

Flexible Firmware Architecture for Smart Transducers Networks

Pasquale Arpaia,

Dept. of Electrical Engineering and Information Technology
Università di Napoli Federico II - Via Claudio, 21 80125
Naples - Italy
pasquale.arpaia@unina.it

Pasquale Cimmino

Dipartimento di Ingegneria
Università del Sannio- Corso Garibaldi 107 82100
Benevento - Italy
pcimmino@gmail.com

Abstract— A firmware architecture, suitable for different applications of transducer networks, and easily configurable to both centralized and distributed customized requirements, is proposed. The key component is a flexible concurrent Finite State Machine not needing for a real-time operating system. The proposed solution was validated experimentally on two cases studies: (i) a telemedicine concentrated architecture for remote monitoring, diagnostics, and in-time prognostics of patients at home, and (iii) a distributed architecture for building remote monitoring and evolutionary diagnostics.

Keywords *Remote Monitoring, Data Acquisition, Wireless Sensor Networks, Open Systems or Open wireless Architecture.*

I. INTRODUCTION

In transducers networks, the advancement in integrated circuits technology has spawned and made very popular open-source, fully-programmable DSP-, FPGA-, and/or microcontroller-based platforms owing to their low cost [1]. In literature, the majority of proposals have been specifically designed for custom applications in the most of the cases [2]-[12]. On the other side, flexible platforms are designed to be comprehensive, and may fail to be effective in specific niche contexts or particularly demanding situations of real time or high accuracy [2].

Several examples are related to low-cost 8- to 32-bits microcontroller-based transducers networks, where hardware and firmware architectures are significantly and effectively customized to the specific application [3]-[12]. These firmware solutions are well suited for several applications [13], but unwilling of scaling and porting to other microcontroller-based platforms. A more flexible solution is presented in [14], where the hardware matches overcomes the limitation, and the firmware is easily customizable for the specific application. However, apart the improved flexibility, still ad-hoc architectures, devoted to low-cost microcontrollers, and strictly related to the platform device and unsuitable for high-performance systems, are adopted.

Hardware abstraction for a wide range of commercially available sensor nodes, with several communication protocols and a library of reusable components for developing wireless sensor networks applications, represents a more generic and application-independent approach. However, hardware

abstraction does not match on-site applications [15]-[18]. All the waveforms are packed and sent to the host PC, first through a suitable interface, and then through a fiber optic datalink. Finally, the host PC performs all the post-processing. This strategy is by far the simplest and most flexible solution but seems to be still missing for a distributed system. In all these solutions [19]-[21], the firmware is mainly focused on the monitoring system, by turning out to be poorly inclined both to distributed diagnostic architecture, and to high-precision data acquisition platforms.

In this paper, a Firmware architecture, based on a flexible concurrent Finite State Machine, not needing for a real-time operating system, and suitable for different kind of remote monitoring needs for wireless transducers networks, is proposed.

II. FLEXIBLE FIRMWARE ARCHITECTURE FOR TRANSDUCERS NETWORKS

In this section, (i) the *Flexible Multi-Behaviors Finite State Machine*, (ii) the *on-site high-precision data acquisition*, and (iii) the *remote monitoring* are described.

A. Flexible Multi-Behaviors Finite State Machine

The flexible firmware architecture proposed for wireless transducers networks is based on the paradigm Finite State Machine (FSM). A classic FSM is a behavioral model composed of a finite number of states, transitions between them, and actions. When more behaviors are provided, it is referred to as Multi-Behaviors Finite State Machine [22]. The Multi-Behaviors Finite State Machine approach has been extended suitably to measurement systems for satisfying different kinds of requirements (e.g. instruments of high and low performance, high and low sampling rate, high and low throughput, and so on) on diverse targets (e.g. DSP, low cost microcontroller, power PC, and so on).

At this aim, the Flexible Multi-Behaviors Finite State Machine has three basic components, (i) a measurement scheduler, (ii) an event receptor, and (iii) a main state machine evolving according to the events. Each data acquisition behavior satisfies a set of measurement and data processing requirements, and is translated into several FSM states [22].

1) Operating System schema with concurrent State Machine

A set of (extended) FSM shall be run simultaneously. The embedded firmware for systems based on this structure shall implement an operating system of concurrent processes (*Tasks*), each one corresponding to an FSM execution, whose activity is scheduled to happen concurrently.

The proposed operating system is based upon a foreground-background scheduling schema: a set of interrupt driven Task (the foreground) and a collection of non-interrupt driven Task (the background) are involved.

The firmware is broken down into standalone, well-defined tasks, so that a particular function or data structure belongs to a particular *Task*. Therefore, the *Task* logic is translated into a state machine, where each step represents a separate state [15][22].

2) Architecture

An architecture, capable of embodying the foreground/background model of interaction in its Finite State Machine policies, thus providing a principled mechanism to support sharing of measurement across multiple applications, is proposed.

The architecture is based on a foreground/background arrangement, with notifications of events and conditions occurring mainly via message queue passing [2]. A set of tasks well described in terms of finite state machine runs concurrently performing the related actions [15][22]. The basic task set consists of the following state machines for: (i) controlling the input data messages for user interaction and instrument setting (e.g. sampling frequency, gain, and so on), (ii) acquiring and filling data into a measurement queue, (iii) the measurement algorithm, and (iv) the output data flow.

Interfaces are essentially composed by message queues, for input data messages, measurement raw data, and final results.

Starting from an initial state (default = IDLE), the single *Task* fetches the table entry corresponding to the state, remaining into it until a triggering event associated to the state occurs. The *Task* then evaluates the associated conditions on the message queue, and if the case, triggers the associated action and register status updates (if any), executes the state transition, and fetches the new table entry for such a destination state. The engine workflow allows code switching to be supported easily: analogously as usual multi-threading, this entails the time-shared execution of different *Tasks* simultaneously on the device.

3) State Chart

The State Chart of Fig. 1 highlights the application of the proposed firmware architecture to smart transducers. The Multi-Behaviors Finite State Machine evolves according to user commands and fault conditions. In the state *Ready*, a new command is detected and the corresponding action is executed by changing the Instrument state. Once the operation is completed, the transducer is reported to the state *Ready*.

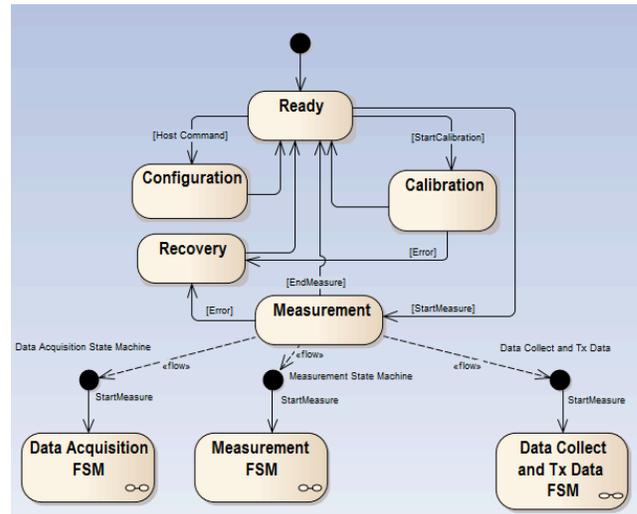


Fig. 1 – State chart of the Flexible Multi-Behaviors Finite State Machine for smart transducers

In case of faults, the instrument moves to the state *Recovery*. In the *Calibration* state, the calibration procedures are executed. The *Measurement* state is in charge of performing the measurement algorithms by triggering the Measurement State Machine.

The main problem related to on-line data processing is the time constraint, therefore the Data Acquisition State Machine operates at foreground level: the ISR belonging to the interrupt received at each ADC conversion is in charge of filling the measurement queue with the ADC numerical values and the trigger times, when present. Simultaneously, the events for the Measurement State Machine transitions are generated and the actions required for the computation and the board management can be performed within one ADC sampling period.

III. CASE STUDIES

A. Concentrated Monitoring for Home Care Telemedicine

Home care telemedicine systems are playing an increasingly important role, providing targeted services at low cost and owing to an improved effectiveness and efficiency.

In this section, the multi-data acquisition Finite State Machine Firmware for an integrated telemedicine platform for monitoring, diagnostics, and in-time prognostics of patients directly at their homes is illustrated. In particular, (i) the measurement requirements, (ii) the firmware design, (iii) the implementation, and (iv) the validation results are detailed.

1) Measurement Requirements

At patient's home, the domiciliary monitoring is composed by the *Biomedical Subsystem*, a wireless network of clinical transducers (e.g., arterial pressure, glucose, electrocardiogram, and so on), and domestic sensors for environment supervision (e.g. motion detection). A *Local Main Board*, acts as network coordinator, and performs some pre-processing onto the clinical measures before sending the relevant data to the

Central Monitoring Station via GPRS channel. A web-based application allows data access to caregivers and clinical doctors. Furthermore, besides a simple alarm generation triggered by a threshold overcome, both on Local Main Board and Central Monitoring Station side, an *Automatic Prognostics* provides a preliminary assessment of risk of the patient monitored for a given pathology [23].

Main requirements for the firmware of the Local Main Board are: (i) coordination of the wireless sensors' network; (ii) data storing of the biomedical input data; (iii) basic biomedical data analysis for immediate alarm signaling in case of anomalies and malfunctioning; (iv) scheduling of the biomedical measurement based on an event timetable; (v) handling of GPRS modem for Central Monitoring Station communication.

2) Firmware Design

In the following, the design of the two main components of the home care telemedicine, the *Local Main Board* and the *Home Care Monitoring Device*, are illustrated.

Local Main Board. The Data Acquisition State Machine is in charge of filling the measurement queue with the measuring data received from the wireless ZigBee network [24]. It operates at foreground level, parsing the input data and triggering the Measurement State Machine transition from *idle* to *measure* in order to execute the algorithm. It mainly checks the threshold overcome triggering immediately the Host Transfer module that send alarms both to Remote Station via GPRS and to caregivers via SMS. The Time scheduling State Machine handles as Input Messages. It is responsible for the state switching of an event timetable which schedules: (i) the patient interaction with the telemedicine system by executing a predefined set of measurements (ETT Table of Fig. 2); (ii) data delivering to the Remote Station (TX Table of Fig. 2). Incoming messages from the Remote Station are only for administrative purposes like firmware configuration or tables update.

Home Care Monitoring Device. It covers a wide range of clinical follow-ups from cardiac to pneumological diseases, and from diabetes to hypertension and obesity [25].

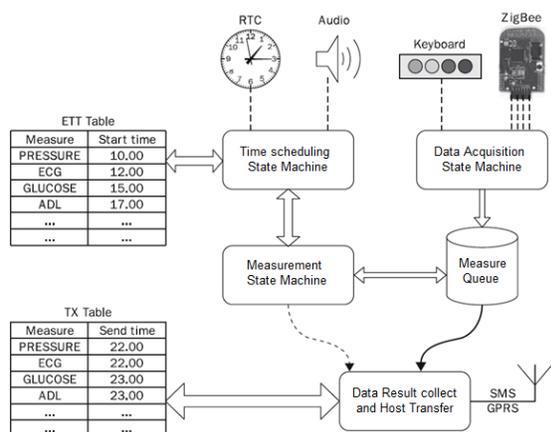


Fig. 2 – Firmware Architecture of the Local Main Board.

The patient itself directly executes the measures after explicit request operated by the Local Main Board via audio message. In addition to commercial devices, the network consists of two custom microcontroller-based devices: (i) an activity monitor device [26]; and (ii) a posturographic plate for home care tele-rehabilitation exergaming, consisting of four force sensors connected to the analog microcontroller inputs [28]. Both the instruments are low-cost microcontroller based ZigBee end-node devices whose firmware architecture is based on the above component diagram. In the following, the firmware design for the posturographic plate device is detailed. According to a foreground/background schema, three State Machines are implemented: (i) Data Acquisition State Machine for ADC data acquisition and packing, (ii) Measurement State Machine for algorithm execution, and (iii) Host Transfer for data transmission. The firmware has also in charge the network handling by a dedicated State Machine, scheduled in time-sharing mode.

The data acquisition of the 4 sensors is operated at foreground level: at each ADC interrupt, the data is filled into the Measure Queue and the input multiplexer switched to the next channel, thus each channel is sampled every 4 ADC interruptions. Since the data processing is performed by the Host PC, the Measurement State Machine has only in charge of checking anomalies and error during the acquisition. Data result are then collected and coded in order to be sent to the coordinator node according to the ZigBee network protocol.

3) Implementation

Local Main Board. The Renesas High-performance Embedded Workshop for 16-bit microcontroller environment was used for developing the Local Main Board firmware. The C code implemented for the Transducer State Machine by means of the switch/case statement.

The Timing Scheduling State Machine operates at foreground level: the ETT look-up table (Fig. 2) with timings expressed in double format has been used in order to be easily compared with the local RTC values. As a result, (i) the corresponding audio message is selected in order to inform the user about the medical measurement to perform; (ii) the Transducer State Machine is triggered in measurement state to execute data elaboration. Due to time constraint in terms of ZigBee network input data throughput and alarm signaling, the on-the-fly elaboration has been adopted for this case, too.

Home Care Monitoring Device. For the implementation, the module Texas Instruments eZ430-RF240 [27] has been used. It is composed of the embedded stack protocol ZigBee transceiver CC2480 and the 16-bit microcontroller MSP430F2274, provided by 4 channels ADC. The end-node device selects the ADC sampling rate up to 20 Sa/s per channel. The data between two ADC samples are processed by a ISR assembler code in order to monitor execution time down to single instruction level.

4) Validation Results

The integrated telemedicine platform based on flexible Multi-behavior Finite State Machine firmware has been tested in the environment of the project PADIAMOND [28] by means of The Healthcare Failure Mode & Effects Analysis (HFMEA)

method. As extension of the classic FMEA applied to clinical frameworks. HFMEA consists of 6 main stages[25][29]: (i) Define the HFMEA Topic (clear definition of the process to be studied), (ii) Assemble the Team (multidisciplinary team expert(s)), (iii) Graphically Describe the Process (identifying process steps), (iv) Conduct a Hazard Analysis: List of potential failure modes with analysis of causes and effects to identify potential failure modes, determining the relevant probability P of occurrence O for each of the potential causes and the severity S , (v) Actions and Outcome Measures (action plan for each failure mode), (vi) Follow-up on Actions Taken. As a final result, for each failure mode the FMEA Risk Priority Number is evaluated. Table 1 shows the Hazard analysis step for the biomedical measurement. The PNR results are reported in Table 2. The product factors Probability, Occurrence and Severity are established considering the range [1..4].

B. Distributed Monitoring for Building Energy Saving

In this section, the firmware architecture for a transducer network with a distributed diagnostic procedure, based on a micro-genetic algorithm [30], is proposed. In particular, (i) the measurement requirements, (ii) the firmware design, (iii) the implementation, and (iv) the validation results are illustrated.

Table 1 – Biomedical wireless transducer measurement potential failure mode, with cause, effects and possible barriers.

Measure Stage	Potential Failure Mode	Potential Cause	Potential Effect	Possible Barrier
Plates setup	Correct patient positioning	Incompetent staff	No measure	
		Patient no self-sufficient	No measure	
		Inadequate User guide	No measure	
Tx Step	ZigBee link loss	Antenna connection	No data TX	
		High distance	No data TX	
	Environment Signal noise	White and Env Noise overlap	Data TX Error	Noise filtering

1) Measurement Requirements

The evolutionary approach of centralized multiple-faults diagnostics in experimental physics is extended to a distributed transducer networks monitoring: the basic idea of the proposed distributed micro-genetic algorithm (MGA) version concerns the possibility to run each micro-cycle of the basic algorithm on each node of a network. The micro-cycles run independently each other, in parallel and asynchronous way. Therefore, on a network of N nodes (each one with a processor and memory capability), the idea is to implement a distributed MGA with N micro-cycles, each one running on a network node.

Table 2 – Biomedical wireless transducer measurement PNR evaluation

Potential Failure Mode	Probability [1..4]	Occurrence [1..4]	Severity [1..4]	PNR
Correct patient positioning	3	3	3	27
ZigBee link loss	2	4	2	16
Environment Signal noise	2	4	2	16

The external layer works only after each micro-cycle has terminated, evaluating a certain number of best solutions and determining the criterion of the algorithm re-initialization of the next micro-cycle. At the end of each cycle the nodes of the network are all free, therefore an arbitrary node is chosen when all the cycles are completed.

2) Firmware Design

The Multi-Data Acquisition Finite State Machine firmware has been proposed to approach the diagnostic architecture for a project committed to remote monitoring and evolutionary diagnostic for civil buildings [30].

The edifices were monitored owing to their criticalities, by installing a network of microcontroller based transducers (smart transducers) with the corresponding diagnostic units which belong mainly to the installed plants, such as Lift, Heating, Air handling, Domotic, Electrical/Alarm.

The firmware architecture of each smart transducer is the same as represented in Fig. 1: the Data Acquisition State Machine fills the measure queue and triggers the Measurement State Machine transition into the state *measure*, where the single-cycle diagnostic algorithm is executed. The anomalies are then collected and shared among the other smart transducers by means of the Host Transfer module. The distributed approach is completed by the *master* node, in charge of the algorithm synchronization, as well as of the communication with the Remote Station, which provides a plant remote view and the relevant diagnostics output.

In this way, the measurement and diagnostic algorithm are diverse according to the monitored plant for the transducers. In the State Machine of the Electrical/Alarm plant, for the diagnostics of the power quality of the three-phase electrical system, the harmonic analysis (FFT) is carried out.

A power electrical black out, or a fault to the line transformer have to be diagnosed by starting from the detection of problems on the voltage lines R, S, T vs the neutral N , or for a too low power factor $\cos(\phi)$.

3) Implementation

The smart transducer has been implemented by using the linux based AMR9 processor Technologic System TS7200 [32]. It is provided by (i) PC104 bus for peripheral connection,

(ii) an 8-channels 12-bit ADC, and (iii) serial interface for GSM modem connection. The firmware developed in C language implements again the Instrument State Machine. During the configuration stage, the GPRS link with the RSS is established and the distributed network setup is performed: the Message Passing Interface (MPI) library has been used for the secured network transducer' connection. The measurement state machine is in charge of algorithm and local anomalies vector filling. The micro-genetic algorithm is run in time boxed mode in order to execute the vector sharing and the best solution evaluation.

4) Validation Results

The experimental validation of the smart transducer network based on flexible Multi-behavior FSM firmware has been deployed on a MONDIEVOB project application [31]. Two main stages have been assessed: (i) functional tests, which carry out checking on Measurements and network communication, (ii) dMGA validation, which includes tests on diagnostics problems. For example, overvoltage line and/or too low power factor $\cos(\phi)$ can be detected by the electrical plant diagnostic unit.

Fig. 3 illustrates the MGA code: number of run at row 75, generation loop at row 82 within which the solution is populated and then shared via MPI library. Population p is a structure which contains all the information required by the algorithm [30]: (i) best and worst solutions, (ii) population size, (iii) fitness, (iii) micro-evolution step, (iv) crossover and mutation probabilities, and (v) random seed. For the distributed algorithm validation up to 3 diagnostic units in parallel with different input parameters were used.

In particular (i) the population size of 7, 10, 15 and 24, (ii) the generation number of 30, 50 and 100, (iii) up to 20 runs were chosen. The resulting faults and likelihood were successfully compared with brute force algorithm executed for the same test case.

```

74 // #run loop
75 for (i = 0; i < p->maxNumRun; i++)
76 {
77     // NEW RUN
78     next_best_chrom = (int *) calloc (p->lchrom, sizeof(int));
79     prec_best_chrom = (int *) calloc (p->lchrom, sizeof(int));
80
81     // #generation loop
82     while (p->gen < p->maxGen)
83     {
84         p->gen++;
85         // new generation
86         generation(p, p->gen);
87         // execute statistic solution
88         statistics(p, p->np);
89         //
90         if ((p->gen % p->cStep) == 0)
91         {
92             if (my_id == 0)
93                 show_pop_stats(p);
94             // MPI communication step
95             comunica_up(p, p->np[p->maxi].chrom, next_best_chrom);
96             comunica_down(p, p->np[p->maxi].chrom, prec_best_chrom);
97             // population adding
98             insert_in_pop(p, next_best_chrom, prec_best_chrom);
99         }
100         // next generation
101         p->microGen++;
102     }
103     // evaluate best solution
104     collect_best_ones_in_ring(p);
105     report_result(p);
106 }
107 // MPI share solution
108 sendFailure (best_chrom_over_runs, p->lchrom);
109

```

Fig. 3 – Micro-genetic algorithm implementation

REFERENCES

- [1] F. John Park, S. Mackay; Practical Data Acquisition for Instrumentation and Control Systems. British: Newnes An imprint of Elsevier, ISBN 07506 57979, 2003.
- [2] Bianchi G., Gallo P., Garlisi D., Giuliano F., Gringoli F., Timirello I.; MAClets: Active MAC Protocols over Hard-Coded Devices; Proceeding CoNEXT '12 Proceedings of the 8th international conference on Emerging networking experiments and technologies archive 229-240 ACM New York, 2012.
- [3] Juca, S.C.S.; Carvalho, P.C.M.; Brito, F.T. A Low Cost Concept for Data Acquisition Systems Applied to Decentralized Renewable Energy Plants. Sensors 2011, 11, 743–756.
- [4] Sumphao, T.; Thanachayanont, C.; Seetawan, T. Design and Implementation of a Low Cost DAQ System for Thermoelectric Property Measurements. Proced. Eng. 2012, 32, 614–620.
- [5] Hjertaker, B.T.; Maad, R.; Schuster, E.; Almas, O.A.; Johansen, G.A. A data acquisition and control system for high-speed gamma-ray tomography. Meas. Sci. Technol. 2008, 19, doi:10.1088/0957-0233/19/9/094012.
- [6] Hwang, Y.S.; Kim, H.J.; Park, H.; Kang, H.; Kim, W.; Kim, S. Development of Embedded DAQ System for Flux Counting. J. Nucl. Sci. Technol. 2008, 45(Supplement 5), 582–585.
- [7] Singh, N.M.; Sarma, K.C.; Singh, N.G. Design and Development of Low Cost Multi-Channel USB Data Acquisition System for the Measurement of Physical Parameters. Int. J. Comput. Appl. 2012, 48, 47–51.
- [8] Gupta, R.; Bera, J.N.; Mitra, M. Development of an embedded system and MATLAB-based GUI for online acquisition and analysis of ECG signal. Measurement 2010, 43, 1119–1126.
- [9] Zoric, A.C.; Perisic, D.; Obradovic, S.; Spalevic, P. PC-Based Virtual DTA Recording System Design. Prz. Elektrotech. 2011, 87, 156–160.
- [10] Yavuz, C.; Yilmaz, S.; Kaya, M. The Design of Computer Controlled Cold and Hot Therapy Device with Thermoelectric Module. J. Med. Imaging Health Inform. 2013, 3, 221–226.
- [11] Kumar, A.; Singh, I.P.; Sud, S.K. Energy Efficient Air Quality Monitoring System. In the 10th IEEE Conference on Sensors, Limerick, Ireland, 28–31 October 2011; pp. 1562–1566.
- [12] Kumar, A.; Singh, I.P.; Sud, S.K. Energy Efficient and Low-Cost Indoor Environment Monitoring System Based on the IEEE 1451 Standard. IEEE Sens. J. 2011, 11, 2598–2610.
- [13] M. Abdallah, O. Elkeelany, A. T. Alouani, "A low-cost stand-alone multichannel data acquisition, monitoring, and archival system with on-chip signal preprocessing," IEEE Transactions on Instrumentation and Measurement, vol. 60, no. 8, (2011), pp. 2813-2827.
- [14] Hercog, D.; Gergič, B; A Flexible Microcontroller-Based Data Acquisition Device. Sensors 2014, 14, 9755-9775.
- [15] Beaubrun R.; Smolau S.; Experiments for evaluating sensors' precision in wireless sensor networks. International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.3, No.3, June 2012.
- [16] Moteiv, "Tmote Sky ultra low power IEEE 802.15.4 compliant wireless sensor module datasheet," 2006.
- [17] "TinyOS: An open-source OS for the wireless embedded sensor networks," <http://www.tinyos.net>.
- [18] Castoldi A.; Guazzoni C.; Signorelli D.; Hardware, Firmware and Software Architecture of the DAQ for High-Resolution Position-Sensing Silicon Drift Detectors With Multiple-Pulse Processing Capability; IEEE Transactions on Nuclear Science, vol. 55, no. 5, October 2008.
- [19] Benganem, M. A low cost wireless data acquisition system for weather station monitoring. Renewable Energy 2010, 35, 862-872.
- [20] Anwari, M.; Hidayat, A.; Hamid, M.I.; Taufik, I. Wireless data acquisition for photovoltaic power system. In Proceedings of International Congress: Telecommunications Energy Conference, Incheon, Korea, October 2009; pp.1-4.
- [21] Rosiek, S.; Battles, F. A microcontroller-based data-acquisition system for meteorological station monitoring. Energ. Conv. Manage. 2008, 49, 3746-3754.
- [22] Xuesen Lin; Multi-Behaviors Finite State Machine. IEEE Information, Computing and Telecommunication, 2009. YC-ICT '09.
- [23] Arpaia, P., Manna, C., Montenero, G. and D'Addio, G.; In-time Prognosis based on swarm intelligence for home-care monitoring: A case study on pulmonary disease; IEEE Sensors, 2012 (3): 692 – 8.

- [24] Official Web site, ZigBee Alliance. <<http://www.zigbee.org/>>.
- [25] G. D'Addio, P. Cimmino, C. Manna, P. Arpaia, "A Telemedicine Home Care Based Activity Monitor Device", IEEE International Symposium on Medical Measurements and Applications, Bary, Italy, May 30-31, 2011.
- [26] P. Cimmino, et al., "Integrated Telemedicine Systems: Patient Monitoring, In-time Prognostics, and Diagnostics at Domicile", The design and manufacture to medical devices, Biohealthcare Publishing (Oxford) Limited, 2012 (research book chapter).
- [27] Texas Instruments technical reference, www.ti.com
- [28] Project PADIAMOND, "Patient Diagnosis and Monitoring at Domicili", Granted by EU in the framework of POR 3.17 ICT of Regione Campania (Italy).
- [29] P. Arpaia, P. Cimmino, E. De Matteis and G. D'Addio, "A Low-Cost Force Sensor-based Posturographic Plate for Home Care Neurophysiological Telerehabilitation Exergaming", Measurement, Vol. 51, N. 1, May 2014, pp. 400-410. 10.1016/j.measurement.2014.01.031.
- [30] P. Arpaia, P. Cimmino, M. Girone, G. La Commara, D. Maisto, C. Manna, M. Pezzetti; Decentralized diagnostics based on a distributed micro-genetic algorithm for transducer networks monitoring large experimental systems; AIP Review of Scientific Instruments, 85, 095103, 2014.
- [31] Project MONDIEVOB, "Buildings Remote Monitoring and Evolutionary Diagnostics", Granted by EU in the framework of POR 3.17 ICT of Regione Campania (Italy).
- [32] Technologic Systems technical reference, <https://www.embeddedarm.com/products/board-detail.php?product=TS-7200>