A RTOS BASED MULTIPROCESSOR SUPPORT MODEL FOR INDUSTRIAL NETWORKS

K.Yasodha
PG Scholar, Embedded System Technologies
Department of Electrical and Electronics Engineering
Sri ram Engineering College, Veppampattu,
Thiruvallur(Dt.)
kyasoleo@gmail.com

J.Singaravelan
Associate Professor
Department of Electrical and Electronics Engineering
Sri ram Engineering College, Veppampattu
Thiruvallur(Dt.)

***Abstract*— The aim of the project is to the design of industrial sensor network (ISN) with the adoption of real time operation system (RTOS) and to provide multi-processor support. 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. In general real time must be able to respond predictably to unpredictable events meet completion deadlines of particular tasks and process multiple tasks at once. The RTOS assigns the task priority based on task deadline instead of the shortest task processing time. Round-robin (RR) is one of the algorithms employed by process and network schedulers in computing. To prove a real-time application is been designed in which the parameters (concentration of gas, level, temperature) of the boiler from a plant is been taken. Here two ARM processor is been used to provide multiprocessor support .The first processor manages the input parameters taken from the plant and the second processor to control the final control elements present in plant side via the actuators . The data's flow via real-time operating system (RTOS) .Real-time operating systems are optimized to provide a low amount of jitter when programmed correctly; a task will take very close to the same amount of time to execute each time it is run. The interprocessor communication is realized by the high speed UART interfaces which are available in both chips.**

***IndexTerms*—multiprocessor,real time operation system (RTOS),Round-robin,industrial, control.**

I. INTRODUCTION

The industrial sensor network (ISN) has gathered more and more notice in recent years. Superior reliability, determinism, timeliness, and security are emphasized. Since 2007, three standards, the WirelessHART, ISA100.11a, and WIA-PA have been released based on the IEEE 802.15.4. One of the strong similarities of them is the TDMA (time division multiple access) -based media access mechanism.

In the TDMA mechanism, all communications among nodes are allocated and limited within corresponding timeslots. For example, Wireless HART uses 10ms fixed timeslots which allows for maximum packet sizes while still allowing time drift to be technical feasible. ISA100.11a needs flexible time slots to allow for duo cast. WIA-PA uses 802.15-4-2006 super-frame with configurable timeslot toothis 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, i.e. the jitter of synchronization should be much smaller than the length of time slot. The stack designer must guarantee that the node can finish everything within the time slot.

Such timing critical requirement has become one of the primary challenges to design the protocol stacks. Firstly, it is challenging to finish the complicated tasks (packet parsing, encryption, decryption, authentication, etc.) within such a short timeslot by the processor with limited resource (clock frequency, memory, energy supply, etc.). Secondly, the ISN stacks are often only a part of the timing critical tasks that thedevice should execute. It is much more difficult to ensure thetiming integrity in a complicated multi-task system.

The design of industrial sensor network (ISN) stacks requires the adoption of real time operation system (RTOS). Challenges exist especially in timing integrity and multi-processor support. As a solution, we propose anRTOS-basedarchitecturefor ISN stacks with multi-processor support. It offers benefits in terms of platform independency, product life cycle, safety and security, system integration complexity, and performance scalability. And future challenges as well as suggestions to standard improvement are discussed. Integrating a number of real-time applications onto a single microprocessor raises issues of resource allocation and partitioning. Disparate applications require access to processor and other resources in a manner that ensures they are able to complete the necessary computations within specified time constraints, whilst ensuring that they do not impinge upon the real-time behavior of other applications.

There is currently considerable interest in hierarchical scheduling as a way of providing temporal isolation between applications executing on a single processor. In a hierarchical system, a global scheduler is used to determine which application should be allocated the processor at any given time and a local scheduler is used to determine which of the chosen application's tasks should actually execute. A number of different scheduling schemes have been proposed for both global and local scheduling. This dynamic priority based scheduling and fixed priority scheduling.

In this paper we focus on the use of fixed priority pre-emptive scheduling (FPPS) for both global and local scheduling. Fixed priority pre-emptive scheduling offers advantages of flexibility over cyclic approaches whilst being sufficiently simple to implement; that it is possible to construct highly efficient embedded real-time operating systems that use this scheduling policy.

II. PROPOSED ARCHITECTURE

In this paper, we design the industrial sensor network (ISN) stacks with the adoption of real time operation system (RTOS) and to provide multi-processor support. The parameters (concentration of gas, level, temperature) of the boiler is been taken and it is given to the controller to make a safety decision before going for a shutdown of a boiler

The decision taken by the first processor is made to control the final control elements present in plant side via the actuators which is again controlled by another processor. The data's flow via real-time operating system (RTOS) .The inter-processor communication is done through UART .

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. RTOS offers benefits in terms of platform independency, product life cycle, safety and security, system integration complexity, and performance scalability.

.

III. FUNDAMENTAL COMPONENTS

In Fig.1 shows the block diagram which contains four sections, they are

- Input section
- Processor section
- Communication section
- Output section

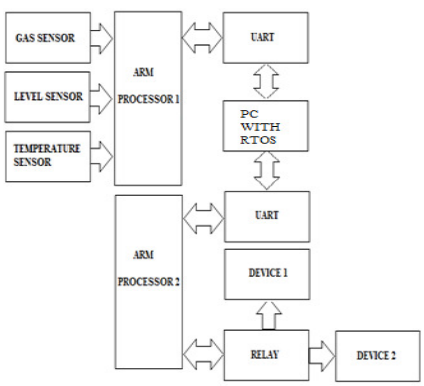

Figure.1 Block diagram of multiprocessor support

IV.IMPLEMENTATION OVERVIEW

A. Input section

In the input section there are three sensors, the sensors are used measure the physical parameters which occurs during process, the values are given to the processor for further processing of the data's .

The sensors are

- Gas sensors
- Level sensors
- Temperature sensors

Mq-6 Semiconductor Gas Sensor ForLpg measures the concentration of gas in its vicinity. Gas sensor interacts with a gas to measure its concentration. Each gas has a unique breakdown voltage i.e. the electric field at which it is ionized. Sensor identifies gases by measuring these voltages. The concentration of the gas can be determined by measuring the current discharge in the device.This is a simple-to-use liquefied petroleum gas sensor, suitable for sensing LPG (composed of mostly propane and butane) concentrations in the air.

The MQ-6 can detect gas concentrations anywhere from 200 to 10000ppm.This sensor has a high sensitivity and fast response time. The sensor's output is an analog resistance. The drive circuit is very simple; all you need to do is power the heater coil with 5V, add a load resistance, and connect the output to an ADC.

The level measurement can be either continuous or point values. Continuous level sensors measure level within a specified range and determine the exact amount of substance in a certain place, while point-level sensors only indicate whether the substance is above or below the sensing point. Generally the latter detect levels that are excessively high or low.

The LM35 series temperature sensors are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling.

B.Processor section

There are two arm processors present in this project. One Arm processor is used in the input section and the other in the output section. The input section arm processor will process the input variables and it is passed to the RTOS for further processing and the second processor is connected to the output section to have control on the final control element i.e the relay section.

A 32-bit ARM7 TDMI-S microcontroller with 40 kB of on-chip static RAMS and 512 kb of on-chip flash memory. It has In-System Programming/In-Application Programming (ISP/IAP) via on-chip boot loader software. Two 10-bit ADCs provide a total of 14 analog inputs Single 10-bit DAC provides variable analog output .A set of two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog are present in the Arm 7.

C.Interprocessor communication

The interprocessor communication is realized by the high speed UART interfaces which are available in both chips. The Fig.2 shows the IPC hardware in which the on-chip interconnection networks. and in the case of multiple cores in multiple chips, it is the off-chip interconnection circuits. In the former case, the IPC interface can be expected to have lower power consumption, shorter latency, and higher throughput. But the latter is suitable for the aforementioned "low high combination" hardware.

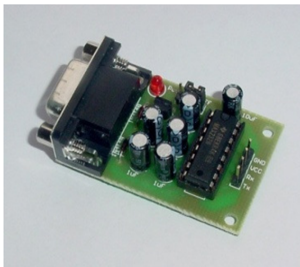

Figure 2.IPC Hardware

D.Real time operating system

The RTOS assigns the task priority based on task deadline instead of the shortest task processing time. To reduce processing overhead and save bandwidth, the RTOS removes the tasks with expired deadlines from the medium. Furthermore, if a real-time task holds the resources for a longer period of time, other tasks need to wait for an undefined period time, causing the occurrence of a deadlock. This deadlock situation degrades the performance of task scheduling schemes in terms of end to end delay. Hence, we would deal with the circular wait and preemptive conditions to prevent deadlock from occurring. We would also validate the simulation result using a real test-bed.

Round-robin (RR) is one of the algorithms employed by process and network schedulers in computing. As the term is generally used, time slices are assigned to each process in equal portions and in circular order, handling all processes without priority (also known as cyclic executive).

Many stacks are allocated for each and every action and they are arranged in layers. In particular, each layer in the stack has at least one dedicated thread and one dedicated mailbox. All the inter layer interactions are through this mail box and is managed by the RTOS. These interactions use two message formats, they are To send or receive a packet: packet payload, packet type, priority, timeout, etc.To query or modify status or configuration of one layer: management command number, target layer, parameter, value, return format, etc. are specified.

There are four main procedures which are used to put a message, get a message, and enter and delete a data from the stack of the processor they are Put(), Get(),Add() and Remove().

To access the messages between the mailboxes four mechanisms have been designed, they are

- Periodical polling: one layer periodically requests the information of another layer.
- Periodical reporting: one layer periodically reports itsInformation to another layer. This is typically used by thelower layers to report its status to top layer (e.g. theapplication layer. Comparing to the periodical polling, itproduces less traffic.
- Event-driven notification: one layer reports its informationto another layer only when a particular event happens.
- Instant request: one layer can request the information fromanother layer whenever it needs, and the target layerresponses when it receives the request message.

E.Output section

The output section consists of relay , which controls two devices here we use a LCD display and a alarm circuit which gives the information about operations of the circuit.The Fig.3 shows the Hardware architecture of the system.

Figure 3. Hardware architecture

VI.SOFTWARE REQUIREMENT

A Keil version 4 is been used to simulate the results and to control the real time parameters also the µVision IDE from Keil combines project management, make facilities, source code editing, program debugging, and complete simulation in one powerful environment. The µVision development platform is easy-to-use and helping you quickly create embedded programs that work. The µVision editor and debugger are integrated in a single application that provides a seamless embedded project development environment.

A. Simulation modules concepts

The simulation and the total system control uses Round-robin with Interrupts architecture concept used.

B.Round-Robin With Interrupts

Round-robin (RR) is one of the algorithms employed by process and network schedulers in computing. As the term is generally used, time slices are assigned to each process in equal portions and in circular order, handling all processes without priority (also known as cyclic executive).Round-robin scheduling is simple, easy to implement, and starvation-free. Round-robin scheduling can also be applied to other scheduling problems, such as data packet scheduling in computer networks.It is an Operating System concept. The name of the algorithm comes from the round-robin principle known from other fields, where each person takes an equal share of something in turn.

The features are

- Interrupt routines deal with the very urgent needs of devices
- Interrupt routines set flags to indicate the interrupt happened
- Main while loop polls the status of the interrupt flags and does any follow-up processing required by a set flag.

There are three modulesAll the three modules will have direct control on timing measurement, ADC values, UART modules, IO port values of the processor. The three modules are

- Input section-ADC module
- Communication section - Serial module
- Output section-LCD module

C.Basic software flow

The Fig.4 shows basic software flow of the system which starts with hardware initialization after that software initialization is done. After that software and hardware events are processed.

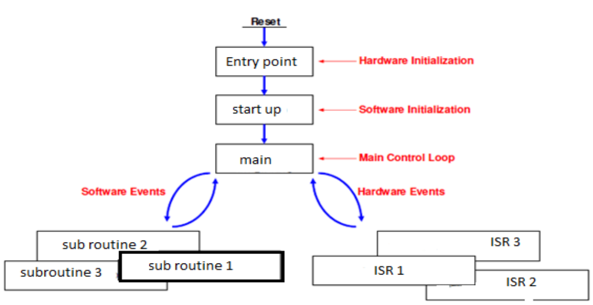

Figure.4 Basic software flow

D.Measured experimental results: Timer values and IO Port details

Timing parameters and the I/O ports are calculated in the code and recorded in static variables. We track these variables by the debugger via the "Live Watch" window. As it captures variables only when the RTOS is in idle state, and thus, this method approximately avoids any visible impact on the timing. It is import to know that, the time spent by the CPU to toggle the output pins and to calculate the timing parameters causes errors to the results, so it should be minimized.

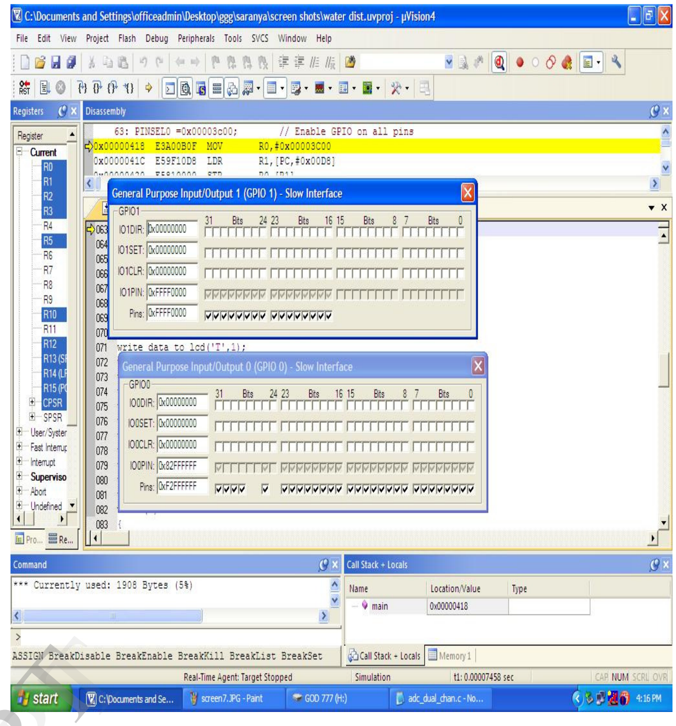

Figure.5 IO port values

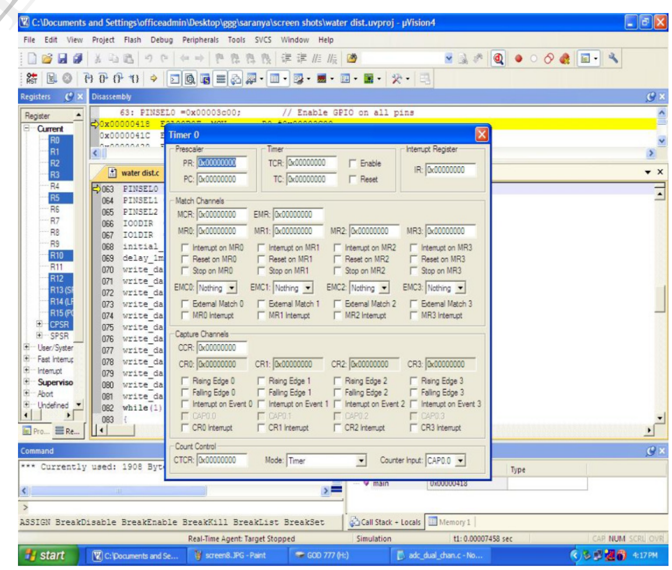

Figure.6 Timer values

The data's seen on the register values are keenly absorbed and from that the timing integrity is acceptable. Thepackets between the manager and field device are properly transmitted within slots. The timing overhead caused by the RTOS is early illegible in the total time. We have observed the impact of the IPC on the timingintegrity. When the data rate of UART is set to higher values or lower values willdamage the timing integrity or packet loss in the serialization layer. In this implementation, we have applied the direct memory access (DMA) for the UART interface which has reduced the CPU load to handle the UARTinterface.

VI. CONCLUSION AND FUTURE WORK

In this paper we addressed the problem of scheduling a number of tasks on a multiprocessor using a set of arm processor with the help of RTOS servers. The motivation for this work comes from the power and thermal industries where the advent of high performancemicroprocessors is now making it both possible and cost effective to implement multiple applications on asingle platform.

Our research has focused on systems that are scheduled using fixed priority pre-emptive scheduling at both local and global scheduling levels.

Future work is required to provide an effective algorithm capable of choosing an optimalor close to optimal set of environmental parameters of the given systems.

Another interesting area of future research involves incorporating Quality of Service (QoS) requirementsinto hierarchical fixed priority pre-emptive systems.

The RTOS and IPC of the proposed architecture may consume extra energy so the power consumption must be improved. the extra traffic caused by the inter layer interaction should be evaluated and optimized in the future

VIII. REFERENCES

[1] T-W. Kuo, C-H.Li. “A Fixed Priority Driven OpenEnvironment for Real-Time Applications”.In proceedings of theIEEE Real-Time Systems Symposium. Madrid, Spain, December1998.

[2] Z. Deng, J.W-S.Liu.“Scheduling Real-Time Applications inan Open Environment”.In proceedings of the IEEE Real-TimeSystems Symposium. December 1997.

[3] UtzRoedig, Sarah Rutlidge, James Brown, Andrew Scott, “TowardsMultiprocessor Sensor Nodes”. ACM Workshop on Hot Topics inEmbedded Networked Sensors (HotEmNets '10), 2010

[4] S. Chachra and M. Marefat, “Distributed algorithms for sleep schedulingin wireless sensor networks,” in Proc. 2006 IEEE International Conf.Robot.Autom., pp. 3101–3107.

[5] P. Guo, T. Jiang, Q. Zhang, and K. Zhang, “Sleep scheduling for criticalevent monitoring in wireless sensor networks,” IEEE Trans. ParallelDistrib. Syst., vol. 23, no. 2, pp. 345–352, Feb. 2012.

[6] F. Liu, C. Tsui, and Y. J. Zhang, “Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks,” IEEE Trans. wireless Commun., vol. 9, no. 7, pp. 2258–2267, July 2010.

[7] J. Liu, N. Gu, and S. He, “An energy-aware coverage based nodescheduling scheme for wireless sensor networks,” in Proc. 2008 International Conf. Young Comput. Scientists, pp. 462–468.

[8] O. Khader, A. Willig, and A. Wolisz, “Distributed wakeup schedulingscheme for supporting periodic traffic in wsns,” in Proc. 2009 EuropeanWireless Conf., pp. 287–292.

[9] B. Nazir and H. Hasbullah, “Dynamic sleep scheduling for minimizingdelay in wireless sensor network,” in Proc. 2011 Saudi InternationalElectron., Communications Photon. Conf., pp. 1–5.

[10] D. Shuman and M. Liu, “Optimal sleep scheduling for a wireless sensornetwork node,” in Proc. 2006 Asilomar Conf. Signals, Syst. Comput., pp. 1337–1341.

[11] S. Paul, S. Nandi, and I. Singh, “A dynamic balanced-energy sleepscheduling scheme in heterogeneous wireless sensor network,” in Proc.2008 IEEE International Conf. Netw., pp. 1–6, 2008.

[12] K. Mizanian, R. Hajisheykhi, M. Baharloo, and A. H. Jahangir, “RACE:real-time scheduling policy and communication architecture for large scale wireless sensor networks,” in Proc. 2009 Commun. Netw. Services Research Conf., pp. 458–460.

[13] M. Yu, S. J. Xiahou, and X. Y. Li, “A survey of studying on taskscheduling mechanism for TinyOS,” in Proc. 2008 International Conf.WirelessCommun., Netw. Mobile Comput., pp. 1–4.