RTOS Based Priority Dynamic Scheduling for Power Applications through DMA Peripherals

Srikanth.K[#1], Narayanaraju Samunuri[*2]
[#]M.Tech & VLSI-ES & Department of ECE & JNTU-Hyderabad
H-No. 9-30/4, S.V.Nagar, Nagaram, ECIL, Hyderabad, A.P, India

Abstract --Embedded Real Time application uses multi threading, a key concept of any conventional OS. The advantage of multi-threading include greater throughput, more efficient use of CPU so that it cannot remain idle for long time, better system reliability, improved performance on multiprocessor computer. The use of Real Time Operating Systems (RTOSs) became an attractive solution to simplify the design of safety-critical real-time embedded systems.

In this modern world, the execution of power applications with their priority at runtime is more important for industrial applications. In this paper, the main focus is on execution of different power applications using priority dynamic scheduling during runtime. According to the User, at runtime he can assign the priorities for power appliances and change the priority according to the user need, each power application named as Task. The embedded software applications require performing a task which is efficiently achieved by dividing the global task into many smaller subtasks. Previously the application of embedded system solutions to power applications are based on unique algorithm which in take maximum amount of time for CPU utilization. Most of such power applications simply go for DSP processors, which in turn takes a simple hardware design based on digital signal controllers or 16-bit microcontrollers. For such simple power applications, the rich peripherals for communications, data logging and compact microcontrollers are used. For an application that does not use or refer RTOS, may go for complex software architecture based on timers that are available inside the microcontroller.

The concept of dynamic assignment of priorities to interrupts is used which reduces the time delay for a lower priority task under some circumstances becomes a higher priority task. We are using ARM with combination of RTOS (ucos-ii) to minimize the complexity of system. In this paper we are doing the scheduling of the tasks or power appliances through Direct Memory Access (DMA) peripherals to optimization of performance of the CPU utilization.

Keywords – ARM controller, RTOS, Task scheduling, Priority based Dynamic scheduling, Direct Memory Access (DMA).

I. INTRODUCTION

In general, an operating system (OS) is responsible for managing the hardware resources of a PC and hosting applications which are called as tasks. An RTOS performs these tasks, but is also specially designed to run applications with very precise timing and a high degree of reliability. The OS meant for RTS is referred as RTOS where the time at which results are produced is of major concern. Basically, real-time systems are classified in to two types Hard and Soft real-time systems [1]. In general terms, real-time systems have to provide not only logically correct results but also in how much time they are producing it. The necessity to adopt Real-Time Operating Systems (RTOS) is increased as the real time systems are getting complex day by day in order to simplify the design. In RTOS the Scheduling is an important aspect for executing a various task at runtime. At runtime depends on their priority the tasks should be executed whenever the user required. For this we use a dynamic priority scheduling for to executive a different applications or tasks dynamically. Dynamic priority scheduling [2] is a type of scheduling algorithm in which the priorities are calculated during the execution of the system. The aim of dynamic priority scheduling [2] is to adapt to dynamically changing progress and form an optimal configuration in self-sustained manner. The idea of dynamic priority scheduling is to confine focus on algorithms that assign priorities based on temporal parameters and optimization of recourses; this utilization measurement of a dynamic scheduling, called schedulable utilization of resources, is scaled from 0 to 1, and the higher percentage of schedulable utilization means the better algorithm.

In present world, managing the various industrials power applications is the most important aspect. But at the runtime the power applications are executed according to their system priority. This paper deals with dynamic priority scheduling with various power applications that should be run dynamically at runtime or execution time. The various power applications are energy meter, LCD, Relays bank, EEPROM etc. For various power applications we are going to assign a various tasks names as task0, task1, task2….etc. In this paper the power applications are executed according to their priority sequence which is provided by user at runtime. The suggested approach in this paper is to use the ucos-ii with the combination of power task applications to perform the priority based dynamic scheduling during run time.

II. IMPLEMENTATION

This paper consists of both the approaches i.e. Hardware approach and Software approach

A. *Hardware Approach*

In this paper, Fig 1 shows the Priority dynamic scheduler for power Applications through DMA Peripherals in which all the power applications are assigned or names as an individual task. System dynamically assigns priorities for the power application tasks. Without using Dynamic priority scheduling for given applications, the tasks at the run time will dynamically assigns their priorities without the intervention of user. Our suggested approach consists of priority dynamic scheduling which change their priorities according to the user requirements. In this we can assign two tasks with static priorities and the remaining tasks with dynamic priorities i.e. these dynamic priorities can be changed by the user according to his requirement.

ARM9 is an ARM architecture 32-bit RISC [3] CPU family. The Friendly ARM Mini2440 is a single board computer based on a Samsung S3C2440 [4] ARM9 microprocessor. The dimensions of board are 10 cm x 10 cm. Mini2440 is a practical low-cost ARM9 development board. It is for S3C2440 [4] processor and the use of professional power stable core CPU chip to chip and reset security permit system stability. The mini2440 [4] Immersion Gold PCB using the 4-layer board design process, such as long wiring to ensure that the factory has been a strict quality control on the key signal lines of signal integrity, the production of SMT machine and mass production;

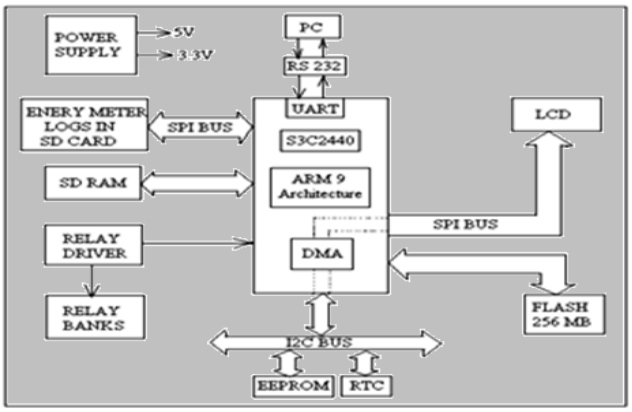

Fig. 1. Block diagram of Priority dynamic scheduler for power Applications through DMA Peripherals

i. Energy Meter Logs in SD CARD '

Energy meter is one of the applications of this project which shows the customer details (e.g. CUSTOMER NAME, CUSTOMER ID, ENERGY (KWH) units, COST/UNIT which sends the date to SD card for storing purposes. The ARM board consists of SD card slots which receives the value from energy meter via SPI bus. The customer details are found on LCD via SPI bus.

ii. Relay Bank

Here we use four relays in hardware which combinedly formed relay bank which shows the status of relay switches.

iii. *EEPROM*

In this paper EEPROM is the temporary storage memory to store the task programs by using I2C Bus. From which we can send the given data or information which is stored temporarily on to the LCD via DMA, which helps the CPU to look out the other task present in our paper.

iv. DMA

The S3C2440A supports four-channel DMA controller [3][6] located between the system bus and the peripheral bus. Each Channel of DMA controller [3][6] can perform data movements between devices in the system bus and/or peripheral bus with no restrictions. In other words, each channel can handle the following four cases:

1. Both source and destination are in the system bus
2. The source is in the system bus while the destination is in the peripheral bus
3. The source is in the peripheral bus while the destination is in the system bus
4. Both source and destination are in the peripheral bus

The main advantage of the DMA [6] is that it can transfer the data without CPU intervention.

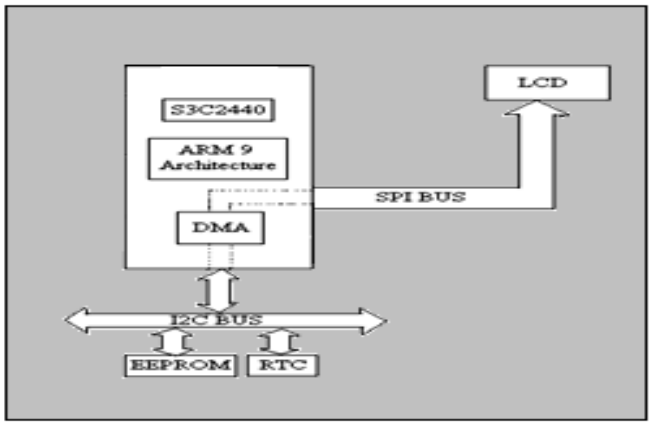

Fig. 2. Execution of applications through DMA (Direct Memory Access) without CPU intervention.

Utilization of DMA, the tasks, which is the data, date or other numeric information which is produced by the RTC , which in turn temporarily stored in the EEPROM which is fed to the DMA via I2C bus. The tasks (date) is displayed on the LCD which is coming through the DMA via SPI Bus. In

this time the CPU look after the other two tasks present in the paper which in turn optimizes the CPU time.

Fig.3. Hardware Setup

B. Software Approach

The software approach consists of ucos-ii (Platform) [7] with the combination of power task applications. The main operation of this approach is to deal with dynamic priority scheduling with various power applications that should be run dynamically at runtime. The various power applications are energy meter, LCD, Relays bank, EEPROM etc. For various power applications we are going to assign a various tasks names as task0, task1, task2….etc. The power applications are executed according to their priority sequence which is provided by user at runtime.

To implement this approach we are using Code warrior ADS v1.2 compiler [8] to generate the *.o file.* Now this generated *.o file* is to be converted into *.bin file.* And then this *.bin file* has to be dumped on ARM9 Kit (S3C2440) to perform the applications.

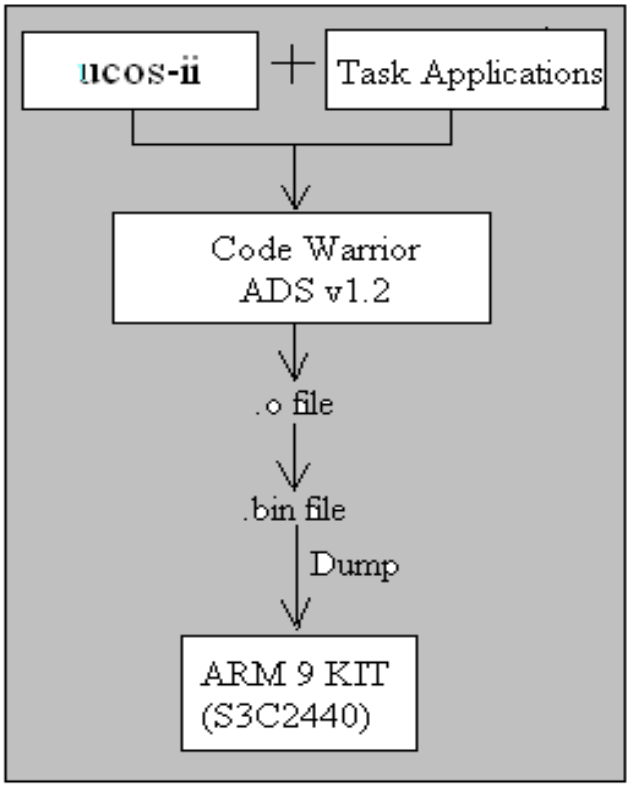

Fig.4. Project process flow (Software approach)

And the complete process flow of the project consists of the combination of software approach, the dumped code in the ARM 9 kit is to be produced on the HyperTerminal with the help of RS 232.

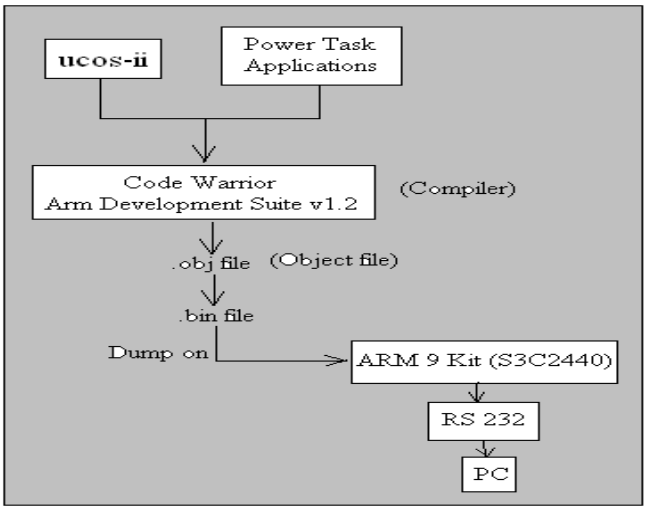

Fig.5. Complete Project Process Flow

III. COMPILATION RESULTS

As mentioned earlier, the application i.e. Power Task applications has to be developed with the combination of RTOS (ucos-ii). The application tests the operation of the proposed approach and produces the required results on the HyperTerminal.

Figure 6 shows how the application task are getting successfully executed dynamically (system dynamically) without priorities. And Figure 7 show that tasks are getting executed dynamically with priorities as send by the user according to his requirements during run time as proposed in our approach.

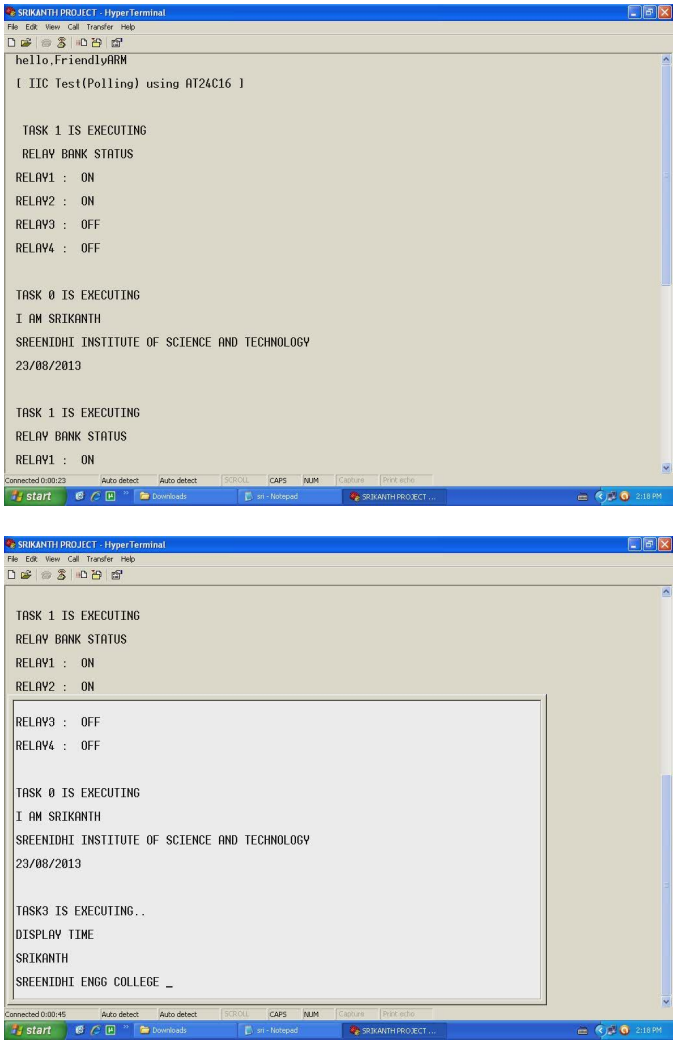

Fig. 6 Execution of task applications without priorities. (Dynamically scheduled)

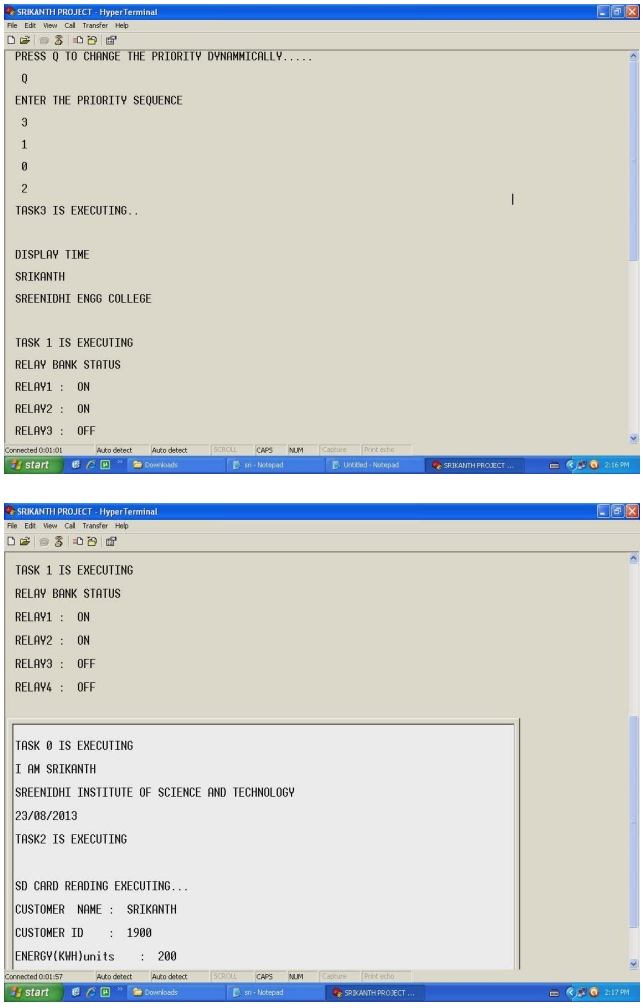

Fig. 7. Execution of task applications with dynamic priorities (dynamically scheduled)

4. CONCLUSION

The proposed system RTOS Based Priority Dynamic Scheduler for Power Applications through DMA Peripherals The algorithm, business logic and dynamic priority scheduling was tested in the hyper terminal. The work includes to use DMA peripherals of advanced microcontrollers and to accelerate the performance of CPU, so that optimization of CPU can be done.

Future works consists of provision of security through GSM module.

As, whenever some one will try to change or changed the priorities of the power task applications from this GSM module we will get message on our mobile that someone has changed the task priorities at run time. We will get the information of the new i.e. changed task priority with the old priorities.

REFERENCES

[1] S.M. Kuo, B.H. Lee, and W. Tian, "Real-Time Digital Signal Processing: Implementations and Applications", Wiley, 2006. ISBN 0-470-01495-4. Section 1.3.4: *Real-Time Constraints.*

[2] Błażewicz, Jacek; Ecker, K.H.; Pesch, E.; Schmidt, G.; Weglarz, J. (2001). *Scheduling computer and manufacturing processes* (2 ed.). Berlin [u.a.]: Springer. ISBN 3-540-41931-4.

[3] "ARM Processor Instruction Set Architecture". Arm.com.Archived from the original on 15 April 2009. Retrieved 18 April 2009.

[4] "Application Processor Product Catalogue | Samsung Semiconductor Global Website". Samsung.com. Retrieved 2012-10-02.

[5] "Samsung Develops Industry's First DDR4 DRAM, Using 30nm Class Technology". Samsung. 2011-01-04. Retrieved 2011-03-13.

[6] DMA fundamentals on various PC platform , from A. F. Harvey and Data Acquisition Division Staff NATIONAL INSTRUMENTS

[7] Micrium Expands RTOS Family with µC/OS-II" (Press release). Micriμm, Inc. 2009-03-24. Retrieved 2010-02-14.

[8] Ruhlmann, William. The Warrior's Code at AllMusic. Retrieved June 201, 2011.

AUTHOR'S BIOGRAPHY

Srikanth.k, pursuing M.Tech in VLSI & Embedded Systems (VLSI-ES), Department of ECE, Sreenidhi Institute of Science and Technology.

Narayanaraju Samunuri, M.Tech in Embedded Systems, with an experience of 11 years in Core Embedded System domain, published International paper on Flex ray Communication