. 100-103, Feb. 1998.

[5] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Coverage Problems in Wireless Ad-Hoc Sensor Networks”, IEEE Infocom 3 (2001) 1380-1387.
[6] D. Gurkan, X. Yuan, D. Benhaddou, F. Figueroa, J. Morris “Sensor Networking Testbed with IEEE 1451 Compatibility and Network Performance Monitoring”, Sensors Applications Symposium, 2007. SAS '07. IEEE, Page(s): 1 - 4
[7] F. Ciancetta, B. D’Apice, D. Gallo, C. Landi "Plug-n-Play Smart Sensor Based on Web Service", IEEE Sensors Journal, vol. 7, issue 5, MAY 2007 Page(s): 882-889
[8] G. Bucci, E. Fiorucci, C. Landi, G. Ocera: Architecture of Digital Wireless Data Communication Network for Distributed Sensors Applications. Measurement vol. 33 n.1, Jan 2004.