Development and Implementation of PRTOS on Robotic Navigation System

A. Chandra Suresh, K.V.N.M. Prasad, R. Upender Rao

Abstract*—A real-time operating system (RTOS) is a software which ensures that time critical events are processed as efficiently as possible. In this paper, an attempt has been taken to implement a real time operating system, named Preemptive Real Time Operating System (PRTOS), in which all of the important tasks regarding to a real time application have been considered. In this PRTOS, strictly preemptive scheduling algorithm has been used. This Scheduling policy makes sure that important tasks are handled first and the less important later. In this paper we are using free RTOS . Free RTOS" is a small but powerful real-time kernel and it is used in many commercial applications. In this paper our intention is to design and construct a self navigating robot which works on Preemptive priority algorithm. This Free RTOS supports Tasking, Semaphores, Timer Management, Message Queues, Single Step Debugging support, and with Task synchronization. In this paper Robot is operated with Global Positioning system which allows the robot to aware of its position on the earth. Here PRTOS concept is implemented to control the different operational tasks .Hence This robots found some Real time applications like war fields ,mines and complex locations where the humans unable to reach.

***Index Terms*— Pre-emitive prioroity algorithm,Task synchronisation,Real Time operating system,Scheduling policy,Self navigation,Global position System**

I. INTRODUCTION

In this project a Global Positioning System (GPS), Zigbee wireless communication system and an ultrasonic sensor are connected to the embedded system to implement PRTOS.The operational features are presented, which are needed to be considered for practical applications The navigation Robot can travel mostly on flat hard terrain. Robot will able to esstablish its own location on earth and use information from the global positioning system to navigate to a user defined point. The navigation robot adjusts itself against any obstacles, and corrects its path towards the destination. If it couldn't cross the obstacle it return to home and sends information to the user. The dynamic location changing can be done using RF link. The tasks will be executed on priority basis.

Manuscript Received on June2014.

A. Chandra Suresh, Department of Electronics and Communications Engineering, Sri Vasavi Institute of Engineering and Technology, Nandamuru, Pedana, Andra Pradesh ,India.

Dr. K.V.N.M. Prasad, APJENCO Vidhyut Southa Telecomm Engineering, Hyderabad, India.

R. Upender Rao, Department of Electronics and Communications Engineering, NRI Engineering College, Vijayawada, India.

High priority tasks will be given highest priority among all tasks and then low priority tasks will be executed. User can cancel all the tasks at any time, the navigation robot will return to home aborting all pending tasks. In this project, we present a concurrent and real-time task management method for distributed control of modules in a robot. The intention of this project is to design and construct a robot competent navigating to a sequence of instructional tasks. This effect is to be accomplished using a global positioning system (GPS) to allow the robot to become aware of its position on the earth and the positions of wave points relative to its current location. Here Real Time Operating System (RTOS) concept is implemented to control the different operational tasks in System. It is a robust and capable of traveling outdoors on mostly flat ground and able to carry the load of what components are necessary for it to accomplish its function

II. DESIGN APPROACH

In this project we are designing a robot using GPS navigation system and free RTOS for real-time task management. The Arduino open-source electronics prototyping platform based on flexible, easy-to-use hardware and software, which is built around ATmega328 microcontroller. A free RTOS port called DuinOS for Arduino, .The application program written on host pc with free RTOS and later dumped in to the Arduino board that is placed on Robot .Now the Robot is moving according to our application programme. This project works on Pre-emptive priority based algorithm.

A. Preemptive Scheduling

The preemptive Schedule means the highest priority task always occupies the cpu and executes first.

1Task T_i with priority i is running.
2. If (task T_i has finished){
a) Select the next highest priority task is ready to run.
}
Else{
if (a task T_j (j >i) is in the ready list){
a) Select the task T_j which i ready to run.
}
Else{
Go to step 1.
}
}

This Scheduling algorithm is implemented and observe its effects on Robotic Navigation System. This application having five tasks with different priorities.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

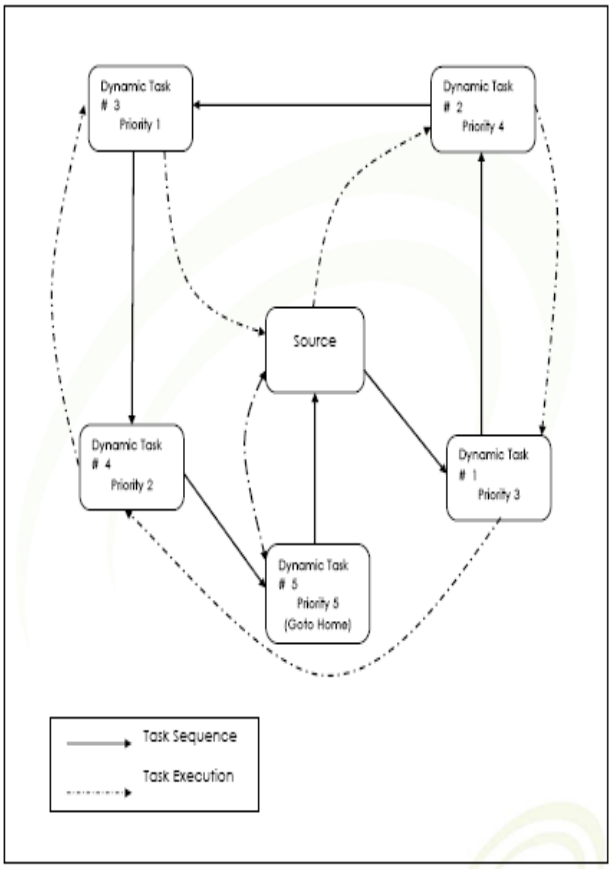
B. Working Example

In this project we are giving a set of maximum 5 tasks and also giving priority for each task during run time. Depending on the priority of each task robot will execute. All tasks can be cancelled at any time by the user.Here Task means Giving the location with priority. So the Robot moving towards the location (latitude, longitude) and continuously sends the current position of the robot to the receiver so the receiver displays that information on the display. If the Task(location) is given by the user the robot can check its priority and execute ,if the current running task having less priority than a new one then lowest priority task suspended temporarily and highest one will be executed very first after completion of highest priority task robot will execute where it was previously interrupted. If the user gives the location with the priority it will move towards respected location with the help of GPS.

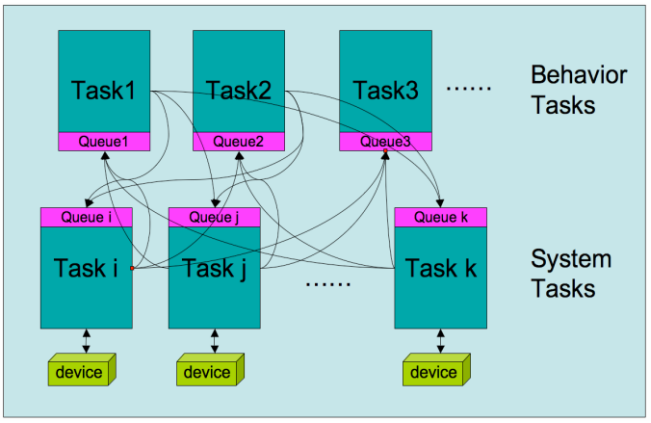
C. Dynamic Priority Change by User

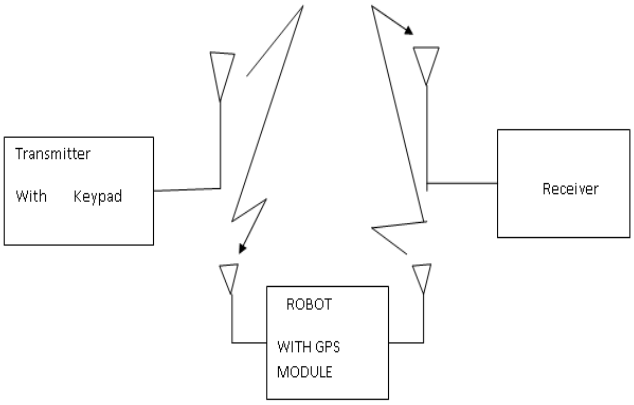
If the user gives the location with the priority it will move towards respected location with the help of GPS. While moving robot continuously sends the location details where it is located by GPS to the receiver and that information is displayed on the display as latitude and longitude. While moving towards defined point if the navigation robot faces any obstacles then it adjusts itself and corrects its path towards the destination. If it couldn't cross the obstacle it return to home and sends information to the user. The dynamic location changing can be done using RF link.

D.Task Synchronisation



Block Diagram





This Bloch Diagram consist of

- Transmitter with Keypad
- Robot with GPS module
- Receiver with Display

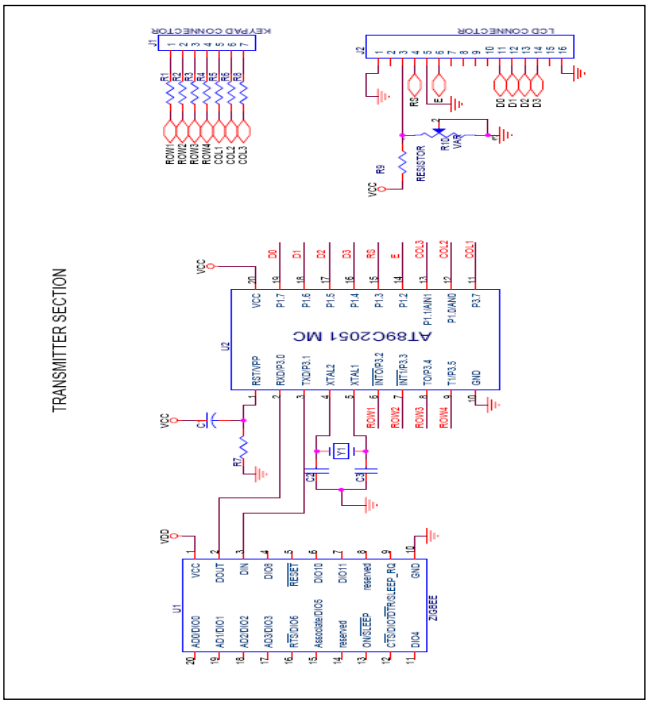
II. FUNCTIONALITY OF TRANSMITTER

The transmitter section having two sections one is for generating task (location) and giving its priority and second section is sends this information to the Robot through Zigbee.The user can give the location through keypad it takes two inputs only one is location second is its priority.The Key pad normal keypad it consist 0 to 9 numbers and constructed with help of 8051 series.Here keypad is constructed with AT89c2051.It supports for two external interrupts.When user gives the location (task) i.e one priority the corresponding interrupt will generate through INTX this is send to ROBOT though ZIGBEE.This Zigbee is interfaced to Microcontroller.Here the ZIGBEE acts RF link between Robot and Transmitter In transmitter the keypad and communication is done with help 8 bit micro controller. In this paper we are using. 89c2051 because it has lot advantages like

- 2K Bytes of Reprogrammable Flash Memory
  – Endurance: 10,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

• Low-power Idle and Power-down Modes
• Green (Pb/Halide-free) Packaging Option.



A. Transmitter Schematic Diagram

Description: The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read-only memory (PEROM). The device is manufactured using Atmel's high-density non-volatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset. The second section of the Transmitter is Zigbee. If the user gives the Location with priority with keypad that generates related interrupt and sends this information to the robot with RF link .Here the RF Link is Zigbee.This Zigbee acts as communication interface between Robot and Transmitter. Helpful Hints

IV. WORKING OF ROBOT

The Robot section having

- Arduino Board
- Microcontroller
- H Bridge Motor
- Gps Module
- Sensors
- Zigbee

Microcontroller ATmega328p functionality: The ATmega328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328P achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed. The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications. So the Microcontroller is available on protype Baord is called Arduino platform board that provides better environment for writing application code and debugging the code..

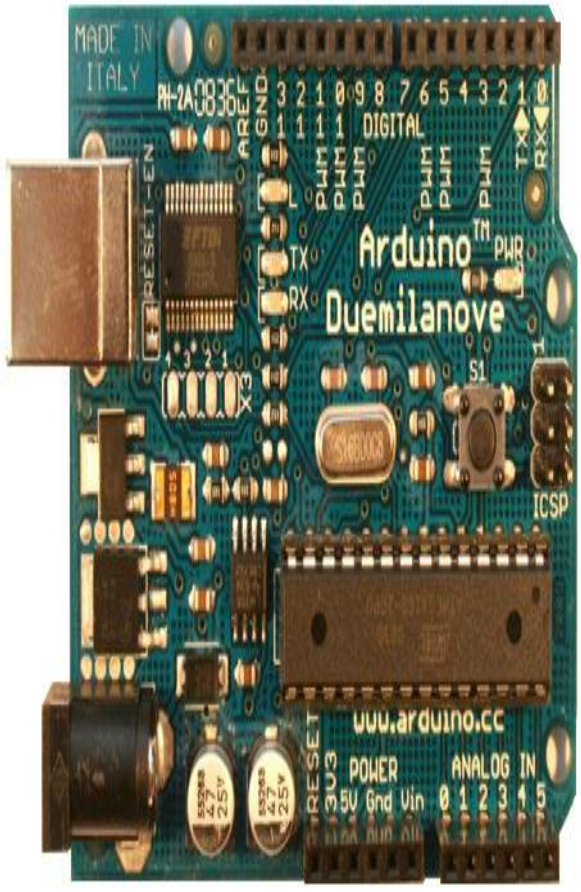
A. Arduino Deemilanove Platform

Now write the application code on Host PC that install with Free RTOS and corresponding Arduino Environment. The application Programme is developed on host machine and generates executable code later that dumed in to the Microcontroller that is available on arduino board. The advantage of using Arduino Board is all sensors and related interfaces are available on single chip.Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The Arduino Duemilanove ("2009") is a microcontroller board based on ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. "Duemilanove" means 2009 in Italian and is named after the year of its release. The Duemilanove is the latest in a series of USB Arduino boards; for a comparison with

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

previous versions, see the index of Arduino boards

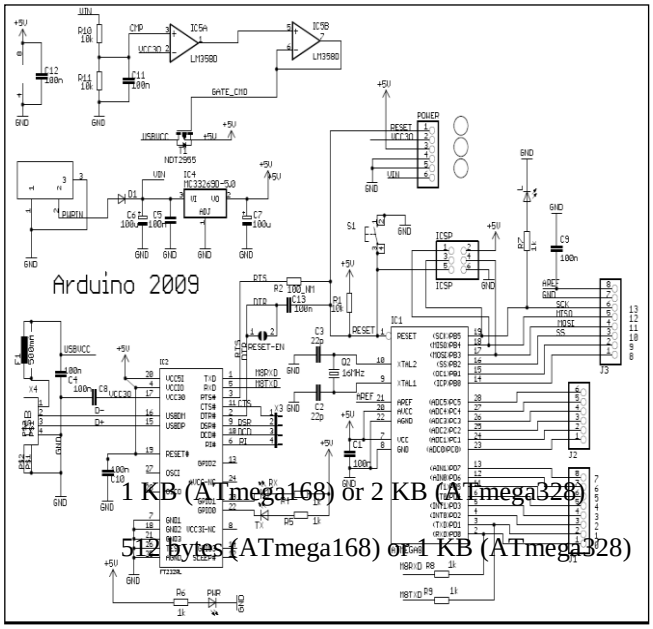
Microcontroller	:	ATmega328p
Operating Voltage	:	5V
Input Voltage :(recommended)	:	7-12V
Input Voltage (limits)	:	6-20V
Digital I/O Pins	:	14
Analog Input Pins:		6
DC Current per I/O Pin:		40mA
DC Current for 3.3V Pin:		50mA
Flash Memory:		32kB
SRAM:		2kB
EEPROM :		1kB



Arduino Platform Baord

The Arduino Duemilanove board features an Atmel ATmega328 microcontroller operating at 5V with 2Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer.

Arduino Hardware Schematics



1 KB (ATmega168) or 2 KB (ATmega328)

512 bytes (ATmega168) or 1 KB (ATmega328)

B. Software

The Arduino programming language is a simplified version of C If user knows C, programming the Arduino will be familiar. If user do not know C, no need to worry as only a few commands are needed to perform necessary functions. An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Here for our application we use Free RTOS.

C. Free RTOS

FreeRTOS is a portable, open source, royalty free, mini Real Time Kernel - a free to download and free to deploy RTOS that can be used in commercial applications without any requirement to expose your proprietary source code

Advantages of Free RTOS:

- Provides one solution for many different architectures and development tools.
- Is known to be reliable. Confidence is assured by the activities undertaken by the SafeRTOS sister project.
- Is undergoing continuous active development.
- Has a minimal ROM, RAM and processing overhead. Typically a kernel binary image will be in the region of 4K to 9K bytes.
- Is very simple - the core of the kernel is contained in only 3 C files. The majority of the many files included in the .zip file download relate only to the numerous demonstration applications.
- Is truly free for use in commercial applications (see license conditions for details).
- Comes with a porting, platform development, or application development service should it be required.
- Is well established with a large and ever growing user base.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

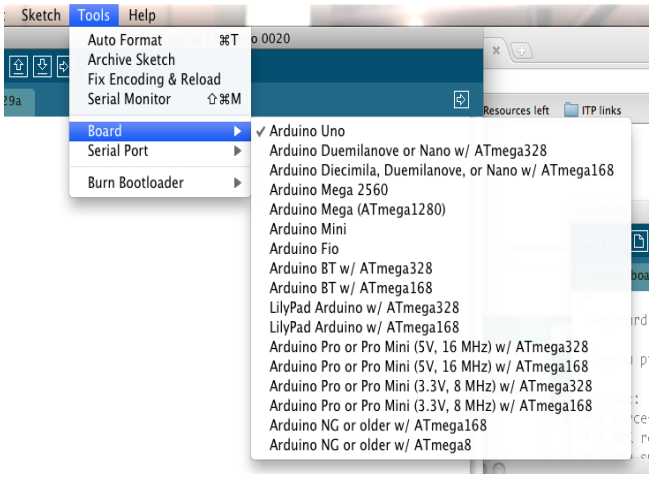
- Contains a pre-configured example for each port. No need to figure out how to setup a project - just download and compile!
- Has an excellent and active free support forum.
- Has the assurance that commercial support is available should it be required.
- Provides ample documentation.
- Is very scalable, simple and easy to use.

D. Project Implementation and Requirements

1. Arduino Duemilanove board
2. USB programming cable (A to B)
3. 9V battery or external power supply (for stand-alone operation)
4. Host PC running the Arduino development environment.

Write the code according to our application that is location finding with priority basis and the robot sends that current location data to receiver module every instant . For that purpose we need to install one cross compiler in Host pc. Now write the code according application and compile on host machine it generates Executable for our application. Now transfer that executable file from host machine to arduino board where the microcontroller ATmege328p is available .For this code transfer we need a USB cable.

Now select arduino baord command from Host machine and transfer th code into the arduino board.This board is placed on Robot .Now Robot is moving according to application programme.



E. Functional Description of GPS Module

In this project LEA-4x modules are used as GPS module. This GPS module is very helpful find the location the Robot. The LEA-4x modules are fully interchangeable, providing the flexibility to use the module with the optimal features for application specific requirements. The LEA-4x modules are designed for use with passive and active antennas. An antenna supervisor is provided on all modules except the LEA-4M. If activated, the GPS receiver is capable of detecting short circuits to the active antenna by checking the bias voltage level and can shut down the voltage bias immediately.

F. Sensor Network

The Robot having two sensors are used to detect the obstacles in the predefined path.If any obstacle comes across its path then the robot takes right deviation the path or left deviation from path .Even though unable to find path it returns to home position and sends this information to receiver .

H Bridge Network

H Bridge network is used to move the Robot forward and back i.It has some advantages over th DC motors. This H bridge ic motors occupies less space.

V. FUNCTIONALITY OF RECEIVER

The receiver receives the current location of the robot and displays that information on display unit as a function of latitude and longitude. The Robot sends the present location through ZIGBEE module. So the receiver receives that information through ZIGBEE that is placed in receiver section. Here the ZIGBEE acts as RF link between Robot and Receiver section. After receiving the information send by the Robot the receiver process that data with help of Microcontroller for the purpose of display. Now the Displays the information like latitude and longitude.

VI. CONCLUSION

In this project we implemented the preemptive priority scheduling algorithm on navigation Robot which executes a set of tasks on priority basis that is high priority task will execute very first and reaming tasks consequently. In this algorithm which task having highest priority that executed first. For this implementation of Scheduling we design a Navigation robot and that fulfills the Scheduling algorithm...

VII. ACKNOWLEDGMENT

The Summarizing the quite broad and active field of Embedded Systems with RTOS research has been a tough challenge. While I remain the only responsible of imprecision's or omissions, there are many people whom I indebted. First with the aim of being as comprehensive and unbiased as possible. Out of the many I invited, I warmly thank for contributing Adi Narayana mallea sr. Project manager in Whirpool india. Abdul Arif Assistant Professor in Sri Vasavi Institute of Engineering and Technology Andhra Pradesh their contributions have been edited incorporated in the paper. And they had given valuable suggestions.

REFERENCES

[1] Dedicated Systems Experts, "What makes a good RTOS. Brussels", Dedicated Systems Experts, Belgium, 2001
[2] Brian Santo, "Embedded Battle Royale," IEEE Spectrum, pp.36-41, December, 2001.
[3] VxWorks Programmer's Guide, 5.1, Wind River Systems, 1993.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

[4] L. M. Thompson, "Using pSOS+ for embedded real-time computing," in COMPCON, pp. 282-288, 1990.
[5] D. Hildebrand, "An architectural overview of QNX," in Proc. Usenix Workshop on Micro-Kernels and Other Kernel Architectures, Apr. 1992.
[6] Sriram V Iyer, Pankaj Gupta, "Embedded Realtime Systems Programming", Tata McGraw-Hill, 2004.
[7] Andrew S. Tanenbaum, Albert S. Woodhull, "Operting Systems Design and Implementation", Second Edition, Prentice-Hall, 2001.
[8] Jean J. Labrosse, ìC/OS: The Real-Time Kernel, R&D Publications, Lawrence, 1993.
[9] S.-J. Oh, et al., "Deterministic Task Scheduling for Embedded Real-Time Operating Systems," IEICE Trans. Inf. & Syst., Vol. E87-D, No. 2, pp. 123-126 , Feb. 2004.

Mr. A. Chandra Suresh, received his B.Tech Degree in Electronics and Communication Engineering from JNTU, Ananthapur and M.Tech degree in Embedded System Design from JNTU, Kakinada, A.P-India. He is currently working as an Associate Professor in the Department of ECE in Sri Vasavi institute Engineering &Technology-Nandamuru Pedana, A.P-India. He had 10 years of teaching experience. His Research interests on Embedded system, Real Time Operating Systems

Dr. K.V.N.M. Prasad, received his B.Tech Degree in Electronics and Communication Engineering from Bapatla Engineering College, Bapatla and M.Tech degree in Electronics and Communication Engineering from Osmania University, Hyderabad, And he awarded Ph.d in ECE from JNTU Hyderabad. Now he is working as Telecom Engineer in APJJENCO Vidhyuth Southa Hyderabad, A.P, India. He had 20 years of teaching experience and 10 years in Research .His interest fields are Embedded system Design, GPS and Digital circuits..

Mr. R. Upendar Rao, received B.Tech degree in Electronics and Communication Engineering and M.Tech degree in VLSISYSTEM DESIGN from JNTU, Hyderabad .He is currently working as an Assistant Professor, in Department of Electronics &communication, NRI Institute of Technology-Vijayawada, A.P, India. Hyderabad 5 years of Teaching Experience. His Research interests on Digital circuits & Embedded Systems.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.