International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 3, Issue 4, pp: (168-174), Month: October - December 2015, Available at: www.researchpublish.com

Design and Implementation of an Embedded In-Vehicle System With Multitask Management Using RTOS

Rajani kumari Pasupuleti[1], Ratna babu Yellamati[2]

[12] Affiliated by JNTUK, Department of ECE, VLITS College, Guntur, Andhra Pradesh, INDIA

Abstract: **Several survey reports on vehicle accidents says that, mostly accidents are happening due to Drunk and driving, Driver's drowsiness and over speed. Unnecessary sudden parking of vehicles on highways at night time is another factor .The above mistakes are corrected accidents can be reduced to great extent. A new system has proposed to reduce accidents and in case if accident happened system will inform the owner and ambulance about this accident by using GPS and GSM services. The designed system involves in two sequences of actions 1.monitoring 2.control. In monitoring phase it sense and read all the parameter values .In controlling phase system responds automatically if any abnormalities in the above readings and controls the vehicle . System software is designed using "RTX kernel" and "MDK Keil UV5" to be worked on the target board of LPC2148. This proposed system can provide better solution to prevent accidents and makes vehicles as highly secured.**

Keywords: **RTX Kernel, LPC2148, MDK UV5, monitoring, controlling, GPS Receiver, GSM Services.**

I.INTRODUCTION

The purpose of a real-time operating system(RTOS) is to schedule tasks in order to respond according to the variations in surroundings and produce outputs with in timing constraints. In robotic applications, tasks periodically receive information from surroundings through sensors or user interfaces, whereas commands to actuators and other outputs are sent at periodic intervals. In the design of software module for this project we consider the idea of using RTX, the Keil small footprint RTOS, on an ARM processor-based microcontroller. If we are used to writing procedural-based C code on microcontrollers, we may doubt the need for such an operating system. The use of an RTOS represents a more sophisticated design approach, inherently fostering structured code development, which is enforced by the RTOS application Programming Interface (API). The RTOS structure allows you to take an object-orientated design approach while still programming in C. The RTOS also provides you with multithreaded support on a small microcontroller. These two features create a shift in design philosophy, moving us away from thinking about procedural C code and flowcharts. Instead, we consider the fundamental program tasks and the flow of data between them. The use of an RTOS also has several additional benefits, which may not be immediately obvious. Since an RTOS-based project is composed of well-defined tasks, using an RTOS helps to improve project management, code reuse, and software testing. The trade-off for this is that an RTOS has additional memory requirements and increased interrupt latency. Typically, RTX requires between 500 Bytes and 5KB of RAM and 5KB of code, but remember that some of the RTOS code would be replicated in your program anyway. We now have a generation of small, low-cost microcontrollers that have enough on-chip memory and processing power to support the use of an RTOS. Developed using this approach is therefore much more accessible.

The job of the RTX kernel is creating tasks and allocation of the resources to the tasks in an orderly and controlled manner. For illustration used in An embedded In-Vehicle system , these kinds of systems are used to prevent road accidents . This kind of systems could be implemented with simple programming and without any scheduling and inbuilt OS, but those systems are failed in the case of delay parameters. Due to lack of coordination between the tasks there would be delay in the operation of the tasks and damage would be huge. For example let us consider detection of distance between the front end object to the vehicle and controls vehicle speed , when vehicle is started it is moving on the road ,all the created tasks are performing one bye one ,if another task is running mean while vehicle reaches the object in such case if the control is not switched to the task distance detection and controlling vehicle collides with the object then accident might occur . In the existed paper delay constraints are the drawback .So in this paper tasks are implemented with real time scheduling by using RT Kernel as OS..By using effective real-time scheduling we can schedule the tasks according to their priorities. In this system we implemented 5tasks comprises detection and corresponding control operations .

The tasks are

- Alcohol detection in both cases on the start of vehicle and while running the vehicle.
- Steering wheel grip detection and control of vehicle engine.
- Engine temperature detection and vehicle engine control.
- Distance between front end vehicle and the corresponding vehicle and controlling speed of the vehicle.
- Accident detection and inform that to the nearby ambulance services and the owner or to the well -Wishers (parents or friends or any known person).

These 5 tasks are scheduled by using pre-emptive scheduling .so if any task need recourses in emergency then the control is transferred to that task and allocate resources after completion resources are available back to the suspended task .

Block Diagram:

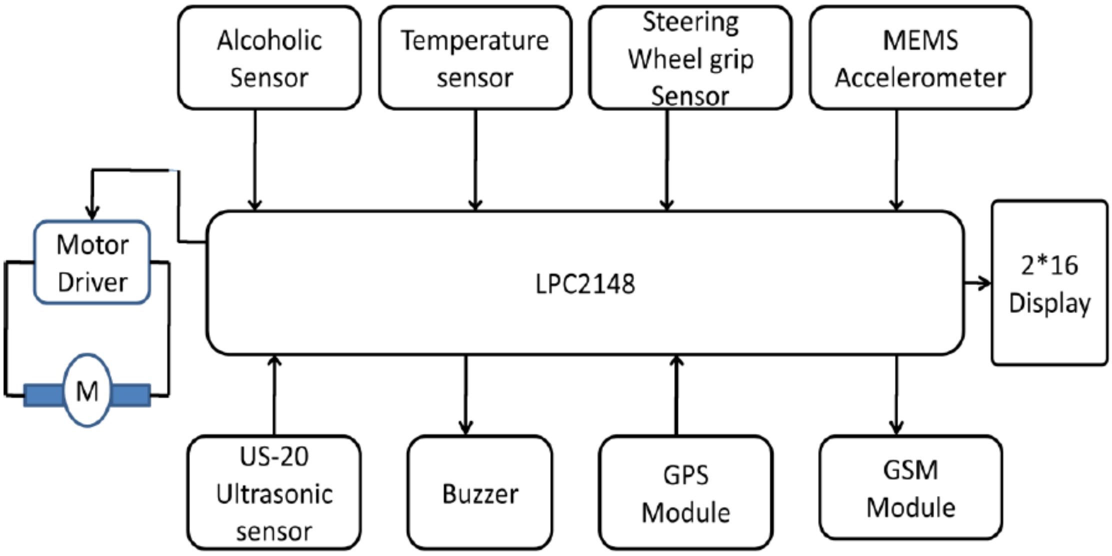

Fig1. Block diagram

II. HARDWARE MODULES

RTOS task scheduling and resource allocation are for the real time changing events; here the resources for the external events processed are a Alcohol Detector, steering wheel grip detector, MEMS accelerometer, engine temperature detector and ultrasonic sensor. Values of these resources change in real time. LPC2148 processor is the target system, code is ported into the target system using serial port.

LPC2148 Processor:

LPC2148µc architecture. The ARM7TDMI-S is a general purpose 32-bit microprocessor, 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 processor core. Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory. The ARM7TDMI-S processor also employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue. The key idea behind Thumb is that of a super-reduced instruction set. Essentially, the ARM7 TDMI-S processor has two instruction sets:

- The standard 32-bit ARM set.
- A 16-bit Thumb set.

The Thumb set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code. Thumb code is able to provide up to 65% of the code size of ARM, and 160% of the performance of an equivalent ARM processor connected to a 16-bit memory system.

GSM Overview:

A GSM modem is a wireless modem that works with a GSM wireless network. Global system for mobile communication (GSM) is a globally accepted standard for digital cellular communication. GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 mhz.GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards

Sending the message

To send the SMS message, type the following command:

AT+CMGS="+31638740161" <ENTER> Replace the above phone number with our own cell phone number. The modem will respond with:

>

You can now type the message text and send the

message using the <CTRL>-<Z> key combination:

TEST GSM! <CTRL-Z>

Here CTRL-Z is keyword for sending an SMS through the mobile device. After some seconds the modem will respond with the message ID of the message, indicating that the message was sent correctly:

+CMGS: 62

Ultrasonic sensor:

Ultrasonic sensor is used to detect distance between two objects .The phenomenon of the ultrasonic sensor is transmission of ultrasonic signals and receives the reflected signals from the obstacles, based on the time intervals of the transmission and reception signals distance between the objects is to be calculate. This US-020 ultrasonic module can realize 2 ~ 700cm of non-contact ranging function, has 5 V of wide voltage input range, static power consumption is less than 3mA. Their own temperature sensor will revise the ranging and has many communication modes like GPIO, just like watchdog inside, very stable and reliable.

Fig 2: Ultrasonic Sensor

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 seminar 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.

Alcoholic Sensor:

Sensitive material of MQ-2 gas sensor is SnO_2, which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electro circuit, Convert change of conductivity to correspond output signal of gas concentration. MQ-2 gas sensor has high sensitivity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

Fig 3 : Alcoholic sensor (MQ-2)

Temperature sensor:

The LM35 series 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. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4˚C at room temperature and ±3⁄4˚C over a full −55 to +150˚C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1˚C in still air. The LM35 is rated to operate over a −55˚ to +150˚C temperature range, while the LM35C is rated for a −40˚ to +110˚C range (−10˚ with improved accuracy).

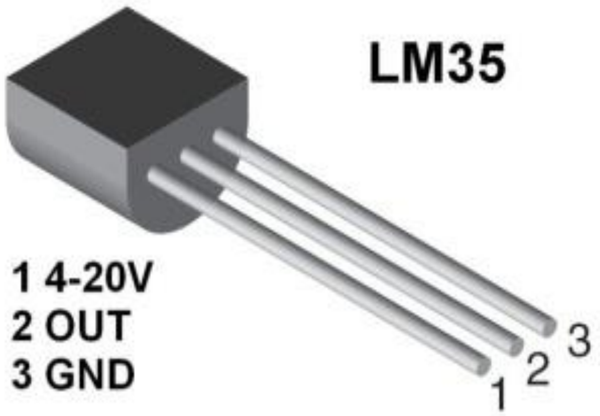

Fig 4: Temperature Sensor

Steering wheel grip sensor:

This sensor works under the principle of piezoelectric effect to measure changes in pressure, acceleration, temperature, strain, or force by converting them to an electrical charge. Based on the pressure applied on the steering wheel it detects grip and warns if it crosses certain time.

GPS Receiver:

GPS receiver locates its position by receiving signals from GPS satellites. The receiver determine the transit time of each signal and computes the distance to each satellite. These distances along with the satellites locations are used with the possible aid of tri-lateration, to compute the Position of the receiver .This position is then displayed, with a moving map display or latitude and longitude.

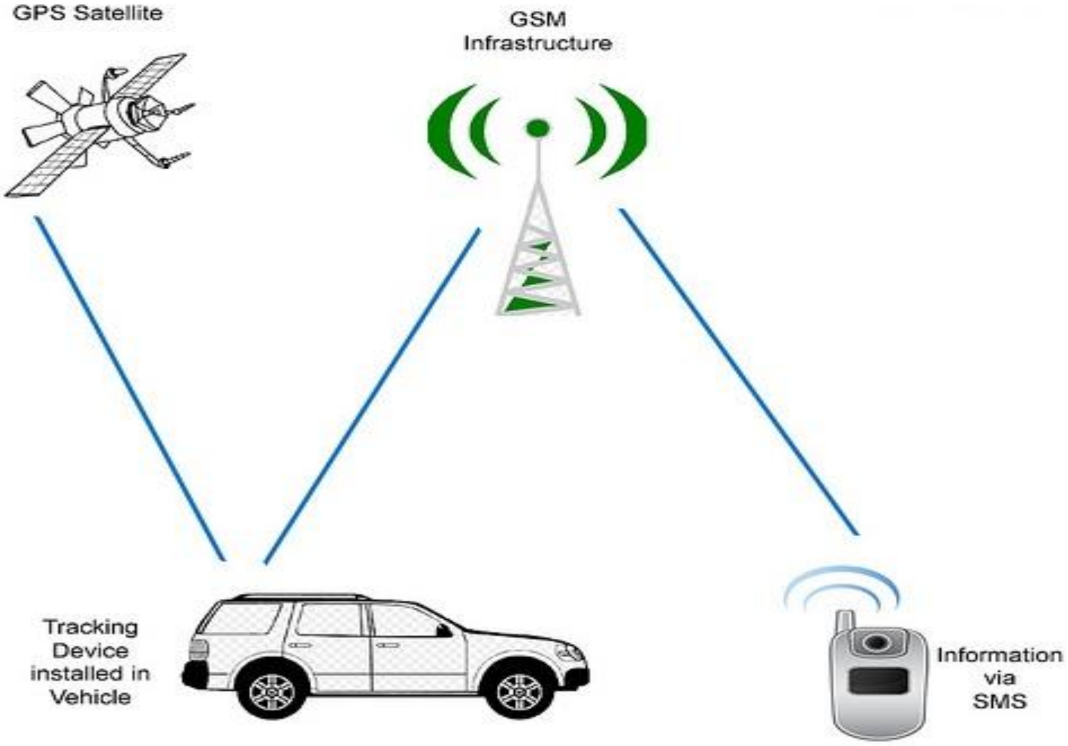

Fig 5: Location tracking using GPS

III. SOFTWARE DESIGN

Software tools:

The software tools used for designing this An Embedded In-Vehicle system are Keil UV5, flash magic.

The Keil Microcontroller Development Kit (MDK) helps us to create embedded applications for ARM Cortex-M processor-based devices. MDK is a powerful, yet easy to learn and use development system. MDK Version 5 consists of the MDK Core plus device-specific Software Packs, which can be downloaded and installed based on the requirements of our application. MDK Core includes all the components that are needed to create, build, and debug an embedded application. The Pack Installer manages Software Packs that can be added any time to MDK Core. For this project software code is developed using RTX Kernel.

The Keil Real-Time Library (RL-ARM) is a collection of easy-to-use Middle-ware components that are designed to work across many different microcontrollers. This allows us to learn the software once and then use it multiple times. The RL-ARM middleware integrates into the Keil Microcontroller Development Kit (MDK-ARM).Programming part for the application that has done in the keil is verified and ported to the ARM board using a tool called flash magic.

Flash Magic is Windows software from the Embedded Systems Academy that allows easy access to all the ISP features provided by the devices. It provides simple and clear user interface and only obtains access to the selected COM Port when ISP operations are being performed.

International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 3, Issue 4, pp: (168-174), Month: October - December 2015, Available at: www.researchpublish.com

Task creation using RTX kernel is as shown below:

```
__task void task1 (void)
  {
     tskID2 = os_tsk_create (task2,0x10);
     tskID3=os_tsk_create(task3,0x10);
     os_tsk_delete_self ();
}
```

IV. RESULTS

It deals with results of different sensors and complete designed model .

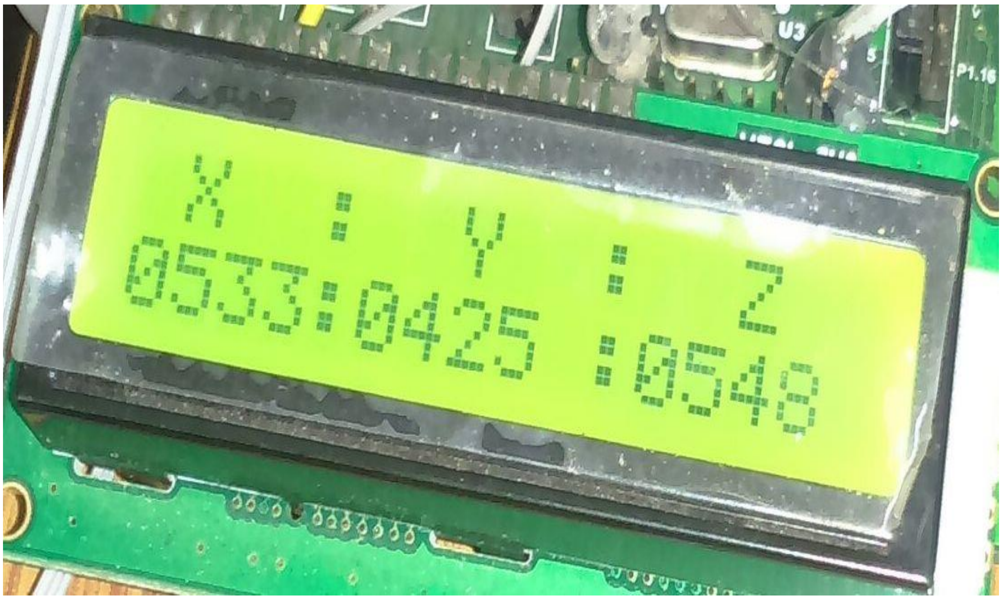

Fig 6: Readings of MEMs accelerometer

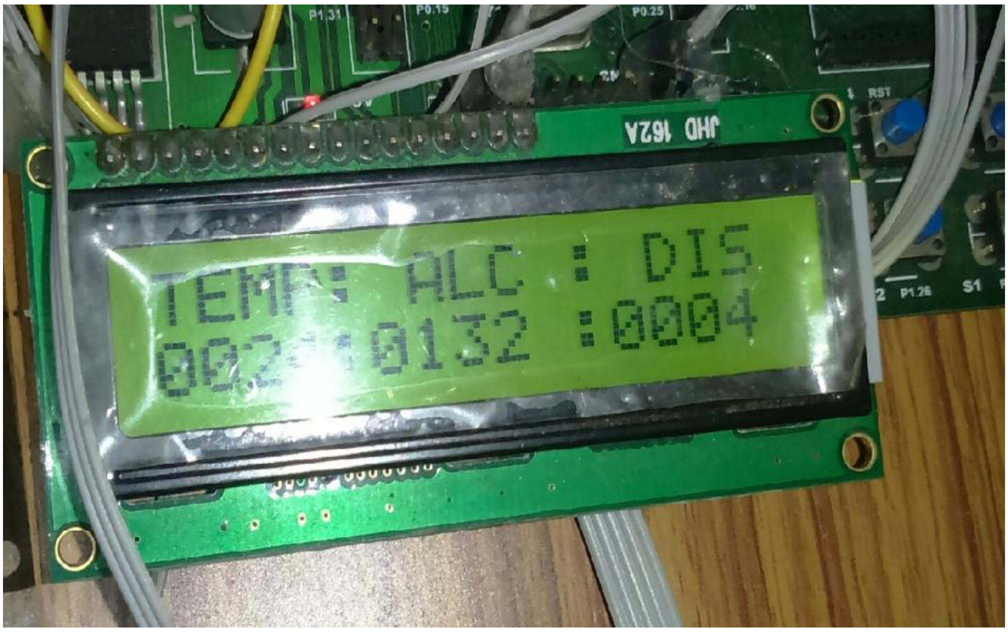

Fig 7 : Readings other parameters

Fig 8: The completed design of an Embedded In- Vehicle System

Fig 6& 7&8 Shows the hardware module of this project. It consists of LPC2148 Board assembled with Different hardware components used to be designed. It shows the results of different sensors for every instant of time and vehicle control corresponding to the abnormalities occurred in vehicle.

V. CONCLUSION

An embedded in-vehicle system ensures secure travelling and the message intimation system through GSM services for abnormal situations. Software module has built up with RTX Kernel by integrating RL – ARM library module with LPC 2148 library module to guarantee the effective working of designed system with the help of scheduling techniques. All the tasks are implemented individually in the previous work in which there is a lack of monitoring other issues. In this design all the tasks are managed using scheduling techniques provided in RTOS. Due to the pipelining structure in ARM 7 processor delay factor between tasks has been enhanced.

ACKNOWLEDGEMENTS

The authors would like to thank the supports given by the Vignan's Lara in terms of financial and technical supervision.

REFERENCES

[1] Zutao Zhang, Jiashu Zhang, "A Novel Vehicle Safety Model: Vehicle speed Controller under Driver Fatigue", "IJCSNS International Journal of Computer Science and Network Security", VOL.9 No.1, January 2012.

[2] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M.Porta, "Artificalvision in road vehicles," Proceedings of the IEEE, vol. 90,no. 7, pp. 1258–1271, 2002.

[3] S. Tsugawa and Sadayuki, "Vision-based vehicle on japan: Machine vision systems and driving control systems," IEEE Trans. on Ind. El.???, vol. 41, no. 4, pp. 398–405, 1994.

[4] Vehicle-highway automation activities in the United States. U.S. Dept of Transportation, 1997.

[5] C. Thorpe, J.D. Carlson, D. Duggins, J. Gowdy, R. MacLachlan, C. Mertz, A. Suppe, and C. Wan, "Safe robot driving in cluttered environments," 11th International Symposium of Robotics Research, 2003.

[6] Qian Martin Eriksson, Nikolaos P. Papanikolopoulos, Eye-Tracking for Detection of Driver fatigue. Proceedingsof the international Conference on intelligent Transportation System, Boston, MA, November 1997, pp.314-319.

[7] QiangJi, Zhiwei Zhu, and PeilinLan, Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue, IEEE Transactions on Vehicular Technology, VOL. 53, NO.4, July 2004, pp.1052-1068.

[8] Hui, S.C. Eleview: Remote intelligent elevator monitoring system, International Journal of Computers and Applications, v 26, No 2, p111- 118, 2004.

[9] Zhang Guangming, Study on city elevator remote monitoring system based on multi-agent, Elevator World, v 54, No 7, p104-111, July 2006.

[10] Phillip A. Laplante, Frederick C. Woolsey, "IEEE 1473: An Open Source Communications Protocol for Railway Vehicles," IT Professional, vol. 5, 00. 6, pp: 12-16, Nov./Dec.2003, DOI: 10.1109/ MITP. 2003.125496.