www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 10 Oct 2015, Page No. 14885-14889

MULTITASKING REAL-TIME SCHEDULING IN A ROBOT

B.BHAVYA[1], B.CHAKRADHAR[2], N.BHARADWAJ[3]
[1]M.Tech, Dept of ECE, CMR College of Engineering & Technology, Hyderabad, AP-India.
[2]Assistant Prof, Dept of ECE, CMR College of Engineering & Technology, Hyderabad, AP-India.
[3]Assistant Prof, Dept of ECE, CMR College of Engineering & Technology, Hyderabad, AP-India.

Abstract

This paper deals with loading artificial functions to a robot based on scheduling. Here, scheduling is the one which is used to avoid the delay between appplications. Based on RTOS scheduling has been done. Using zigbee communication the indications are given to the monitoring section. The semantic time scheduling is done to run all applications at a time without any time delay. The paper involves two sections 1. Robot module and 2. Monitoring section. The robot section deals with the data receiving from sensor nodes without any delay. The robotic section runs with RTOS and LPC2929 and acts as master node to which sensors are connected. In the monitoring section the data will be received from the sensor nodes .

Keywords: RTOS, ZIGBEE, ARMLPC2929 ,sensor modules.

I.INTRODUCTION

Semantic-aware real-time which is a framework specifically designed for multi task scheduling in robotics applications. The purpose of a real-time operating system (RTOS) is to schedule tasks in order to guarantee that inputs are acquired and outputs are produced according to timing constraints. In robotic applications, tasks periodically receive information about the environment through sensors or user interfaces, whereas commands to actuators and other outputs are sent at periodic intervals.

SEMANTIC AWARENESS is an ideal system for issues related to timing integrity, the extra traffic caused by the inter layer interaction in robotics.

In the existing prototype, we have noticed that, bulks of messages are transmitted between nodes so there are chances of message collision in transmission. In the proposed system we avoid this problem by optimizing the architecture and enhancing the system resources by implementing Real Time Operating System which manages the shared resources in real time environments, Besides the RTOS this system also provides power efficiency, this Real Time Operating System is developed on micromiun ucos-2.

The basic system requirements for porting this OS is that the device must have sufficient flash memory, on chip interrupts and timers. This OS file structure is mainly divided into three sections, they are:

1) Processor specific files, where all the library files of the devices are available.

2) Application specific code, where all the header files of the user application code are present,

3) Processor independent code where all the OS concepts and their functions are available, such as OS_MUTEX.c, OS_SEM.c, etc,.

The job of the RTOS is to manage the allocation of these resources to users in an orderly and controlled manner. The basic view of this technique is to reduce the damages to the human and gives the information about mine in the border section. If the light intensity is reduced means based on the sensor output the lighting system will be in on condition. Any sound will come due to mine explored it will be detected by the sensor and through zigbee communication it will send information to military section.

This sensor node is composed of a micro-processors, transceivers, displays and analog to digital converters. Sensor nodes are deployed for military process monitoring and control.

This paper is based on RTOS task scheduling and the resource allocation for the real time changing events, here the resources for the external events are a sound sensor, metal detector, mems accelerometer and light dependent resistor. Values of these resources change in real time and these values are recorded at the monitoring section through zigbee communication technique without much time delay between the sensors.

II RELATED WORK

Semantic awareness is based on RTOS task scheduling and resource allocation for the real time changing events, here the resources for the external events processed are a sound sensor, metal detector, mems accelerometer, and light dependent resistor. Values of these resources change in real time. Initially all of these task functionality are present in task dormant state, but TASK functionalities are not visible to the processor, to get these functions into TASK ready state we have to create a task by using this function OS_TASK_CREATE(), and OS_TASK_CREATE_EXT(), and the function is given below.

```
INTU OSTaskCreate (void (*task) (void *pd)),
                   Void *pdata,
                   OS_STK *ptos,
                   INTU8U prio)
```

Where we have the task name its data, stack memory size and priority of each task. The hardware includes ARM lpc2929 , ZIGBEE , sensor.

A. ARM-LPC2929

The LPC2929 combine an ARM968E-S CPU core with two integrated TCM blocks operating at frequencies of up to 125 MHz, Full-speed USB 2.0 OTG and device controller, CAN and LIN, 56 kB SRAM, up to 768 kB flash memory, external memory interface, three 10-bit ADCs, and multiple serial and parallel interfaces in a single chip targeted at consumer, industrial and communication markets. To optimize system power consumption, the LPC2926/2927/2929 has a very flexible Clock Generation Unit (CGU) that provides dynamic clock gating and scaling.

The ARM968E-S is a general purpose 32-bit RISC processor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of micro programmed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective controller core.

Amongst the most compelling features of the ARM968E-S are:

- Separate directly connected instruction and data Tightly Coupled Memory (TCM) interfaces
- Write buffers for the AHB and TCM buses

Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. The ARM968E-S is based on the ARMv5TE five-stage pipeline architecture. Typically, in a three-stage pipeline architecture, while one instruction is being executed its successor is being decoded and a third instruction is being fetched from memory. In the five-stage pipeline additional stages are added for memory access and write-back cycles. The ARM968E-S processor also employs a unique architectural strategy known as THUMB, which makes it ideally suited to high-volume applications with memory restrictions or to applications where code density is an issue.

The key idea behind THUMB is that of a super-reduced instruction set. Essentially, theARM968E-S processor has two instruction sets:

- Standard 32-bit ARMv5TE set
- 16-bit THUMB set

B. LDR Sensor:

The system needs to know what the light level is in a particular room so when automating internal lighting it needs to know if the lights should be activated or not. Otherwise it defeats the purpose of energy saving by Automating the lights for cost savings.This LDR wires directly into a M1 Zone Input (Any Zone). The Zone need to be programmed as a Analog Zone. The more light the LDR sensor has on it the lower the voltage the zone will read and the lower the light level, the higher the zone voltage.

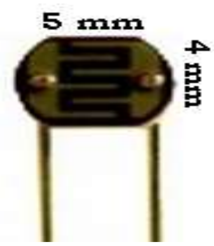

Fig 2. LDR Sensor

*C. Mems Sensor***:**

MEMS accelerometers are one of the simplest but also most applicable micro-electromechanical systems. They became indispensable in automobile industry, computer and audio-video technology. This paper presents MEMS technology as a highly developing industry. An accelerometer is an electromechanical device that measures acceleration forces. These forces may be static, like the constant force of gravity pulling at our feet, or they could be dynamic - caused by moving or vibrating the accelerometer.

Fig 3. Mems Sensor

D. Sound sensor:

Sound sensor uses the principle of microphone. The Microphone, also called a "mic", is a sound transducer that can be called as a "sound sensor". This is because it produces an electrical analogue output signal which is proportional to the "acoustic" sound wave acting upon its flexible diaphragm. This signal is an "electrical image" representing the characteristics of the acoustic waveform. Generally, the output signal from a microphone is an analogue signal either in the form of a voltage or current which is proportional to the actual sound wave.

Fig 4. Sound Sensor

***E. Metal Sensor*:**

A metal detector is a portable electronic instrument which detects the presence of metal nearby. Metal detectors are useful for finding metal inclusions hidden within objects, or metal objects buried under ground. The simplest form of a metal detector consists of an oscillator producing an alternating current that passes through a coil producing an alternating magnetic field. If a piece of electrically conductive metal is close to the coil, eddy currents will be induced in the metal, and this produces a magnetic field of its own. If another coil is used to measure the magnetic field (acting as a magnetometer), the change in the magnetic field due to the metallic object can be detected.

Fig 5.Metal Sensor

III SYSTEM IMPLEMENTATION & RESULTS

The block diagram shows that the different sensors are connected to the ARM CORE. The job of RTOS is to schedule theses tasks and the resource allocation should be done for the real time changing events. Here, the resources for the external events are sound sensor, metal detector, mems accelerometer and light dependent resistor. These values will change accordingly and recorded in the monitoring section.

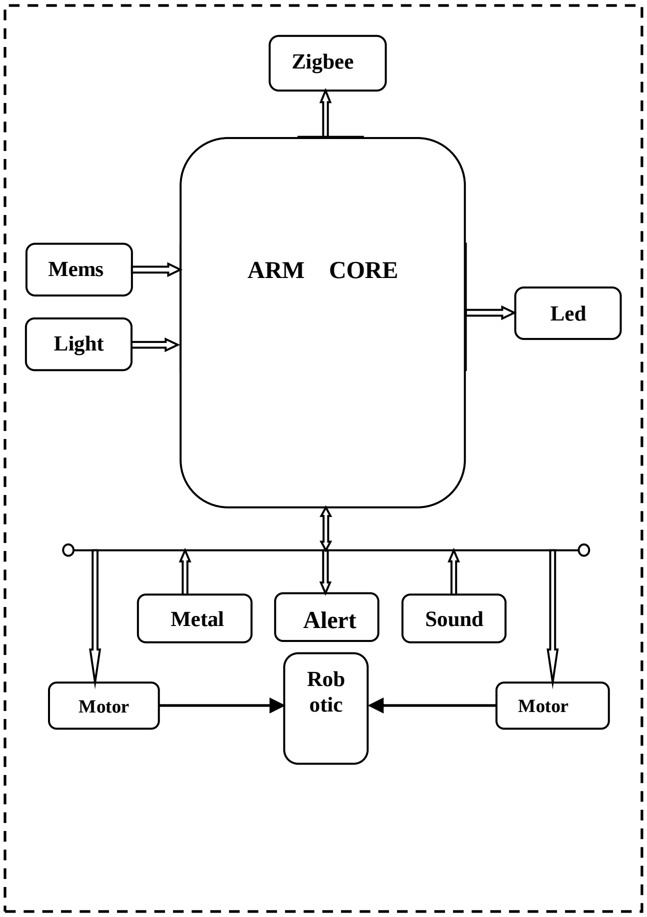

Fig 6. Block Diagram

Here the sound detector which is connected to GPIO pins , when sound is detected the respective GPIO pins goes low and the status of the pin becomes low, this is read through IOPIN register. And the condition if any sound is to send an information serially through UART1 which is configured with 9600 BR with 8 bit transfer, one stop bit, and no parity. This task is created through OSTASKCREATEEXT() function by specifying the task name, its relative data, stack memory, and PRIORITY etc., Metal detector which detects any metal explosives comprise of magnet, if any metal is detected a DC voltage is generated, this sensor is connected to GPIO pins of ARM, if any metal is detected the condition is to forward an information, relating to the metal.

The light dependent resistor where the resistance of it changes with the intensity of the light, detects the day/night light intensity, this sensor is connected to ADC0 of ARM to its respective channel and through AD0CR register and for respective value PWM output will be changed. The mems accelerometer which detects any vibrations (movements in un predictive direction which is caused due to bomb explosion) the incoming sensor values are in analog values, Through on chip ADC0 we convert these signals into digital through AD0CR register by configuring its channel, frequency, resolution and start condition. If the mems value exceeds the threshold value an information will be transferred to main loop.

Here each and every task goes into TASK READY state whenever the its TASK create function is called, the OS_SCHED() function automatically schedules these tasks and sends the higest priority task into TASK READY state, inorder to do this we have to call OS_START() OS_INIT() FUNCTIONS, where the functionality of the respective tasks starts executing according to there priority. Here each and every task must have at least anyone kind of delay and every TASK must be written in infinite.

This paper is based on RTOS task scheduling and the resource allocation for the real time changing events, here the resources for the external events are a sound sensor, metal detector, mems accelerometer and light dependent resistor. Values of these resources change in real time and these values are recorded at the monitoring section through zigbee communication technique without much time delay between the sensors.

Finally, Fig 7 shows the multiple tasks working in the real time. First the processor will be initialized and then it will read the values of ADC, GPIO, UART and then after reading, if the sensors values are greater than the threshold value then the operation of that particular sensor will be in active state accordingly. And the direction of the motor is controlled accordingly.

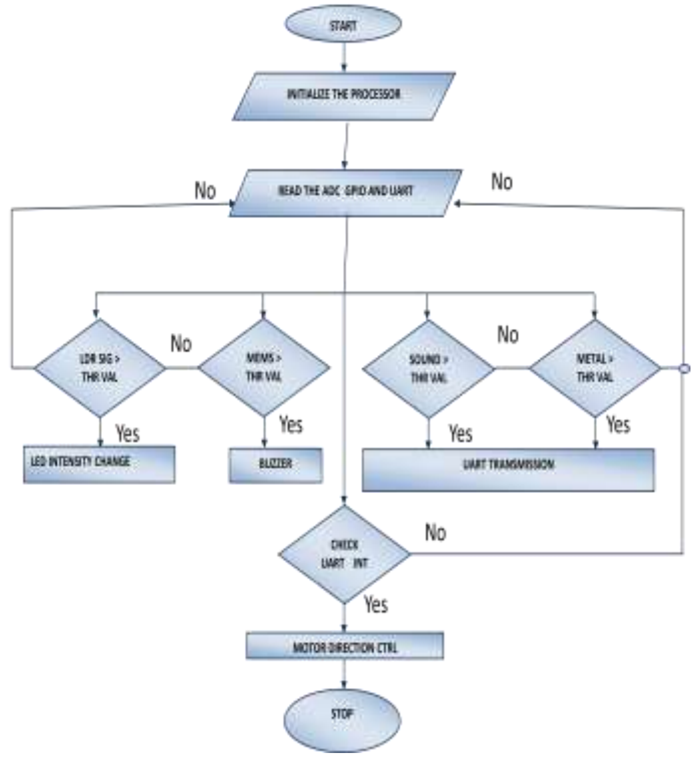

Fig 7. Flow Chart

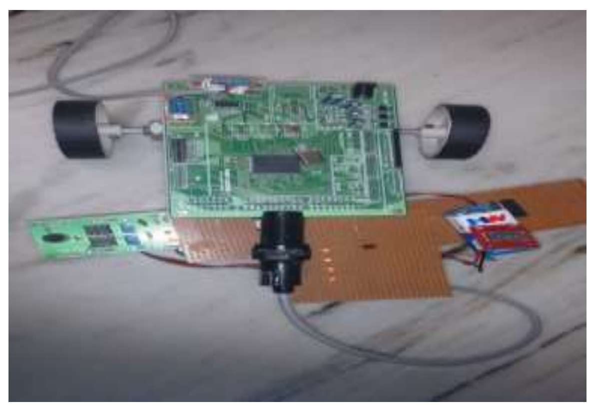

Fig 8.System Hardware

Fig 8. shows a robotic vehicle which is designed for military purpose in order to detect the metal. These robotic vehicle is connected to four different types of sensors in which these sensors are scheduled accordingly and then the data is received from the sensor nodes without any time delay. Here RTOS plays an important role in managing and allocation of the resources in an orderly and controlled manner.

DOI: 10.18535/ijecs/v4i10.45

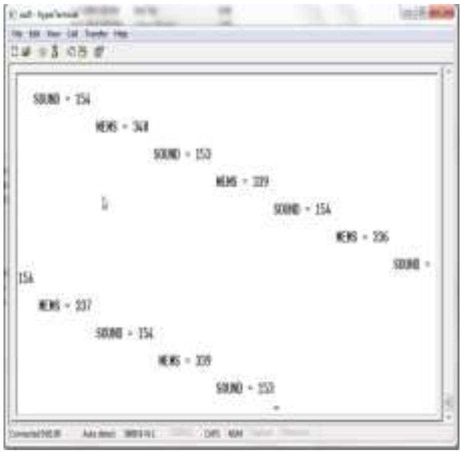

Fig 9. When mems and sound sensor were detected.

The above figure indicates that the data has been received from the sound and mems sensor without any much time delay.

IV CONCLUSION & FUTURESCOPE

In this paper we are making a self decision make and monitor robot application Here, scheduling is used to avoid the delay between one application with another. Based on RTOS, scheduling has been done and a framework designed to deal with time period programming and scheduling of task sets depending on the the present context and on the "semantic content of tasks."

In this paper we implemented mine detected robot using sensor networks. In future we can add camera and gps module. The camera will capture if any person is detected it will capture the image and it will display in military section. Gps used for where exactly the mine was detected it will display the latitude and longitude values.

V REFERENCES

[1] R. Brennan, M. Fletcher, and D. Norrie, "An agent-based approach to reconfiguration of real-time distributed control systems," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 444–451, Aug. 2002.

[2] I. A. D. Nesnas, A. Wright, M. Bajracharya, R. Simmons, and T. Estlin, "CLARAty and challenges of developing interoperable robotic software," in *Proc. 2003 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2003, pp. 2428–2435.

[3] A. Brooks, T. Kaupp, A. Makarenko, S. Williams, and A. Oreback, "Towards component-based robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2005, pp. 163–168.

[4] Y. hsin Kuo and B. MacDonald, "A distributed real-time software framework for robotic applications," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 1964–1969.

[5] L. B. Becker and C. E. Pereira, "SIMOO-RT—An object-oriented framework for the development of real-time industrial automation systems,"

[6] A.K.Ray, M.Gupta, and et al. Sonar Based Autonomous AutomaticGuided Vehicl (AGV) Navigation Proc. Of IEEE Int. Conf. onSystem of Systems Engineering pp.1-8, 2008

[7] B.Hofmann-Wellenhof and et al. *Navigation – principles of positioning and guidance*, Springer, 2003.