On-line self-diagnosis based on power measurement for a wireless sensor node

Van-Trinh Hoang, Nathalie Julien, Pascal Berruet

On-line self-diagnosis based on power measurement for a wireless sensor node

Van-Trinh HOANG, Nathalie JULIEN, Pascal BERRUET
van-trinh.hoang@univ-ubs.fr, nathalie.julien@univ-ubs.fr, pascal.berruet@univ-ubs.fr
Lab-STICC Research Center, University of South-Brittany, BP 92116, 56321 Lorient, France.

***Abstract*—A self-diagnosis design for wireless sensor node is a big challenge for designers. Particularly, when sensor nodes are deployed in harsh environment, it's very difficult for human to intervene in case of hardware failure of node components. In this paper, we present our novel self-diagnosis for the discrete event systems (DES) like sensor node, which includes a complete strategy of self-diagnosis based on both functional and non-functional tests. Our approach helps sensor node to detect automatically its component failure, and then to take a corrective solution. And then, the implementation of our approach in the real material, which is based on the results of power measurement of node component, is presented. Finally, we also indicate how to optimize our self-diagnosis to make it more energy-efficient.**

Index Terms: *discrete event system, hardware failure, energy-efficiency, self-diagnosis, reconfiguration, PAM, FPGA.*

I. Introduction

Major breakthroughs in the deep sub-micron technology have led to the emergence of ubiquitous computing [1]. Small and inexpensive devices like Wireless Sensor Network (WSN) have a lot of attention in recent years. Thanks to its portability, it may be carried out on everywhere from the human body to be deeply embedded in the environment. WSN can easily be deployed in large space with dramatically less complexity and cost compared to wired networks. Additionally, sensors can self-organize to form routing paths, collaborate on data processing, and establish hierarchies. The WSN is also re-configurable by easily adding and removing sensor nodes. Thus it is the most favorite candidate for many applications such as area monitoring, environment monitoring, and industrial monitoring.

The wireless sensor nodes interface with the environment using actuators and sensors, have a limited processing capability and memory storage, and communicate via wireless links. Moreover, because these devices are battery-powered, hence their autonomy is determined by their battery life. Therefore, energy-efficiency has emerged as an important design constraint. Nowadays, sensor nodes are usually deployed in harsh environment where there are many potential hardware failures, and hence very difficult for human to intervene in case of hardware failure of node component. In case of failure, battery energy will be wasted if sensor node continues supplying its failed component.

Additionally, in several applications such as fire detection, hazardous gas detection, intrusive-detection in security zone, etc, the reliability of sensor node is also an important metric to take into account. Therefore, an integrated approach including the self-diagnosis, in which the node is able to detect exactly its component failure, is essential for energy-efficient and reliable application development in such a network. Based on this approach, the sensor node could take a suitable corrective solution to make them less vulnerable. Besides, our approach also alleviates the maintenance cost of these networks.

This paper is organized as follows. In section II, the background of our work is presented. We introduce the hardware configuration of sensor node, and our approach including the node self-diagnosis for each component in section III. Section IV shows how our approach will be implemented on a physical node. And section V contains conclusion and future works.

II. Background

A. Fault Detection Isolation (FDI)

The problem of identifying crashed nodes in a distributed sensor network has been extensively studied in literature [4-6] for traditional wired network, where energy consumption is not an issue. Contrary to the case of wired network, WSN requires a diagnosis protocol itself should be as efficient energy as possible. Thus, the diagnosis approaches presented in [4-6] are not suitable for WSN.

Ringwald and Rmer introduce their passive inspection method [8-9] by overhearing and analyzing the message exchange between the nodes, so as to detect a problem in wireless sensor network. Their method does not require any modification of sensor network but need materials complementary. Moreover, they do not specify any method to fix these problems.

Stefano Chessa and Paolo Santi [10] present their crash fault identification in wireless sensor network called WSNDiag by transmitting the "I'm alive" message between the neighbor nodes to identify the faulty nodes. But they don't indicate exactly the reasons of node failure. And their method also increases the power consumption in transmission while wireless node is strictly battery energy constraint.

Besides, lots of other works concentrate particularly in error-detection and fault-tolerance of data sensing and processing of sensor or processor such as [11-14]. But these approaches do not deal with the hardware failure of each component in sensor node.

B. Our objective

As mentioned in previous section, hardware failure of sensor node is one of the most critical point for designers in term of reliability and energy. For example, our sensor node is equipped with two sensors, while the first one is usually enable to capture environment data, the second one is only activated

TABLE I
ISSUES AND CORRECTIVE SOLUTIONS FOR SENSOR NODE

Problem	Software and Hardware causes					Energy causes		Solution
	Down Processor	Down Ram	Down IAS	Down RTM	Soft Bug	Low Energy	Energy Depletion	
Dead Node	X							PAM enables FPGA processor to replace processor
		X						PAM enables FPGA memory to replace RAM/Flash memory
							X	Wait for recharging battery
Malfunctioning Node			X					PAM changes mode of operation to relay point
				X				PAM changes mode of operation to local processing
					X			Processor reboots
						X		PAM selects consistent mode of operation and wait for battery recharge

by users to verify the obtained data. In case of failure of the first sensor, energy is wasted if it is still supplied with voltage. Additionally, the node reliability is also downgraded when the node continues using it. Therefore, if the node turns off the power supply for the first sensor and switches to the second sensor, there will be no wasted energy and node reliability is maintained.

Thus, our goal is to provide novel self-diagnosis based on functional and physical tests to detect hardware-failure for each component in wireless sensor node. This method can identify exactly which node component is down. That leads to take a suitable solution to correct it.

C. Hardware configuration and operating modes of our sensor node

The hardware configuration of our self-reconfigurable sensor node is illustrated in Figure 1. It consists of the original components such as a processor, a RAM/FLASH memory, an Interface for Actuator and Sensors (IAS) to interface with the environment, a Radio Transceiver Module (RTM) to transmit and receive data, and a battery with power switches (DC-DC converters). Besides, a Power and Availability Manager (PAM) combined with a configurable zone of FPGA are included in sensor node, while the first one is considered as the intelligent part for the best use of energy, auto-diagnosis and fault-tolerance, while the other enhances the availability of sensor node. To mitigate the data conflict, two First In First Out (FIFO) buffers are used in which CapturingBuffer stores the captured data, and ReceptionBuffer saves the data sent by other nodes.

Based on the utilization of PAM and FPGA, our sensor node can react against the node issues as illustrated in Table I. In this paper, we do not deal with the software bug problem, we focus on the hardware failure of each component that is more serious. All operating modes of our node are described in Figure 2 using Finite State Machine (FSM), because it is suitable to model the discrete event system as such a sensor node.This FSM model consists of a set of states and transitions. When each state represents a particular mode (On-Duty, Performance Enhance, Monitoring, Observation,...), and each transition represents one or more discrete events that make the transition from one operating mode to another one. The FSM model is divided into two parts marked with blue border and green rectangle. The blue one mentions the

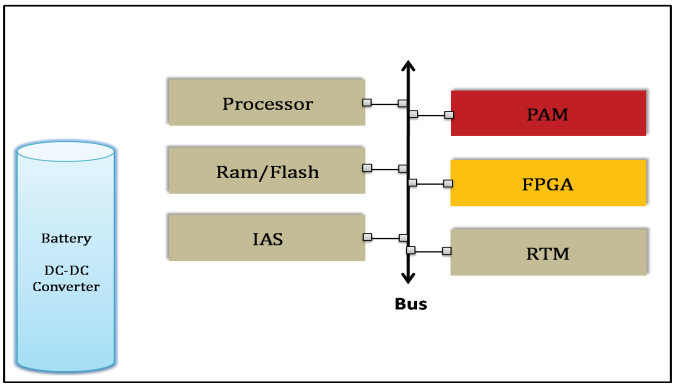

Fig. 1. Hardware configuration of wireless sensor node

availability management of the system, while the green one relates to the compromise between performance and energy-efficiency. At beginning, FSM model enters in the Monitoring state, only Processor, Ram, Sensor1, and Receiver are active. The operating modes of our sensor node are described more in detail in [3].

III. SELF-DIAGNOSIS FOR SENSOR NODE

Our wireless sensor node has limited processing capacity, and memory storage. Moreover, its autonomy is determined by battery lifetime, thus, energy-efficiency has emerged as an important design metric, even in case of auto-harvesting energy because the energy from the environment is generally unpredictable, discontinuous, and unstable. Additionally, the more energy is saved, the more node lifetime is extended. Our approach helps sensor node to run self-diagnosis and detect automatically hardware failure of each component occurred in it. Thus, our sensor node could react against these problems by selecting the appropriate corrective solutions to make it less vulnerable. Therefore, wasted energy that is used to supply failed components will be saved, which leads to extend the battery lifetime. By mean of utilization of our PAM block as described in the previous section, our self-diagnosis of hardware failure for each node component is realized using the functional and physical tests as listed in Table II.

In the next sub-sections, the self-diagnosis process of each component is described in detail using Petri Net. Since the behavior of node system refers to Discrete Event System,

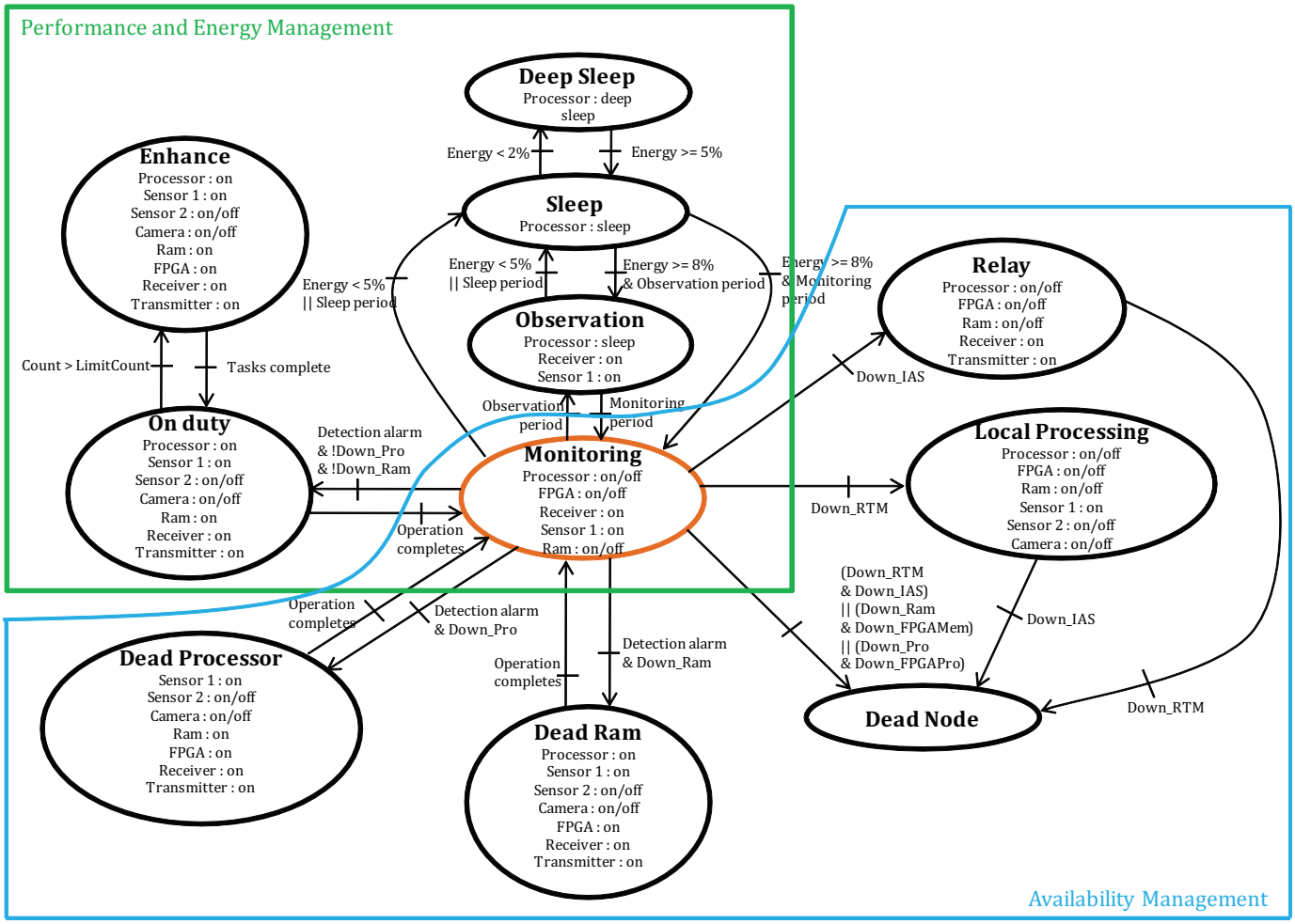

Fig. 2. FSM model of operating modes for sensor node

TABLE II
SELF-DIAGNOSIS TYPE FOR EACH NODE COMPONENT

Component	Self-diagnosis type
Processor	Functional test
Ram	Functional test
Sensor	Functional test
Radio transceiver	Functional + physical test

Petri Nets (PNs) [2] are suitable for modeling behavior of communicating and synchronized processes. These Petri Net models allow us to perform the concurrent operations and asynchronous events of node system, and then to compute the availability of each component in the future work by using SPNP tool as cited in [15].

A. Node self-diagnosis for processor

The sensor node is initially at Monitoring mode as illustrated in Figure 2. To run self-diagnosis for processor based on functional test, our PAM block sends a beacon signal and waits the feedback from the processor. The self-diagnosis process of processor is illustrated in Figure 3 using Petri Net.

The above Petri Net includes places P: (as depicted as circle) and transitions T: (as depicted as black bars or white rectangles). A transition is connected to its input places by input arcs shown as directional arrows. Conversely, output arcs drawn from the transitions to its output places. In the Figure 3, the transitions are modeled as immediate ones (black bars), when the others are modeled as timed ones (white rectangles), because the firing time of immediate transition is small comparing to timed transition. When the node system is in Monitoring mode, if the data processing in processor is longer than a time interval T_1, i.e. no token is present in the place *P:SucProcess*, our PAM block will check the processor availability as depicted in Figure 3. In this case, a token appears in the place *P:NotProcess* (i.e. processor may fail in data processing) or in *P:ProProb* (i.e. processor may be down). Next, the PAM block sends a beacon signal to processor, and then waits its response. If our PAM does not receive any feedback from the processor, the processor is considered as failed (corresponding to the appearance of a token in the place *P:DownP*).Otherwise, it is still available, i.e. the token is back to the place *P:UpP* that allows the processor

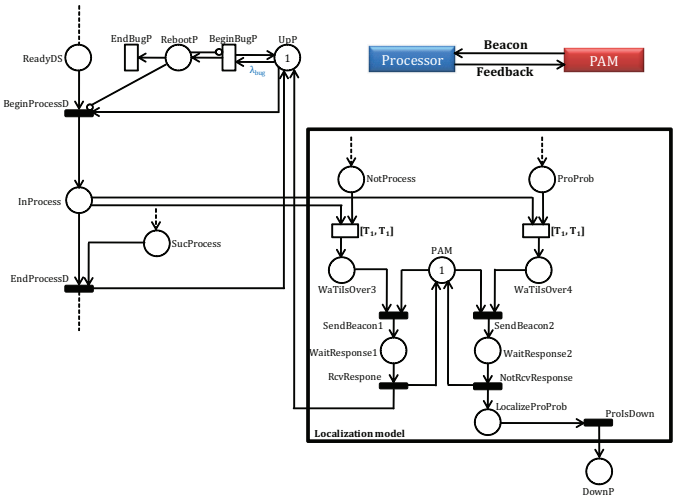

Fig. 3. Self-diagnosis for node processor

to continue to operate. The self-diagnoses for RAM memory and sensors (IAS) are nearly similar as the processor diagnosis, thus we do not show it in this paper.

B. Node self-diangosis for radio transceiver module

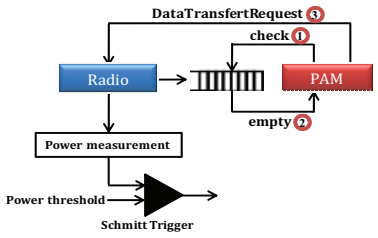

Fig. 4. Self-diagnosis for node radio module

In case of radio transceiver diagnosis, the functional test combined with the physical test based on the power consumption of radio device is applied as illustrated in Figure 4, because the case where the destination node is down is taken into account. For example, the radio transmitter of source node sends data and PAM waits the feedback of destination node in radio buffer. If the destination node is down, it can not send acknowledge signal back to the source node. Therefore, the functional test is not sufficient enough to detect correctly the failure of sensor node.

When the reception buffer is empty, the node processor enables the watchdog timer that is associated with a time interval of T_2. The failure localization procedure of radio transceiver is modelled in Figure 5. If no data is sent to the sensor node during this interval, i.e. a token appears in the place *P:NotRcvPack* or *P:RadioProb*, which leads to the expiration of watchdog timer (a token appears in the place *P:WaTiIsOver5* or *P:WaTiIsOver6*). Then, the PAM block makes a request of data transmission to a neighbor node to check the availability of radio module, this step is called as such a functional test.

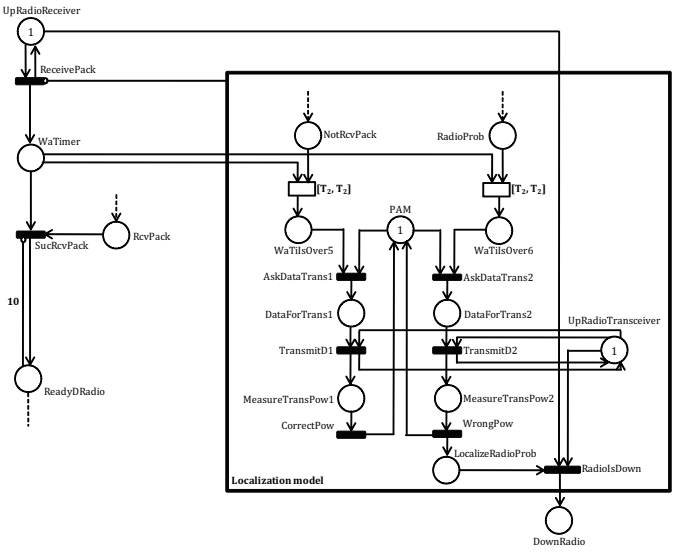

Fig. 5. Self-diagnosis for node radio module

When the data transmission is executed, its power consumption is measured by an embedded electronic device, and then this power value is compared to a power threshold using a *Schmitt trigger* (as seen in Figure 4). When the output result of this trigger is true (i.e. the measured power value is correct), the radio transceiver is still available, otherwise it is considered as failed (i.e. the measured power value is wrong that leads to the appearance of a token in the place *P:DownRadio*), this step is called as such a physical test. The next section presents the efficient implementation in term of energy of node self-diagnosis in real material that will be realized in our future work.

IV. Physical Implementation of Node Self-Diagnosis in Real Material

In this paper, the power consumption of node materials, which are used in the application of hazardous gas detection for area such as harbor or warehouse, are measured. This application is developed by the ERYMA Company that offers an expertise in material solutions and security system for the prevention, monitoring, maintenance and remote control. The original sensor node consists of a PIC24FJ256GB110 processor, a M48T35AV ram memory, a Miwi radio transceiver module, two Oldham OLCT 80 gas detectors, the power switches (LM3100 and MAX618), and a battery. This sensor node has not yet integrated our material approach including the PAM block combined with FPGA. The measured power consumption of node materials is given in the Table III. And then using the CAPNET-PE simulator [7], the node autonomy can be estimated.

The FPGA IGLOOV2 of Actel manufacturer will be selected to be implemented in our sensor node due to its reliability and ultra-low power consumption [16]. Actel manufacturer also offers soft-CPU such as ARM Cortex-M1, or core8051s, or coreABC that are available in FPGA. In our case, core8051s

TABLE III
POWER MEASUREMENT OF NODE COMPONENT

Component	Power (mW)
PIC24FJ256GB110	36
M48T35AV memory	150
Oldham OLCT 80	867
Miwi Radio transmitter	130
Miwi Radio receiver	70

is selected due to its trade-off between performance/energy consumption. This soft-CPU is activated when the performance enhance is needed or to replace the main processor in case of its failure. Additionally, our PAM block will be also programmed to be implemented in FPGA using small resource, since PAM block only checks the state of other components, and does not realize any complex computation. Hence, its power consumption is much less than other components (see Table IV). As being known, the power consumption of an electronic component is computed as follows:

$$P_{total} = P_{static} + P_{dynamic} \quad (1)$$

The static power Pstatic is product of the power supply voltage and the static current, which itself includes two dual components: leakage current and through current. Leakage currents are parasitic effects, while through currents occur in normal operation and are due to transistors being continuously operated in their saturation. The dynamic power is caused mainly by switching activity of charging and discharging the internal cell capacitance. Based on the power calculation methodology in ACTEL datasheet [16], the consumed power of soft-CPU core8051s and our PAM is given in the Table IV.

TABLE IV
POWER ESTIMATION OF PAM AND FPGA

Component	Power (mW)
Core8051s	26
PAM block	0.1
FPGA Bram	0.31

Based on the active component of each operating mode as illustrated in the Figure 2, the power consumption of a mode is the sum of the power of all components of this mode. For the sake of simplicity, we assume the following:

- There is no hardware failure occurring during the execution of application such as On-Duty or Enhance modes.
- Two or more hardware failure can't happen at the same time.

TABLE V
FOUR TYPES OF MONITORING MODE

Component	Monitor0	Monitor1	Monitor2	Monitor3
Processor	On	Failed	On	Failed
Ram memory	On	On	Failed	Failed
FPGA Pro	Off	On	Off	On
FPGA Bram	Off	Off	On	On
Sensor	On	On	On	On
Receiver	On	On	On	On

The FSM model described in the Figure 2 presents the general control that can be applied for all WSN application, in this paper the specific application of hazardous gas detection for area such as harbor or warehouse is used to show the physical implementation of our approach. In this application, the operating mode of node system is initially at Monitoring, if no event occurs during an interval of time, the system enters in Sleep mode to save energy. Otherwise, it will be in On-Duty mode if an alarm is detected. There are four types of Monitoring modes as described in Table V, which include their active components. The power consumption of each mode is given as following:

$$P_{Sleep} = P_{SleepOfPro} = 10mW \quad (2)$$

$$P_{Monitoring0} = P_{Pro}+P_{Ram}+P_{Sen1}+P_{Receiver} = 1123mW \quad (3)$$

$$P_{OnDuty0} = P_{Monitoring0} + P_{Transmitter} = 1253mW \quad (4)$$

$$P_{OnDutySen2Cam0} = P_{Monitoring0}+P_{Sen2}+P_{Cam} = 2425mW \quad (5)$$

As seen in the Equations 4 and 5, the power difference derives from the activation of the second sensor and the camera. These components are enabled by the user if he wants to check the environment or verify the accuracy of sent event from this sensor node. The consumed power when a component is down is also calculated. Based on the previous assumption, these consumptions are calculated as following:

$$P_{ProFail} = P_{Monitoring0} - P_{Pro} = 1087mW \quad (6)$$

$$P_{RamFail} = P_{Monitoring0} - P_{Ram} = 973mW \quad (7)$$

$$P_{SensorsFail} = P_{Monitoring0} - P_{Sen1} = 256mW \quad (8)$$

$$P_{RadioFail} = P_{Monitoring0} - P_{Radio} = 1053mW \quad (9)$$

According to the power results, two groups of operating modes are classified at the Monitoring mode into two criteria: fault-free modes and faulty modes. This separation is depicted in the Figure 6. In the case of faulty modes, our self-diagnosis approaches are applied to detect correctly the failed component.

As related in the previous section, the energy-efficiency is one of the most important design constraints for sensor node because it is a battery-powered. The more number of self-diagnosis like the functional tests is used, the more the energy dissipation is increased. To optimize our approach, we aim to minimize the number of utilization of functional test in the node system by using the online power measurement.

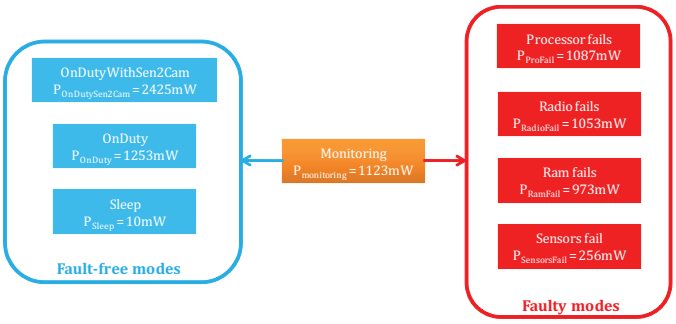

Fig. 6. Fault-free modes and faulty modes of sensor node

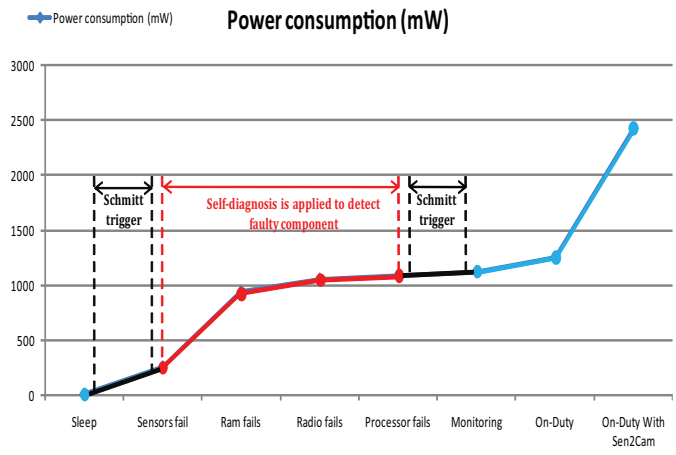

Fig. 7. Power discrepancy between the fault-free and faulty modes

This measurement is realized by implementing the current measurement device in the electrical input of node component. When online power measurement result is obtained, the node system compares it with several power thresholds using Schmitt triggers. The value of these thresholds is selected during the node design, in order to detect where the system mode locates: in fault-free mode or faulty mode. If the system mode is fault-free, no functional test is applied. Otherwise, the functional tests are used to detect the failed component, and then to take an appropriate solution to make our sensor node less vulnerable. The physical implementation of node self-diagnosis is described in detail in Figure 7. For example, two case of on-line self-diagnosis are tested, when in the first case, the self-diagnosis is run for all component one time per four hours, while in the second case, the self-diagnosis is enable when power measurement is located in the zone of faulty modes (between 100W and 1100W as depicted in Figure 7). We assume that a sensor failure occurs in seventh day, the power consumption is 201J (5J, respectively) in the first case (the second case, respectively). Apparently, the energy consumption of second case is 97.5% less than the first case. If a failure component like Ram is detected, our PAM block will enable the FPGA Bram memory to replace it. And then the value of the power consumption of all operating modes is automatically updated with the consumption of Bram memory.

V. Conclusion and Perspective

In this paper, we have introduced a novel self-diagnosis for wireless sensor node in order to identify correctly the failed component, and then our PAM block can take a corrective solution to make our node less vulnerable. And then the physical implementation of our approach is introduced for minimizing the energy consumption of the self-diagnosis. Based on power consumption of each component, our approach could be easily integrated in sensor node. In our future works, the faulty modes will be integrated in the FSM modelling, and the implementation of our self-diagnosis approach combined with PAM block and FPGA in real material will be soon realized.

References

[1] M. Weiser, *The computer science issues in ubiquitous computing*, Communication of ACM, July, 1993, Vol.36, No.7,*pp. 75–84*.

[2] S. Lafortune, C.G. Cassandras, *Introduction to Discrete Event System*, Springer Publishers, *second edition*, Harvard, USA, 2008.

[3] V.T. Hoang, N. Julien and P. Berruet, *Design under Constraints of Availability and Energy for Sensor Node in Wireless Sensor Network*, IEEE Conference on Design and Archtectures for Signal and Image Processing, Karlsruhe, Germany, October 2012.

[4] A. Bagchi, S.L. Hakimi, *An optimal algorithm for distributed system level diagnosis*, Proceedings of FTCS-21, 1991, *pp. 214–221*.

[5] D.M. Blough, H.W. Wang, *The broadcast comparison model for online fault diagnosis in multicomputer systems: theory and implementation*, IEEE Transactions on Computer 48, May, 1999, *pp. 470–493*.

[6] S. Hosseini, J. Kuhl, S. Reddy, *A diagnosis algorithm for distributed computing systems with dynamic failure and repair*, IEEE Transactions on Computer 33, March 1984, *pp. 223–233*.

[7] Nicolas Ferry , Sylvain Ducloyer , Nathalie Julien and Dominique Jutel, *Power/Energy Estimator for DesigningWSN Nodes with Ambient Energy Harvesting Feature*, EURASIP Journal on Embedded Systems, January , 2011.

[8] M. Ringwald, K. Romer, *Deployment of sensor networks: Problems and passive inspection*, Fifth IEEE Workshop on Intelligent Solutions in Embedded Systems, June, 2007, *pp. 179–192*.

[9] K. Romer, M. Ringwald, A. Vitaletti, *Snif: Sensor network inspection framework*, ETH Zurich, Zurich, 2006.

[10] S. Chessa, P. Santi, *Crash faults identification in wireless sensor networks*, Computer Communications International Journal, September 2002, Volume 25, Issue 14,*pp. 1273–1282*.

[11] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentell, *Fault Tolerance Techniques for Wireless Ad Hoc Sensor Networks*, Proceedings of IEEE on Sensor, November 2002, Volume 2, *pp. 1491–1496*.

[12] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentell, *On-line Fault Detection of Sensor Measurements*, Proceedings of IEEE on Sensor, November 2003, Volume 2, *pp. 974–979*.

[13] Myeong-Hyeon Lee and Yoon-Hwa Choi, *Fault detection of wireless sensor networks*, Computer Communications International Journal, September 2008, Volume 31, Issue 14,*pp. 3469–3475*.

[14] Myeong-Hyeon Lee and Yoon-Hwa Choi, *Detection and diagnosis of data inconsistency failures in wireless sensor networks*, Computer Networks International Journal, June 2006, Volume 50, Issue 9,*pp. 1247–1260*.

[15] J. Muppala , G. Ciardo, K.S. Trivedi, *Spnp: Stochastic petri net package*, Proceedings of the Third International IEEE Workshop, 1989, *pp. 142–151*.

[16] http://www.actel.com/documents/IGLOO.