An RTOS based Industrial Wireless Sensor Network using Multiprocessor Support

R. Aswini Bavani
Department of Electronics and Communication
Hindustan University
Chennai-603 103

Aby K Thomas
Department of Electronics and Communication
Hindustan University
Chennai-603 103

ABSTRACT

RTOS (Real Time Operating System) is a Process which is done between hardware and application in order to provide time constraints. Each industry has their own priority among the various tasks according to their process. RTOS is used here to assign the priorities. The values obtained in plant have to be transmitted to the control room. These values are sent in the form of packets. Packets have to be sent at particular time. During the packets transmission some collision may occur. To overcome this collision wireless HART (Highway Addressable Remote Transducer) is used. This project is implemented in two sections as prototype. First one runs with RTOS and LPC2148 as master node and another as normal data acquisition node to which sensors are connected. Second section contains a controller as per the need. Communications between two nodes are accomplished through wireless HART. The basic view of this technique is to reduce the possibility of collision (and thus increase the communication reliability), and to meet the critical requirement of industrial applications. This is essential to increase the communication reliability, and to meet the requirement of timing determinism of industrial applications. To do this, all the nodes must be synchronized precisely. Also, the stack designer must guarantee that the node can finish everything within the time slot. It offers benefits in terms of platform independency, product life cycle, safety and security, system integration complexity, and performance scalability.

General Terms

RTOS, Micro Controller OS, Wireless Hart, Multiprocessor system.

Keywords

RTOS, ARM LPC2148, Wireless HART, IWSN, Master control and Data Acquisition System.

1. INTRODUCTION

The industrial wireless sensor network (IWSN) is an emerging technology in recent years. Superior reliability, high determinism, timeliness, and security are emphasized**.** Since 2007, there are three standards, the Wireless HART, ISA100.11a, and WIA-PA, have been released based on the IEEE 802.15.4. One of the important characteristic of the wireless HART is the TDMA (time division multiple access). In the TDMA mechanism, all communications among nodes are allocated and limited within corresponding timeslots [1]**.** HART is a digital protocol for two-way communication between a host application and smart field instruments, providing access to diagnostics, configuration and process data[3]. This is essential to reduce the possibility of collision (and thus increase the communication reliability), and to meet the critical requirement of timing determinism of industrial applications [4].

At the same time, the rapidly increasing complexity and other specific requirements of industrial systems have made it necessary to adopt the real time operating system (RTOS) in the IWSN [2]. These requirements include safety, security, availability, support for actuators, system integration, network size, product life cycle, etc...[6] Furthermore, the use of multiprocessor (and/or multi-core) architecture has become another trend in the industrial system design because it is an effective way to improve the system scalability and manage the complexity and cost [7]. The multi-processor support of IWSN is, therefore, needed to follow this trend. However, the adoption of RTOS and support of multiprocessor have made it more challenging to guarantee the timing integrity [9]. RTOS (Real Time Operating System) is a Process which is done between hardware and application. Each industry has their own priority among the various tasks according to their process. RTOS is used here to assign the priorities. Application code designed on an RTOS [11] can be quite diverse, ranging from a simple application for a digital stopwatch to a much more complex application for aircraft navigation [10]. Here Micro C/OS II is used to assign the priorities [5]. Good RTOSes, therefore, are scalable in order to meet different sets of requirements for different applications [8].

2. PROPOSED SYSTEM

This project is implemented in two sections as prototype. First one runs with RTOS and LPC2148 as master node and another as normal data acquisition node to which sensors are connected. Second section may contain any controller as per need. Communications between two nodes are accomplished through wireless HART. The basic view of this technique is to reduce the possibility of collision (and thus increase the communication reliability), and to meet the critical requirement of timing determinism of industrial applications. This is essential to reduce the possibility of collision (and thus increase the communication reliability), and to meet the critical requirement of timing determinism of industrial applications. To do this, all the nodes must be synchronized precisely. Also, the stack designer must guarantee that the node can finish everything within the time slot. It offers benefits in terms of platform independency, product life cycle, safety and security, system integration complexity, and performance scalability. An implemented Wireless HART stack has proven the feasibility of the proposed architecture in practical product design.

The main aim of this paper is to monitor the industrial machine condition, intruder detection and to monitor these values at remote location through IWSN. To monitor the machine condition, we are using LM35 sensor which determines the heat generated by the machine, capacitive level sensor to determine the level of the fluid and the MEMS which senses the movement of the machine. By that we can know the condition of machine. We have to convert these analog values into digital in order to monitor industrial parameters. In the existing system bulks of messages are transmitted between nodes so there are chances of data collision in transmission also priorities are not given to the parameters out of several parameters. Without priority the system may leave the most important task. Instead it will perform some low priority task (which has effects less than the effects of failure of high priority task). To overcome this problem, priority based scheduling is given in this paper. Micro Controller Operating System II is used to assign priorities to the task. It uses pre-emptive scheduling method to run the task according to its priority level.

When a low priority task is executing, if there is an arrival of high priority task then the CPU will stop executing the current task and goes to high priority task, executes it and returns to the low priority task. This process is pre-emptive scheduling which is more efficient than the round robin scheduling.

2.1 Block diagram

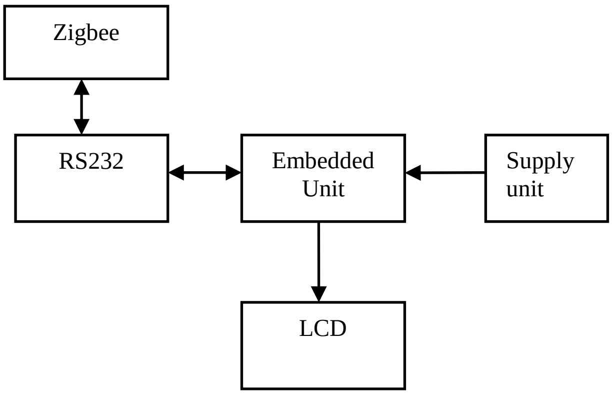


Figure 1 Master node

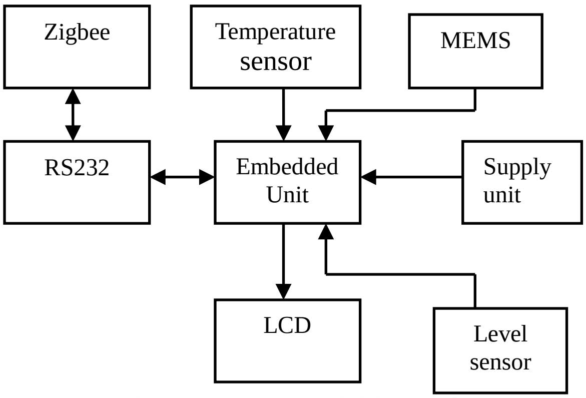


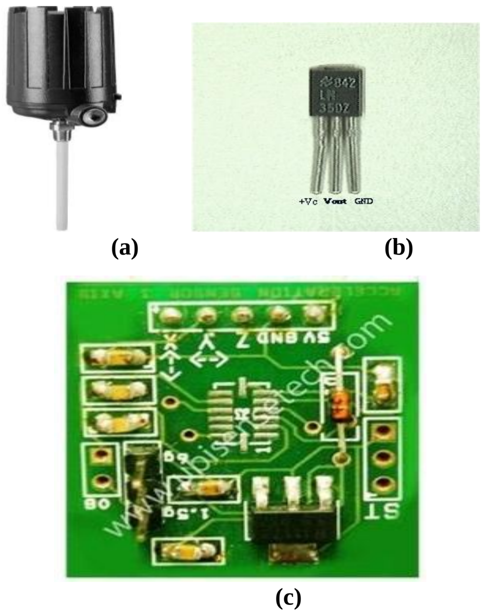
Figure 2 Data Acquisition node

2.2 ARM LPC2148

The LPC2148 micro-controllers are based on a 32/16 bit ARM7TDMI-S CPU core. They have real-time emulation and embedded trace support, which combines the micro-controller with embedded high speed flash memory of 512 kB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode (16bit instruction set) reduces code by more than 30 % with minimal performance penalty.

2.3 Sensor

The sensors used in this paper are temperature sensor, level sensor, MEMS sensor to monitor the industrial parameters like temperature of the boiler, level of the liquid in the tank and to detect the vibration of the machineries in the industrial plant.



(a) **(b)**

(c)

Figure 3 Sensors
a) Level sensor b) Temperature sensor c) MEMS

3. EXPERIMENTAL SETUP

High priority is given to MEMS sensor, second priority is to temperature sensor and low priority is given to level sensor. All the data are acquired in data acquisition node and sent to master node. In master node according to the priority the required task is processed and action is taken.

Table 1 Conditions provided for testing

MEMS	Temperature	Level
Normal	Normal	Normal
Normal	Abnormal	Normal
Abnormal	Abnormal	Normal

For level sensor, the normal value should be between the ranges 60-80, for temperature sensor the normal range is 25-50 and for MEMS it is between 70-100. For every sensor there will be a set point and the ranges mentioned above are allowable values. The values should not exceed the limit, if it exceeds then it enters into abnormal condition.

4. EXPERMENTAL RESULT

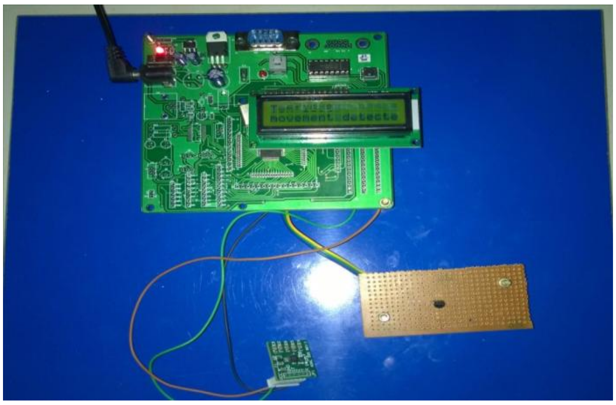
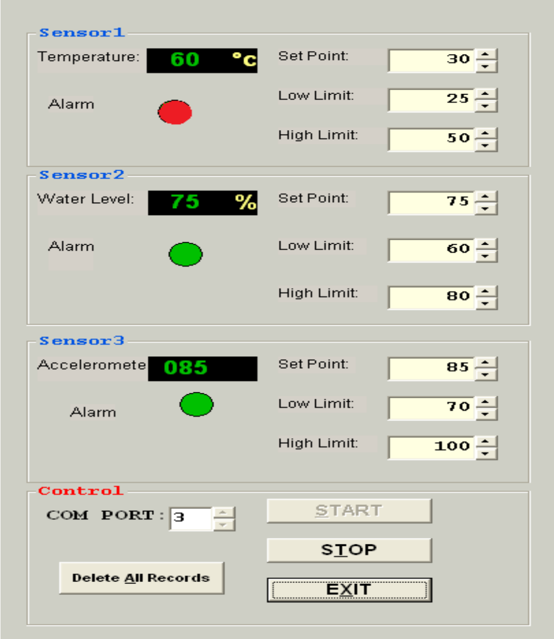


Figure 4 Data Acquisition Result

The three sensors are connected with the processor which acts as a data acquisition node. It is placed in every node to get the values. All the values are displayed in LCD.



Sensor1
Temperature: 60 °c
Alarm 🔴
Set Point: 30
Low Limit: 25
High Limit: 50

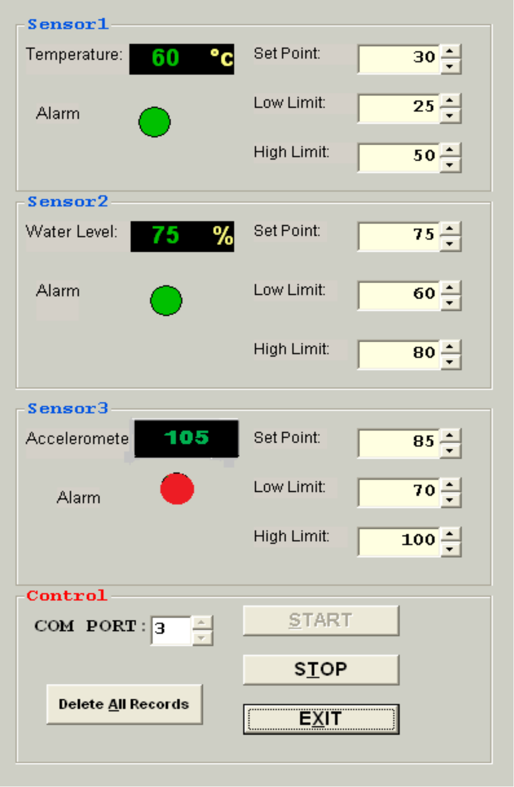
Sensor2
Water Level: 75 %
Alarm 🟢
Set Point: 75
Low Limit: 60
High Limit: 80

Sensor3
Acceleromete 085
Alarm 🟢
Set Point: 85
Low Limit: 70
High Limit: 100

Control
COM PORT : 3

Figure 5 Abnormality in Temperature value

When the temperature value exceeds the limit the particular alarm is indicated as red color in software result. The other two values are normal hence it is shown in green color.



Sensor1
Temperature: 60 °c
Alarm 🟢
Set Point: 30
Low Limit: 25
High Limit: 50

Sensor2
Water Level: 75 %
Alarm 🟢
Set Point: 75
Low Limit: 60
High Limit: 80

Sensor3
Acceleromete 105
Alarm 🔴
Set Point: 85
Low Limit: 70
High Limit: 100

Control
COM PORT : 3

Figure 6 Abnormality in MEMS and Temp

When the MEMS value enters into abnormal condition it is denoted as red color. Though the temperature is abnormal, as per the priority given to the sensors the MEMS will get high priority. After the MEMS attain its normal condition the temperature is controlled.

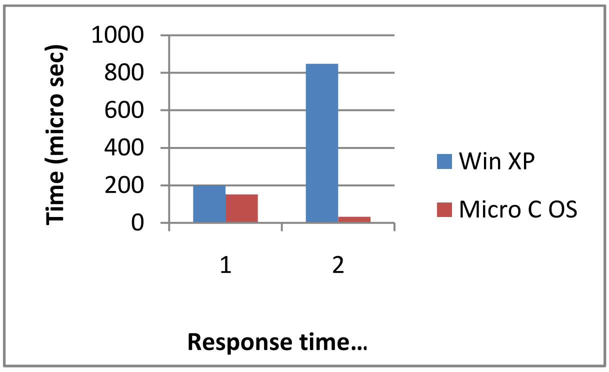


Figure 6 Performance Analysis

The experiment is done using 2 operating system. It shows a satisfied result when Microcontroller OS is used than the Windows XP. The y-axis is time and the x-axis is the response time and latency.

5. CONCLUSION

In this paper the industrial parameters are monitored and processed according to the priority given to it. Data Acquisition node is used to collect the data from various nodes and sends data to the master node through wireless HART protocol which is also to minimize the data collision. Based on the priority given to the various parameters the master node will perform the task. Many nodes can be implemented to monitor and control the operations from the master node. In future, the experiment will be done in different OS like Linux, Neutrino.

6. ACKNOWLEDGEMENT

Our sincere thanks to the staff members of the Hindustan University who have contributed towards the development of the thesis.

7. REFERENCES

[1] Jianping Song; Song Han; Mok, A.K.; Deji Chen; Lucas, M.; Nixon, M.. A RTOS Based Reconfigurable Architecture for IWSN Stack with Arm Cortex and EM250RF Processor Support. International Journal of Science and Research, 2014

[2] Yi Yang, Ping Xiang, Mike Mantor, Huiyang Zhou. CPU-assisted GPGPU on fused CPU-GPU architectures, High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium, 25-29 Feb. 2012

[3] Xiuming Zhu; Song Han; Mok, A.; Deji Chen; Nixon, M.. Hardware challenges and their resolution in advancing WirelessHART. IEEE Int. Con. on Industrial Informatics (INDIN), 2011, 416-421

[4] Faisal A. Al- Nasser,Hosam Rowaihy "Simulation of Dynamic Traffic control system based on Wireless sensor network", IEEE Symposium on Computers & Informatics 2011, pp 40-45.

[5] Liu Zhongyuan, Cui Lili, Ding Hong, published a paper titled "Design of Monitors Based on ARM7 and Micro C/OS-II" 2010 IEEE

[6] Abusayeed Saifullah; YouXu, ChenyangLu, and Yixin Chen, Real time scheduling for wireless HART networks. Real time systems symposium(RTSS),2010,150-159

[7] Utz Roedig, Sarah Rutlidge, James Brown, Andrew Scott, “Towards Multiprocessor Sensor Nodes”. ACM Workshop on Hot Topics in Embedded Networked Sensors (HotEmNets '10), 2010

[8] Vehbi C. Gungor; Gerhard P. Hancke. Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. IEEE Transaction on Industrial Electronics, 2009, 4258-4265

[9] Z. K. Wang and L. Zhao, "Research on task schedule strategy in µC/OS-‡U" , Journal of Shandong University of Technology, vol. 23, no. 2, pp30-35, Mar. 2009

[10] A. Sangiovanni-Vincentelli and M. D. Natale, "Embedded System Design for Automotive Applications", IEEE Computer, vol. 40, no. 10, pp. 42-51, Oct. 2007

[11] Hangci Zhou, Guangwen Wu, “the programming based on embedded real time operating system,”Beijing: Beihang University press, 2006