Communication and processing consideration for alerting systems implemented through mobile sensor networks

DAN POPESCU, RADU DOBRESCU, MAXIMILIAN NICOLAE, and ALEXANDRU IORDAN
Department of Control and Computers
POLITEHNICA University of Bucharest
313 Splaiul Independentei, Bucharest 6,
ROMANIA
dan_popescu_2002@yahoo.com

Abstract: - At the beginning of the paper it is presented an alerting system architecture based on mobile sensor network. Then it is analyzed the current available radio communication infrastructure which can be used for implementation of alerting systems in conjunction with the alerting systems requirements. Further we shown how the digital signal processor (DSPs) are the best fit for the processing level. We propose some ideas of selection criterion based on this analyze and an implementation of a mobile sensor network with applicability in alerting systems. Built with the purpose of surveillance, data gathering and warning, alert systems could benefit from the increase flexibility offered by a mobile sensor network. In order to ensure a low cost and an extended battery life, for communication between nodes the ZigBee standard was chosen. For the prompt response of the system we design a software kernel with real time operation and implement on DSP architecture.

Key-Words: Mobile sensor networks, Alert system, Communication protocols, Digital signal processing, Power consumption, Software structure.

1 Introduction

An alerting system has the purpose of monitoring, detecting and acting against a predefined alert situation. The alert situation, in general, can be anything from a forest fire to an earthquake.

The final “clients” of the system can be a closed, specific group like the workers in a plant or the general public in the case of earthquake alerting.

An alerting system is composed of three major components (Fig.1): sensor network, processing centre, and dissemination system.

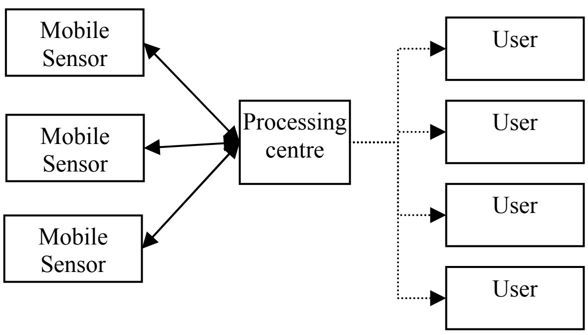

Fig.1 General architecture for mobile sensors based alerting system

The sensor network gathers relevant data from the field and transmits it using a short range wireless standard to a management node. From here, using a long range wireless standard, the data are sent over long distances to a central processing centre which is able to transform data into information.

After receiving data from sensors it is applies certain algorithms to determine the presence of an alert condition. In some cases, the presence of an alert condition is determined using prediction methods. For this purpose, data storage and retrieval are necessary. If an alert condition is detected it must generate proper response.

The dissemination system must transmit the alert to the final users of the system. They can be civilians, industrial plants, local fire stations and others. Alert messages can be informative, easily recognised by humans, or digital, meant for electronic equipment. For the general public, alerts can be disseminated via television, radio, mobile phones, sirens, smart displays etc. For industrial users, the alert system is interconnected with the automatic control system. Alerts are used as input information for the control loop, allowing for the whole system to self regulate. In critical situation the entire installation is shut down.

Special precautions should be taken regarding false alerts. For the general public, such events degrade the trust in the system and could lead to a poor efficiency. For the industrial users, false alarms could have a negative economical impact.

We have chosen to implement a solution for the field network that involves mobile platforms for sensors (Fig.2). Mobile sensors permit a greater degree of freedom when devising a monitoring algorithm. This enables dynamic algorithm to be used. Sensors can be relocated according to specific conditions in the field for the case when a certain area necessitates more attention. Another advantage in various applications is that a small number of platforms can cover a greater area. By following a predefined circuit, a sensor reads data periodically.

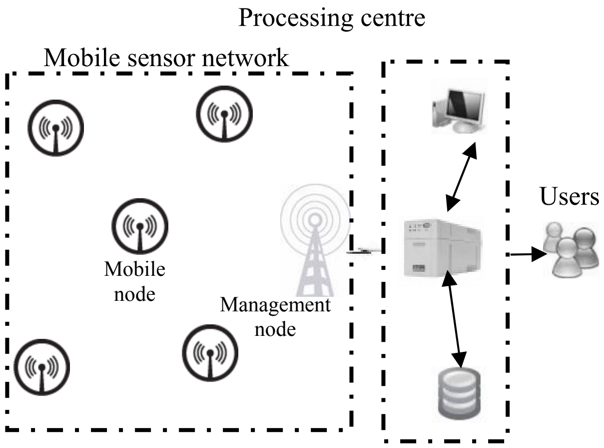

Fig.2 Mobile sensor alarm system deployment

For the purpose of a proper radio communication infrastructure choosing, we will analyze the current available standards and propose some criterion for selection. After this we will present our architecture implementation and discuss the real time software kernel for communication.

2 Mobile nodes characteristics

The mobile sensor network is part of the larger family of wireless ad-hoc networks in terms of communication. Ad-hoc networks, as the main characteristic, do not have a fixed structured topology. The nodes come online at arbitrary positions and at arbitrary moments in time. Also, during functioning, nodes can change position or go off-line abruptly. Many of the aspects described below are common both to wireless ad-hoc networks and to sensor networks. The difference is that conditions imposed to sensors are stricter. Some of these constraints are presented farther.

Size and weight are important factors to take into account when designing a mobile sensor. For it to be indeed mobile it must be as small and as light as can be. This makes the task of installing such sensors on mobile platforms much easier. Also, for some applications, small size and weight are sine-qua-non conditions. For example, installing large or heavy sensors on wild animals in a forest area or in the sea is impossible.

In typical applications, mobile sensors are designed to have a *very long battery life*. In many cases, such networks are composed of a great number of similar devices. This makes periodic sensor retrieval a burden for the system administrator. For some applications, the lifetime of a mobile sensor is equal to the lifetime of its battery. Replacing the whole device is sometimes more cost efficient than replacing just the battery. For this kind of implementations, the battery is embedded in the device and this actually gives more freedom to the designers.

Often, the sensors are thought of as replaceable. For example no one can afford to simply lose a device endowed with a GPS receiver which is quite expensive. They must be produced in great numbers and with *a cost as low as possible*. Of course this forces the whole mobile sensing part of the system to be more efficient and ingenious.

Communication will be less than ideal. A mobile sensor network is not deployed in a controlled environment but in a real life situation, usually outdoor. Thus, the characteristics of the medium will vary greatly because of many factors: weather changes, appearance, repositioning or disappearance of certain obstacles and others.

A mobile sensor network must accommodate a large number of mobile nodes. From the start of the design stage it must be taken into account that the network can grow up to several thousand nodes and thus the network must be *scalable*.

3 Architecture analysis and selection

In this paper we approach only two issues of a mobile sensor node, also the most important from our opinion. These are the communication level and the data processing level. In this paragraph we consider only the architecture of each module based on available infrastructure as selection criterion.

3.1 Communication level

In order to choose the proper technology for our communication infrastructure, we had two options. One option was to develop our own proprietary standard for radio communication or to adopt an existing standard.

First option has big disadvantages from the fact that radio communication is a mature field in which advances take place in IC manufacturer companies who have the required technology and a proprietary method will be hard to interface with other systems which the application could require.

Further we made a comparison between available wireless communication technologies in order to choose the proper technology for our communication infrastructure. This comparison is presented synthetic in Appendix A.

For better overview the Appendix A can be presented graphically as in Figure 3. On the axes are represented two features very important for selecting the technology, namely distance versus data rate, both in logarithmic scale. The area of each technology representation figures the level of energy consumption approach in design of the respective technology. A small area represents a standard with energy consumption optimization; instead a wide area represents a technology which wasn't designed for energy efficient.

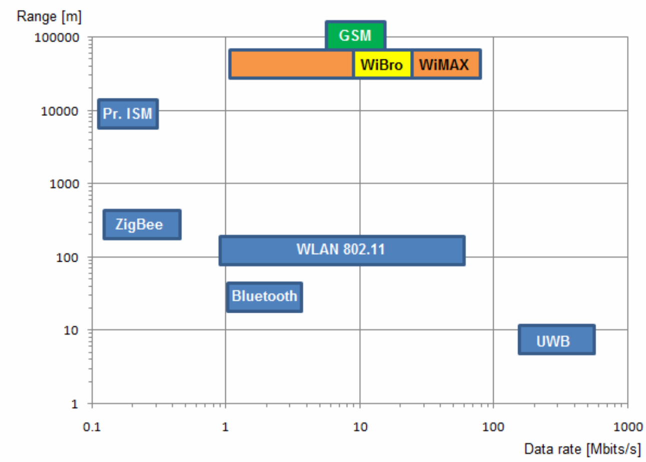

Fig.3 Graphical representation of wireless technologies performances

Based on requires demanded by the mobile sensor network, this representation can be seen like a guide from which, can be chosen the suitable wireless technology for a communication module. An example of how to choose the proper technology is presented in Table 1 where the eliminatory demand was low power consumption.

Table 1. Low power wireless technologies

		Range	
		Low	Wide
Data rate	Low	ZigBee	GSM(GPRS)
	High	UWB	WiBro, GSM(HSDPA)

Everybody would desire, if possible, that the covering range to be very wide, as well the data rate much higher and the energy consumption very low [1]. Of course, all of these must be realized without supplementary costs for traffic. It is obvious that these demands are contradictory and some compromises must be made. If it is analyze the variants resulted from Table 1 one can observe that the WiBro technology would be the best choice, may be this is the reason why this technology is still on developing study and it is available only in particular areas of the globe and with very expansive equipments. The next choice is GSM technology represented by its third generation (3G) standard HSDPA. In this case it is the same thing as previous one. It doesn't have a high price for equipments but it has a high cost for data traffic.

Because the UWB technology (Ultra Wide Band) is designed for PANs (Personal Area Network) and has a covering range of 10m, it also loses its viability in the mobile sensor network field. Finally we remain only with two options for communication infrastructure technology, ZigBee (practically it was especially develop for such application as mobile sensor networks), and GSM through its lower cost data transfer services GPRS and EDGE.

But the most important conclusion following this communication technology analysis makes imperative the demand for distributing the algorithms at the node level. In other words, the node becomes an intelligent one, capable of complex data processing with low power consumption. These two aspects are another compromise that must be treated.

On the other hand, we consider that in the very probable eventuality of reaching the technology which meets the requirements of high data rates with low power consumption, low equipments price and free of traffic charge (what WiBro proposes) will still not be a feasible implementation for the case where large amounts of information are departing from mobile nodes taking advantage of the available communication wide band. This kind of implementation which was proposed by others authors will lead quickly to flooding the central processing node and to network congestion. Also, in this case solutions exist by prioritizing the messages so that the critical messages have high priority. If the information will be processed with delay and only the critical information will be processed faster, in the image processing case, the question is how will someone (from the network) know that some particular image acquired is a critical one in order to be prioritized? We consider that much efficient will be if each image will be processed local at the node level and, in the critical case or on demand only, the true relevant information must to be transmitted to the central management node (without additional, irrelevant or redundant information which it is often meet in image transmission applications). This solution will ensure a truly real-time functionality.

3.2. Processing level

Depending upon their system demands, the designers have a variety of architectures to choose, when they develop digital signal processing applications. The solutions are the following.

ASIC: Designing their own chip is a route that few developers take on lightly. While ASICs give the best performance for specific applications, they are limited in their flexibility. If any design parameters change, the entire ASIC must be redesign, a process that consumes months and potentially hundreds of thousands of dollars in NRE (non-reoccurring engineering) costs. Given the complexity of designing with ASICs, the lack of adaptability to changing markets, and slow time-to-market, ASICs are most often a less than ideal choice.

FPGA: Field Programmable Gate Array (FPGA) devices provide performance close to ASICs without the delays and costs associated with re-spinning an ASIC. FPGAs, however, are as complex to design with as ASICs, and even small changes can result in the need to completely re-layout a design. FPGA devices also come with a price premium that makes them unviable in high volume applications. In certain applications, FPGA can be used to complement the DSP.

ASSP: Application-specific standard products (ASSP) are ASICs that can serve a wide market. Such devices include dedicated processing engines such as MPEG-2 decoders or communications links such as USB. While these devices are low in cost because of the high market volume they typically experience, they still lack flexibility. For example, an MPEG-2 ASSP will only support a limited range of display resolutions; to introduce a new resolution requires a new ASSP and accompanying hardware redesign.

GPP: General purpose processors (GPP) offer programmability and therefore flexibility. However, many non-signal processing applications such as email, database management, and word processing don't require extensive use of multiplication. In order to keep silicon costs down for these applications, GPPs typically offer a moderately efficient multiplication instruction that takes several cycles to complete. In addition, adding the result of each multiplication requires another instruction. As result, this makes them less than ideal for signal processing applications.

DSP: Without getting too technical, a majority of signal processing functions multiply two series of numbers and sum the results:

$$result = x1 * c1 + x2 * c2 + x3 * c3 \ldots xn * cn \quad (1)$$

DSPs are optimized for signal processing applications compared to general purpose processors (GPP). DSPs offer many architectural features that actually reduce the number of instructions necessary for efficient signal processing. In other words, comparing performance is much more than counting instructions. The true measure is how much work is actually done.

The programmable flexibility of DSPs enables developers to implement complex algorithms in software. Not only can a DSP support a video codec like MPEG-2 and easily handle different resolutions with a simple software upgrade, it can implement emerging codecs and standards as they arise without the hardware redesign.

Thus, having in mind the previous consideration we choose for our processing level a **DSP core**. Some of the most important characteristics we analyzed were: programmability, throughput, on-chip memory, integration, power, real-time, responsiveness, high availability, standards, broad product, platforms, performance.

4. Chosen Solution for Mobile Node

The project requirements and the limitation imposed by building a mobile node, determined us to use an AT86RF230 transceiver from ATMEL. The technical characteristics of the AT86RF230 transceiver, which recommended it for our application, are:

- High Performance RF-CMOS 2.4 GHz Radio Transceiver Targeted for IEEE 802.15.4 and ZigBee Applications
- Programmable Output Power from -17 dBm up to 3 dBm
- Receiver Sensitivity -101 dBm
- Ultra-Low Power Consumption: SLEEP- 20 nA, RX- 15.5 mA, TX- 16.5 mA (at max transmit power of 3 dBm)
- Fast registry and buffer access via a SPI interface
- Industrial Temperature Range: -45 ^{o}C to 85 ^{o}C

Its characteristics are consistent with our requirements for a low-power, cheap, modular, reliable, mobile node. Using this transceiver, our platforms are able to communicate acquired information to a management node. This node is capable to transmit date over long distances to a monitor and command centre.

The IEEE 802.15.4 standard specifies that communication should occur in 5 MHz channels ranging from 2.405 to 2.480 GHz. In the 2.4 GHz band, a maximum over-the-air data rate of 250 kbps is specified, but due to the overhead of the protocol the actual theoretical maximum data rate is approximately half of that. While the standard specifies 5 MHz channels, only approximately 2 MHz of the channel is consumed with the occupied bandwidth. At 2.4 GHz, 802.15.4 specifies the use of Direct Sequence Spread Spectrum and uses an Offset Quadrature Phase Shift Keying (O-QPSK) with half-sine pulse shaping to modulate the RF carrier.

The transceiver is controlled by the central processing unit of the node, a DSP microprocessor. In order to ensure the modularity of the system, we have chosen our electronic components so that the communication between them is standardised.

The DSP and the AT86RF230 communicate via a SPI bus (Fig. 4).

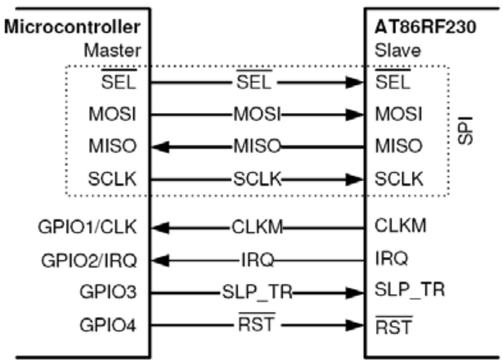

Fig. 4. SPI signal diagram

Microcontrollers or microprocessors, with a master SPI, interface directly to the AT86RF230. The SPI is used for frame buffer and register access. The additional control signals are connected to the GPIO/IRQ interface of the DSP.

The most important characteristic was the energy consumption and for this we choose the C55x DSP family from Texas Instruments. The new C55x DSP generation builds on the C54x generation, considered the industry's most power-efficient DSPs. Specific features of the C55x DSP core include:

Advanced Automatic Power Management - the C55x DSP core provides advanced automatic power management for all peripherals, memory arrays and individual CPU units. In effect the C55x DSP core continually monitors which parts of the chip are in use, powering them off when they are not needed.

Increased Idle Domains - For even more advanced power management, user configurable idle domains customize power consumption for a specific application. The C55x DSP core extends the three fixed idle choice of the C54x with a total of 64 user configurable combinations of the following six components: CPU, Cache, Peripherals, DMA, Clock generator, External Memory Interface (EMIF).

The result is extremely efficient power usage and lower system cost. By using less power and dissipating less heat, the C55x DSP core gives designers much greater flexibility regarding board design.

One of the C55x DSP core's most innovative features is its support for variable-length instructions, based on a new byte-addressing scheme:

- Instruction length may be 8-, 16-, 24-, 32-, 40-, or 48-bits.
- Instruction fetch increased from 16-bits to 32-bits.
- On-chip instruction buffer unit automatically unpacks instructions to make the most efficient use of each clock cycle.

The reduction in C55x DSP core memory bus activity cuts power requirements, while longer instructions carry out more functions per clock cycle-resulting in increased performance and lower system cost.

The C55x DSP core's focus on increased parallelism provides cycle efficiency improvements with:

- Additional Hardware - Dual 17 x 17-bit MACs, a second 16-bit ALU, four new data registers (which can be used for simple computations) and four 40-bit accumulators make it possible to accomplish more work per cycle, dramatically lowering overall power requirements.
- New instructions capabilities: auto-parallel instructions, implicit or built-in parallel instructions, user programmed parallel instructions, additional instructions for orthogonality increasing.
- Additional Buses and Expanded Addressing - To make sure that throughput can attain the theoretical maximum made possible by the new computational hardware, the C55x DSP core features: three 16-bit data read buses, two 16-bit data write buses, one 32-bit program bus, six 24-bit address buses.

Improved Control Code Density - several new control-code features in the C55x DSP core make it possible to integrate control code on the DSP, eliminating the need for a separate microcontroller:

▪New Instruction Buffer Unit - Capability to handle variable instruction lengths means control code is packed tightly and man-aged efficiently, reducing power.

▪New Data Registers and ALU - Four new data registers, together with a new 16-bit ALU, make it possible to carry out simple arithmetic and logical operations that typify control code.

▪Conditional execution - Many control code operations involve branching on a certain condition.

To speed execution, the C55x DSP core prepares for both possibilities, so that once the condition occurs, the DSP is ready to act immediately.

External Memory Interface - the C55x DSP core EMIF includes increased bandwidth, expanded memory options, and an automatic power-off feature, plus 32-bit wide interface to high-speed, low-cost, synchronous memories such as:

▪Synchronous Burst SRAM and synchronous DRAM

▪Asynchronous SRAM, DRAM, ROM & Flash

Instruction Cache with Burst Fill - the C55x DSP core features an instruction cache. At the core operating speed of the C55x DSP core, the distance between external memory and on-chip functions become factors in keeping the chip operating at top speed. In addition, it takes only slightly more power to reach information not stored on the chip. Loading instructions from external memory into an on-chip cache helps ensure that instructions will be available to the CPU at clock speeds. It also helps cut power requirements, because several instructions can be loaded into cache at the same time, and the CPU does not have to access memory for each instruction.

Shortened Debug - New Advanced Emulation hardware included on the C55x DSPs works with eXpressDSP™ Software and Development Tools to speed and simplify debugging.

▪Non-intrusive debugging: Set a "watch point" and observe changes in specified registers while the code runs, without stopping the DSP.

▪Real-Time Data eXchange (RTDX): See what program output will look like, without stopping the DSP.

▪Trace FIFO: Works with TI's XDS510 emulator, which allows the DSP to save the last 16 PC discontinuities and 32 PC values.

Because of the node being mobile, the topology of the network is of mesh type. Every node is capable of communicating with all neighbours that are in his transmitting range (Fig.5).

Through peer-to-peer connections any node in the network can contact any other node within the covering range of the network.

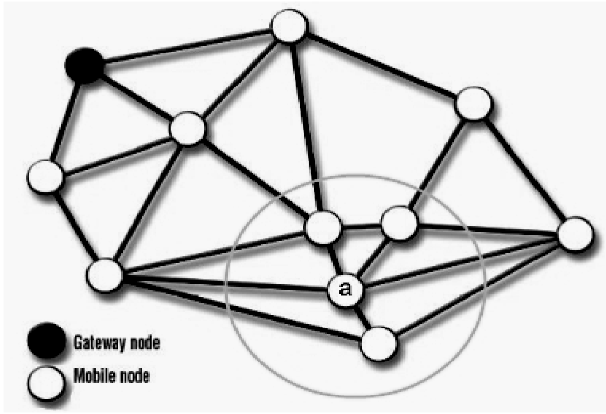

Fig. 5 Mesh topology

Deployed in an interest area, the mobile sensors network offers a flexible way of surveillance. Each node is designated a fraction of the area. The movement pattern is design in such a matter, that the transmitting range of each node is constantly overlapping with at least one neighbouring node. In this way the transmission of acquired data to the management node (or gateway node) and eventually to the command centre, is assured.

Another advantage of this transceiver is the possibility of adjusting transmission power level and also to measure the received signal strength at the input. This last information is used to proper adjust the transmission power for energy efficient consumption.

In embedded devices, the received signal strength is converted to a received signal strength indicator (RSSI) which is defined as ratio of the received power to the reference power (P_{Ref}).

An increasing received power results in a rising RSSI. Figure 6 illustrates the relation between RSSI and the received signal power.

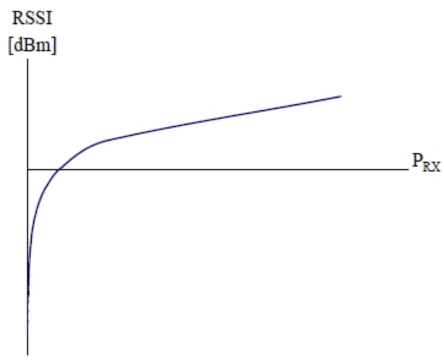

Fig. 6. RSSI - P_{RX} relation

For the fault tolerant measure the following mechanism is used. If a node has detected an alarm situation, firstly it asks the closest neighbour to come and confirm the finding. So, false alarms due to faulty sensors are avoided. While the neighbouring node confirms the alarm, the other nodes reposition themselves so that data transmission from the affected area to the management node will be as fast and as power efficient as possible. Once the message is sent, all nodes resume their previous position and tasks.

For the purpose of demonstrating our idea, we built four mobile nodes (Fig. 7), each equipped with two miniature video cameras. This allows us to create a multitude of scenarios in which a node detects an alarm situation and then we can test the communication and the coverage algorithm. We chose to work with video cameras because an image is a complex set of data, and if we manage to transmit imagine through our network then any other type of data would work as well.

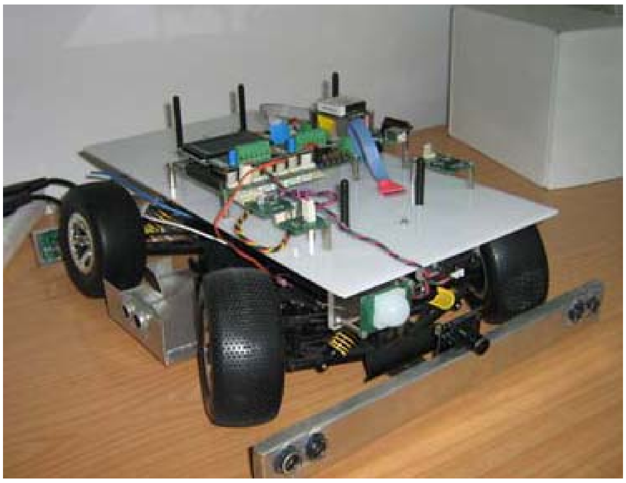

Fig. 7 Mobile platform

In real world implementation, the mobile node can be equipped with a wide range of sensors, depending on the type of the alarm system. We can use smoke sensors or infrared cameras for a fire alarm system, wind and humidity sensors for a precipitation and flood alarm system and many others.

4. Real-time software architecture

The software of communication module is divided in a communication part and an application part. The link between the communication and application part is the dictionary of objects. In Fig.8 it is showed the software structure.

The communication part contains the task of communication in network and a real time kernel witch includes a task launcher, a Round Robin scheduler, resource sharing flags and handles of interrupts. The application part contains up to seven tasks witch solve all the equipment demands less those in connection with the communication in network.

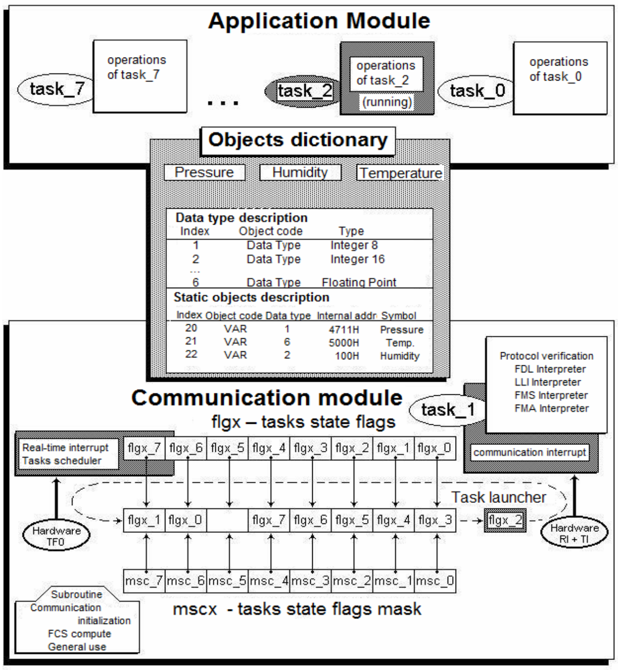

Fig.8 Software architecture of communication module

Each task has a status flag in FLGX byte and a mask flag in MASX byte. Thus the bit 0 of FLGX, called flgx_0 is the status flag of task number 0, task_0, while bit 0 of MSCX, mscx_0 is the mask flag of task_0, bit 1 - flgx_1 and mscx_1 for task_1 and so on. A task is active or inactive if the mask flag of this task is set or not. A task is ready to be launched in execution if the status flag of this task is set and it is cleared after the task was launched.

The launching in execution of a task can be prepared in an interrupt service routine different to the real time interrupt service routine, in other task or in the task scheduler. The real time interrupt service routine includes the task scheduler. It prepares the launching in execution of tasks which should run periodically (base time, multiple of base time, seconds, minutes, etc.). These tasks have one counter of launching and a constant of launching (initial value of counter). The counters of launching are decremented with one to each interrupt of real time. If the launching counter of one task becomes zero will be set the status flag of this task and the counter will be reinitiate. The task will be launch in execution only if it is active (its mask flag set).

The task launcher is the main loop of software and it is included in the real time kernel. It is showed in Fig.9 where with TF0 is noted the real time interrupt.

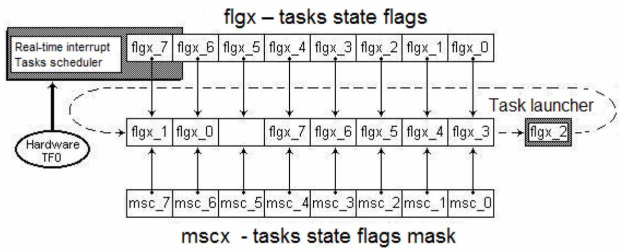

Fig.9 Task launcher

In the main loop is tested in Carry, bit by bit, the result of logical AND between the copy of FLGX (the byte of all eight status flags) and MSCX (the byte of al mask flags). After the test of all eight bits of result it is made again logical AND between the copy of FLGX and MSCX and the test in Carry bit by bit continue. In figure 7 we can see the test of bit 2 of result of logical AND. To each rotation to right by Carry is incremented a pointer to a jumping table. This table contains the addresses to the beginning of all eight tasks. If a bit is set, it is saved the context and the adequate task will be launched in execution. To end of task the program return in loop (in task launcher) and the test continue.

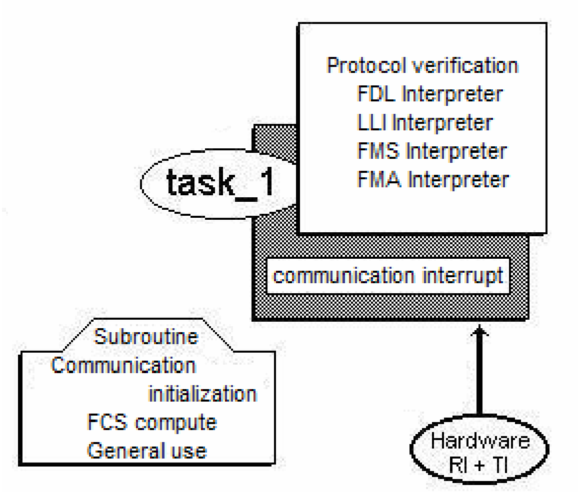

Fig.10 Communication task

The task number 1, task_1, is allocated for communication in network. In Fig.10 it is showed this task which includes all the functions for communication in network. This task, together with task launcher, task schedule, real time interrupt service routine, and communication interrupt service routine form the communication part (Fig.8). For an existing network and for participants to communication which occupy the same hierarchical position, the software differences appear only to level of application part and to the dictionary of objects.

5 Conclusion

We consider that this approach of using mobile sensor networks for alerting systems is a fair one in this fast growing domain. Based on wireless technologies, we demonstrated that the key solution for developing real-time system (ex: emergency system for environment or disaster management) is to distribute the processing capabilities to the sensorial nodes transforming a node in an intelligent autonomous one. In this way we reduce the network traffic and increase the reliability of the entire system being much prompter in response and increasing the life of each node. Further, we came to the conclusion that the best available solution for our system was the ZigBee protocol and we develop our implementation based on this one.

Concerning the proper choosing, regarding the complexity of algorithms running on the node, the processing core is a digital signal processor (DSP). At this time the software architecture is tasted on an 8-bit microcontroller platform. This explains why the software is divided in up to eight tasks: one task, which form the communication part and the remained seven tasks grouped together in the application part. The next step is to implement this architecture on a 32-bit DSP platform end extend the number of tasks to this value, this way obtaining a more powerful software kernel.

Appendix A. Comparison between available wireless communication technologies

	Bluetooth	UWB	ZigBee	802.11a/b/g	Proprietary ISM	WiMAX (802.16a)	WiBro (Mobile WiMAX - 802.16e)	2G/2.5G/3G (GSM/GPRS/ EDGE/ HSDPA)
Typical Range	<30m	<10m	70-300m	100m	10km	50Km	50Km	Cellular Network
Frequency Range	2,4GHz	3.1-10GHz	868/915MHz 2.4GHz	2.4GHz –b/g 5.8GHz – a	868/915MHz 2.4GHz	2-11Ghz	2.3-2.4 GHz	900MHz/ 1800MHz
Multiple Access Method	Adaptive FHSS	OFDM/ DS-UWB	TDMA	CSMA/CA	FHSS	TDMA/ OFDMA	OFDMA	TDMA/ FDMA/ CDMA
Duplex Method	TDD	TDD	FDD	TDD	FDD/TDD	TDD/FDD	TDD	FDD
Users per Channel	7 active, 200Inactive	-	255	127	variable	variable	variable	8/ 15-50
Channel Spacing	1MHz	-	5MHz	FHSS:1MHz DSSS:25MHz OFDM:20MHz	-	20/25/28 MHz	9MHz	200KHz 5MHz
Modulation	GFSK/ DQPSK/ D8PSK	OFDM	GFSK/ O-QPSK	OFDM, 64QAM, 16QAM, etc	-	...256QAM	4/16/64 QAM	GMSK/ 8-PSK/QPSK/ OQPSK
Peak Data Rate	1/2/3Mbit/s	480Mbits/s	250kbps	54Mbps	256kbps	70Mbits/s	18Mbit/s	14.4kbits/s 115kbit/s 384kbits/s 10.8Mbit/s
Network Type	P2P	P2P	Mesh	IP&P2P	P2P	IP	IP	IP
Internet Access	No	No	No	Yes	No	Yes	Yes	Yes
Cost of Data	Free	Free	Free	Free	Free	Free	Free	Monthly Charge
Energy consumption optimization	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Application	Cable replacement	Sync or Transmission of video/audio data	Sensor networks	LAN, Internet	Point to point connectivity	Metro area broadband Internet connectivity	Mobile metro area broadband Internet connectivity	Cellular telephones and telemetry

References:

[1] Foerster, J., Green, E., Somayazulu, S., Leeper, D. (2001). "Ultra-Wideband Technology for Short- or Medium-Range Wireless Communications". In*: Intel Technology Journal Q2*

[2] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks***,** Wiley, 2007;

[3] S. Olariu, L. Wilson, M. Eltoweissy and K. Jones, Training a wireless sensor network, *Mobile Networks and Applications*, 10, 2005, pp. 151-167;

[4] F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, Wireless sensor networks: A survey, *Computer Networks*, 38 (4), 2002, pp. 393–422;

[5] Z. Bojkovic, B. Bakmaz, A Survey on Wireless Sensor Networks Deployment, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 12, Volume 7, December 2008

[6] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, Next century challenges: Scalable coordination in sensor networks, *MobiCom*, 1999, pp. 263-270;

[7] W. Ye, J. Heidemann, and D. Estrin, An Energy-Efficient MAC protocol for Wireless Sensor Networks, *INFOCOM*, pp. 1567-1576, June 2002;

[8] A. Gani, Q. Han, N.B. Anuar, O. Zakaria - Enhancing DSR Protocol Performance in Mobile Ad Hoc Network Using ACK Reply, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 2, Volume 8, February 2009

[9] W. Ye, J. Heidemann, and D.Estrin, Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks, *IEEE/ACM Transaction*, vol. 12, pp.493-506, June 2004;

[10] T.van Dam and K.Langendoen, An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, *SenSys*, pp.171-180, November 2003;

[11] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks, *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 243-255, Apr 2004;

[12] J.-Y. Choi, S.-J. Yim, Y. J. Huh, Y.-H. Choi - A Distributed Adaptive Scheme for Detecting Faults in Wireless, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 2, Volume 8, February 2009

[13] R. Dobrescu, D. Hossu, S. Mocanu, M. Nicolae, New Algorithms for QoS Performance Improvement in High Speed Networks, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 12, Volume 7, December 2008

[14] E. J. Duarte-Melo and M. Liu, Data-gathering wirelesssensor networks: Organization and capacity, *Computer Networks*, 43(4), 2003, pp. 519–537;

[15] S. Meguerdichian, S. Slijepcevic, V. Karayan and M.M. Potkonjak, Localized algorithms in wireless ad hoc networks: Location discovery and sensor exposure, *Proc. MobiHoc* , 2001, pp.106-116;

[16] C. Savarese, J. Rabaey and J. Beutel Locationing in distributed ad-hoc wireless sensor networks, *ICASSP Proeedings*, 1, 2001, pp. 301-305;

[17] R. Dobrescu, M. Dobrescu and St. Mocanu, Using Self Similarity to Model Network Traffic, *WSEAS Transactions on Computers*, Issue 6, Vol. 3, Dec.2004, pp. 1752-1757;

[18] C. Gui and P. Mohapatra, Power conservation and quality of surveillance in target tracking sensor networks, *Information Processing in Sensor Networks*, 2005, pp. 246 – 253;

[19] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hämäläinen, M. Hännikäinen, T. D. Hamalainen, Ultra-Low Energy Wireless Sensor Networks in Practice: Theory, Realization and Deployment, Wiley, 2007;