# ACQUISITION, PROCESSING, AND COMMUNICATION MODULE OF A WSN-BASED SYSTEM FOR ENVIRONMENT MONITORING

Dan POPESCU[1], Radu DOBRESCU[2], Constantin VASILESCU[3], Maximilian NICOLAE[4]

*Articolul prezintă arhitectura unei rețele de senzori wireless (WSN) şi structura hardware/software, realizată de autori, pentru nodul mobil inteligent, elementul central al rețelei. Configurația nodului mobil inteligent asigură achiziția de imagini şi date, prelucrarea, comunicația şi vizualizarea la distanță. A fost dezvoltată o aplicație a rețelei de senzori wireless pentru un sistem de monitorizare a mediului.*

*The paper presents the architecture of a wireless sensor network (WSN) and the hardware/software structure, which was developed by authors, for the intelligent mobile sensor node, the central element of this network. The structure of intelligent mobile sensor node ensures image and data acquisition, processing, communication and remote visualization. It was developed a WSN-based application for an environment monitoring system.*

**Keywords**: alerting system, data acquisition, wireless communication, digital signal processing, hardware structure, and software structure

## 1. Introduction

A wireless sensor network (WSN) consists of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations [1], [2]. Sensor nodes may change their location after initial deployment. Mobility can result from environmental influences such as wind or water. Also, sensor nodes may be attached to or carried by mobile entities, and sensor nodes may possess automotive capabilities [1].

[1] Prof., Faculty of Control and Computers, University POLITEHNICA of Bucharest, Romania, dan_popescu_2002@yahoo.com
[2] Prof., Faculty of Control and Computers, University POLITEHNICA of Bucharest, Romania, radud@isis.pub.ro
[3] PhD Eng., Faculty of Control and Computers, University POLITEHNICA of Bucharest, Romania
[4]Assist., Faculty of Control and Computers, University POLITEHNICA of Bucharest, Romania

In the recent past, wireless sensor networks (WSN) have found their way into a wide variety of applications and systems with vastly varying requirements and characteristics. As a consequence, it is becoming increasingly difficult to discuss typical requirements regarding hardware issues and software support. This is a specific problem in multidisciplinary research areas such as wireless sensor networks, where close collaboration between users, application domain experts, hardware designers, and software developers is needed to implement efficient systems [1].

Today sensors are internetworked via a series of multi-hop short-distance low-power wireless links (particularly within a defined sensor field); they typically utilize the Internet or some other network for long-haul delivery of information to a point (or points) of final data aggregation and analysis. In general, within the sensor field, WSNs employ contention-oriented random-access channel sharing and transmission techniques that are now incorporated in the IEEE 802 family of standards [3]. The limited battery life of sensor nodes raises the efficient energy consumption as a key issue in wireless sensor networks.

The classical architecture for controlling mobile robots is through remote control and a lot of advances was made in this direction [4] but when a large number of mobile robots are involved, sending images as a feedback from all the robots to one or more remote locations is not an option because whatever method of multiplexing is used for communication there will not be enough bandwidth [5].

The main objective of our research consists of a conceptual model and corresponding architecture elaboration for intelligent mobile sensor nodes, based on internal processing theory and complementary data fusion. As result, a functional model, embedded system type, for image and data acquisition, processing, communication and remote visualization was implemented. The functional model which is a demand configured one, has high technical and economical performances.

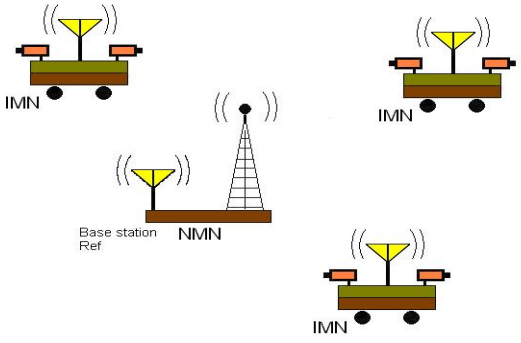


Fig.1. Mobile sensor network architecture

The network is composed by moving objects (modular nodes) with time depending topology, and therefore, with geometrical and temporal node coordinates. They are endowed with complex programmable equipments, low power consumption, able to acquire image and process data, to interact with other nodes in order to identify its position, and to communicate wireless. The system architecture is presented in Fig.1, with the following abbreviations: NMN - network management node, IMN - intelligent mobile node, REF – reference system for tracking and localization. A sensor node (IMN) consists of two electronic modules, the first: Power and Motion Module and the second: Acquisition, Processing and Communication Module.

In acquisition, processing and communication module the streaming data coming from video interface, sensor interface, and wireless communication interface must be processed in real-time. Every time current data arrives over the interface and the software must read it before new data arrives. Thus, the software, which is necessary to guarantee a certain capability on a specific time constraint, must be a real-time one. The above solution was used for the software of IMN modules. Both modules can be reprogrammed in-system to provide an application-tailored functionality [6].

## 2. Hardware Structure

Compared to general purpose processors, the DSP is optimized for signal processing applications and offers 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 DSP 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 hardware redesign. Thus, having in mind the previous consideration we choose for our processing module a DSP core.

The structure of the Intelligent Mobile Node (Fig. 2) contains two functional modules: Power and Motion Module and Acquisition, Processing, and Communication Module. These modules communicate one with other using the full-duplex multichannel buffered serial ports McBSP2 configured as Serial Peripheral Interface, SPI in Fig. 2.

The Acquisition, Processing, and Communication Module contain four functional blocks (Fig. 3). It uses the High-Performance, Low-Power, Fixed-Point DSP, TMS320VC5509A, with 5-ns Instruction Cycle Time to 200-MHz Clock Rate, based on the TMS320C55x DSP generation CPU processor core [4].

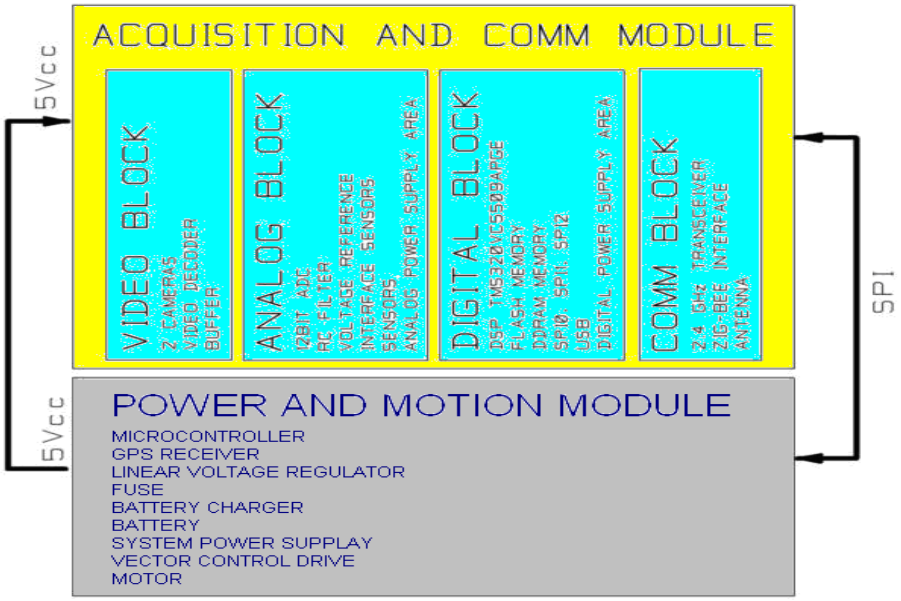


Fig.2. Structure of Intelligent Mobile Node

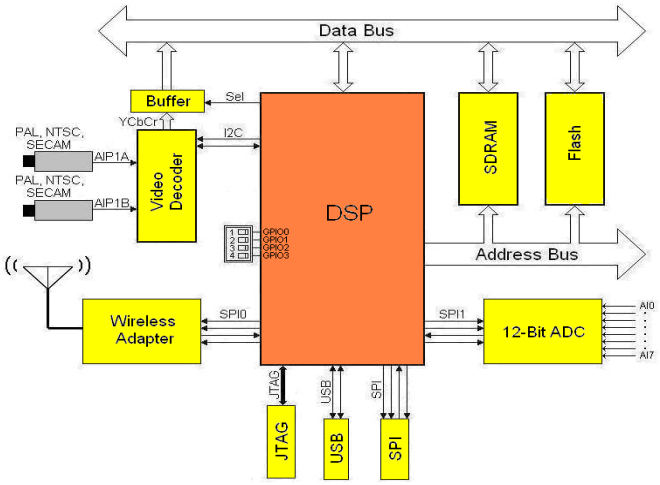


Fig.3. Block diagram of the Acquisition, Processing and Communication Module

The C55x™ CPU provides two multiply-accumulate (MAC) units, each capable of 17-bit x 17-bit multiplication in a single cycle. A central 40-bit arithmetic/logic unit (ALU) is supported by an additional 16-bit ALU.

The peripheral set includes an external memory interface (EMIF) that provides glueless access to asynchronous memories like EPROM and SRAM, as well as to high-speed, high-density memories such as synchronous DRAM.

Additional peripherals include Universal Serial Bus (USB), real-time clock, watchdog timer, $I^2C$ multi-master and slave interface and two channel 10-Bit successive approximation A/D. Three full-duplex multichannel buffered serial ports (McBSPs) provide glueless interface to a variety of industry-standard serial devices, and multichannel communication with up to 128 separately enabled channels. The DMA controller provides data movement for six independent channel contexts without CPU intervention, providing DMA throughput of up to two 16-bit words per cycle. Two general-purpose timers, seven dedicated general-purpose I/O (GPIO) pins, and digital phase-locked loop (DPLL) clock generation are also included.

*Digital block*

Digital block includes (Fig. 3) the DSP with 32Mx16bits synchronous DRAM MT48LC32M16A2, 512Kx16bits Flash memory AT49BV802DT, three full-duplex multichannel buffered serial ports (McBSPs) which are configured as Serial Peripheral Interfaces protected by three 10-bit FET bus switches SN74CBT3384A, the USB interface protected by low-capacitance 2-channel +/-15kV ESD-protection array for high-speed data interface TPD2E001, JTAG interface (IEEE 1149.1), and digital power supplies.

*Video block*

The video block contains the video decoder TVP5150AM1-EP, two video cameras, and the low voltage octal bidirectional buffer with 3-state inputs/outputs SN74LVT245. This video decoder is an ultralow-power NTSC/PAL/SECAM type. Available in a space-saving 32-terminal TQFP package, the decoder converts NTSC, PAL, and SECAM video signals to 8-bit ITU-R BT.656 format. Discrete syncs are also available. The decoder consumes 115mW of power in typical operation and consumes less than 1mW in power-down mode, considerably increasing battery life in portable applications. The decoder uses just one crystal for all supported standards. The TVP5150AM1 decoder can be programmed using an $I^2C$ serial interface and it converts baseband analog video into digital YCbCr 4:2:2 component video. Composite and S-video inputs are supported. The decoder includes one 9-bit ADC. Sampling is ITU-R BT.601 (27.0MHz, generated from the 14.31818-MHz crystal or oscillator input) and is line locked. The output formats can be 8-bit 4:2:2 or 8-bit ITU-R BT.656 with embedded synchronization.

Video decoder communicates with the DSP using $I^2C$ interface and data bus (Fig. 3). The $I^2C$ controls the decoder's internal configuration registers while data bus is used to read digital video data through a buffer. The transfer of digital data output of video decoder is perform by the DMA of DSP. For each data available the video decoder generates an interrupt which is synchronized with a

DMA event. Each frame taken at high speed (20 fps) is transferred in the SDRAM. Then, the DSP will analyze some rare frames (2fps) for which, if it will find something "interesting", it will be capable to access the neighbor frames from the memory in order to improve the decision.

*Analogue block*

To enrich the analogue capabilities of Acquisition, Processing and Communication Module it is used the ADC AD7888 which contains eight single-ended analog inputs, AIN1 through AIN8, Fig. 4. This converter is a high speed, low power, 12-bit, and operates from a single 2.7V to 5.25V power supply, VDD. The AD7888 is capable of a 125 kSPS throughput rate. The AD7888 features an on-chip 2.5 V reference that can be used as the reference source (VREF) for the ADC. The REFIN/REFOUT pin allows the user access to this reference. Alternatively, this pin can be overdriven to provide an external reference voltage for the AD7888. The voltage range for this external reference is from 1.2 V to VDD. The analog input on each of these channels is from 0 to VREF. CMOS construction ensures low power dissipation of typically 2 mW for normal operation and 3 µW in power-down mode.

It is used a 2.5V external reference with AD780. This ultrahigh precision band gap reference AD780 provides a pin programmable 2.5V or 3.0V output. Low initial error and temperature drift, combined with low output noise and the ability to drive any value of capacitance, make the AD780 the ideal choice for enhancing the performance of ADC. The 12-bit ADC AD7888 communicates with the DSP using the full-duplex multichannel buffered serial ports McBSP1 configured as Serial Peripheral Interface, SPI1.

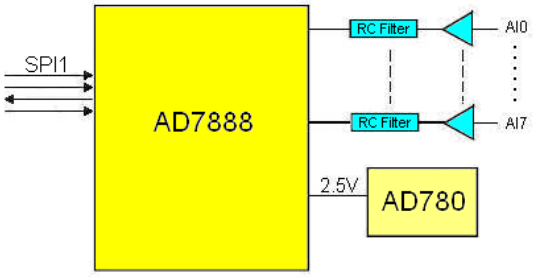


Fig.4. Analogue block of Acquisition, Processing and Communication Module

*Communication block*

The communication block uses the MC13202 transceiver. This is a short range, low power, 2.4 GHz Industrial, Scientific, and Medical (ISM) band transceivers. The MC13202 contains a complete 802.15.4 physical layer (PHY) modem designed for the IEEE® 802.15.4 Standard which supports peer-to-peer,

star, and mesh networking. The transceiver includes: low noise amplifier, 1.0mW power amplifiers (PA), onboard RF transmit/receive (T/R) switch for single port use, PLL with internal voltage controlled oscillator (VCO), on-board power supply regulation, and full spread-spectrum encoding and decoding. The device supports 250 kbps Offset-Quadrature Phase Shift Keying (O-QPSK) data in 2.0 MHz channels with 5.0 MHz channel spacing per the 802.15.4 Standard. The MC13202 transceiver communicates with the DSP using the full-duplex multichannel buffered serial ports McBSP0 configured as Serial Peripheral Interface, SPI0 in Fig. 3.

## 3. Software structure

The software of Acquisition, Processing and 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. 5 it is showed this structure of software.

The communication part contains the task of communication in network and a real time kernel including 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.

The type of objects of the dictionary of objects is in accord with Manufacturing Message Specification, ISO/IEC-9506. These objects correspond to the measurement results of those eight analog inputs, to the captured images by those two cameras connected to video inputs of Acquisition, Processing and Communication Module and so on.

The type of measurement results is integer or float and the type of captured images is an array of octet-string, where the octet-string are dates from each video line.

The real-time behavior of this software is imposed of task launcher, Fig. 6. The task launcher is the main loop of software and it is included in the real time kernel. Each task is tested if it is ready to be launched in execution. So each task has a status flag in FLGX byte and a mask flag in MASX byte. The bit 0 of FLGX, called flgx_0 is the status flag of task number 0, task_0, while the 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.

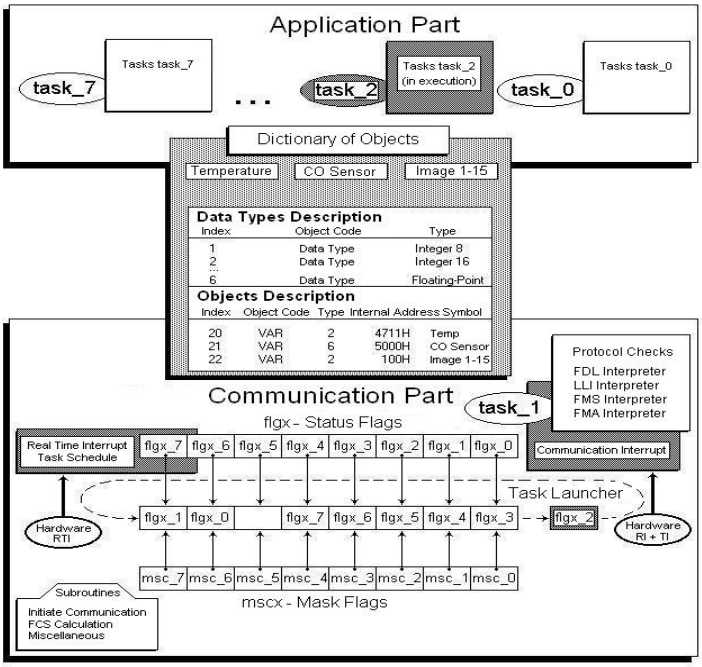


Fig.5. Software structure of Acquisition, Processing and Communication Module

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 witch should run periodically (to base time, multiple of base time, seconds, minutes, etc.). These tasks have by 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's counter of one task becomes zero, the status flag of this task will be set and the counter will be reinitiate. The task will be launch in execution only if it is active, its mask flag set.

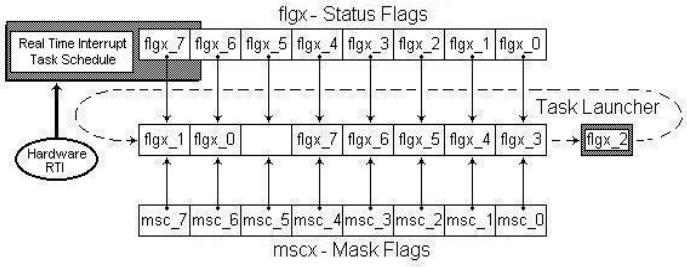

Fig.6. Task launcher

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. To each rotate right by Carry is incremented a pointer to a jumping table. This table contains the addresses to 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 in fact, and the test continue.

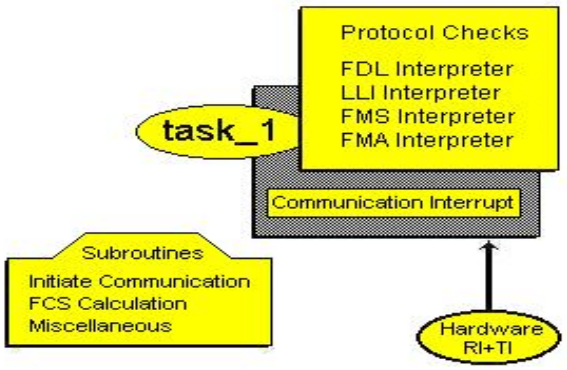


Fig.7. Communication task

The task number 1, task_1, is allocated for communication in network. In Fig. 7 is showed this task witch include all the functions for communication in network. This task with task launcher, task schedule, real time interrupt service routine and communication interrupt service routine form the communication part. For a network done and for participants to communication which occupy the same hierarchical position the software differences appears only to level of application part and to the dictionary of objects.

The task number 0 and the tasks number 2 to 7 are for application. They form the application part of software structure, assuring the functionality of Acquisition, Processing and Communication Module. Task_0, process results of A/D conversions and calculate averages for sixty samples. All eight analog inputs are measured one time per second. After all eight analog inputs are measured it is prepared the launch in execution of task_4, setting the bit flgx_4 (the status bit of task_4). Those eight analog inputs of Acquisition, Processing and Communication Module are used to measure two temperature sensors and six gas sensors.

Task_2 manages the video interface. It is responsible to prepare the response a one image demand. Task_3 supervises the process. This task is responsible for all decisions in the control and the monitoring of application. The task schedule prepares the launch in execution of this task to base time. Task_4 ensures the physical unit conversion of the results obtained in task_0. Task_5 resolves the USB interface. Task_6 solves the communication with the Power and Motion Module (SPI2 interface). Task_7 keeps the event history.

## 4. Conclusions

We consider that our approach is a fair one in this fast growing domain. Based on wireless technologies, we demonstrated that the key solution for developing real-time system (like emergency system for environment or disaster management) is to distribute the processing capabilities to the sensorial nodes and to transform that node in an intelligent autonomous one. In this way it is reduced the network traffic and it is increased the reliability of the entire system. Further we demonstrated that for achieving this goal, the proper chose for the processing core is a digital signal processor (DSP). Finally, we presented our node implementation which is tested on a DSP development board with encouraging results. The method based on the software partition in a communication part and an application part, developed around a task launcher and a Round Robin scheduler allows the achievement of new equipments which operate in real-time, for image and data acquisition, processing and communication.

## R E F E R E N C E S

[1] *K. Römer, M. Friedemann*, The Design Space of Wireless Sensor Networks, IEEE Wireless Communications 11 (6), p.54–61, 2004

[2] *T.Haenselmann*, An FDL'ed Textbook on Sensor Network, Published under the GNU Free Documentation License, 2006

[3] *K. Sohraby, D. Minoli, T. Znati,* Wireless Sensor Networks: Technology, Protocols, and Applications, John Wiley & Sons, p.3, 2007

[4] *C. Negrescu, D. M. Postelnicescu,* Teleoperation Controller Architecture for Mobile Robots, U.P.B. Sci. Bull., Series C, Vol. 70, No. 1, p.77-86, 2008

[5] *C. Negrescu,* A sensor fusion user interface for mobile robots teleoperation, U.P.B. Sci. Bull., Series C, Vol. 69, No. 3, p.169-178, 2007

[6] *R. Dobrescu, D. Popescu, M. Nicolae*, Design of a Modular Node for Tracking, Localization and Data Acquisition for WSN Applications, Proceedings 4th WSEAS International Conference on REMOTE SENSING (REMOTE'08) – Venice, Italy, November 21-23, p.51-56, 2008

[7] *S.M. Kuo, B.H. Lee, W. Tian,* Real-Time Digital Signal Processing, Second Edition Wiley, 2006.