EMBEDDING RTOS ON ARM 7 ARCHITECTURE

B.C. Goradiya¹, Dr. H.N. Pandya², V.V. Kamdar³

¹Asso. Prof. & Head, Electronics & Telecommunication dept., B.V.M. Engineering college, V.V.Nagar, Rajkot,
²Head & Prof., Electronics dept., Saurashtra University, Rajkot,
³B.E. Electronics & telecommunication dept., B.V.M. Engineering College, V.V.Nagar

ABSTRACT- ARM architecture is one of the latest and upcoming architecture of today's era. This paper is highlighting the use of the ARM architecture for embedding RTOS on basic ARM 7 architecture. Paper mainly emphasis on the implementing the interfaces of LCD, LED, UART and Buzzer. The paper is implementing the multitasking of these four peripherals.

1. INTRODUCTION

In 21st century there is a"race” among individuals to win it. At every moment of life Darwinians applies i.e. the fittest would be the only survivor. So in order to survive I need to have extra qualities talent than others. Here the person who is “jack of all and master of none” has no right to participate in the race. Only the person who is “jack and master of all” has chance to become supreme survivor. One needs to be a multitalented personality and multitasking in order to win the race. My design has all the possibilities to win the race. My design consists of ARM 7 controller and RTOS developed by me that would be using pre-emptive scheduling for the tasks and thus would help in showing multitasking of different peripherals connected to the controller. I have used the evaluation board of the ARM 7 and will be connecting peripherals externally. And the RTOS is developed in keil simulator.

2. GENERAL DESIGN SCHEME OF SYSTEM

Title consists of following parts that are: main controller, RTOS, USB loader, peripherals.

Main controller i.e. Development board of LPC2138, here the LPC2138 is responsible for the control of the whole system and coordination work of each module. RTOS with pre-emptive scheduling i.e. it will give the control to the task with higher priority and complete it first and then carry on the regular scheduling between the peripherals that will show the multitasking between the peripherals, USB loader for loading the program on controller, Man-machine interaction peripherals like LCD, LED, UART, Buzzer that are been multitasked.

3. HARDWARE DESIGN OF SYSTEM

System's development cost was considered, as well as multitasking of various peripherals needs to take massive system resources when it is processing. Therefore, the system selects LPC2138 as the main controller.

LPC2138 is a 32-bit Microcontroller which is produced by NXP Corporation. LPC2138 is ARM7TDMI-S based high-performance 32-bit RISC Microcontroller with Thumb extensions 512KB on-
chip Flash ROM with In-System Programming (ISP) and In-Application Programming (IAP).

Two 8-ch 10bit ADC32KB RAM, Vectored Interrupt Controller, Two UARTs, one with full modem interface. Two I2C serial interfaces, Two SPI serial interfaces Three 32-bit timers, Watchdog Timer, Real Time Clock with optional battery backup, Brown out detect circuit General purpose I/O pins.CPU clock up to 60 MHz, On-chip crystal oscillator and On-chip PLL.

4. SOFTWARE DESIGN OF SYSTEM

![Fig.-2 General Structure of the Software](image1)

Title used C as the software platform, and adopting modularity design. So that it has good portability and expandability. Here RTOS is developed in the software KEIL 4 MDK version. The bottom is drive layer. It is composed of various device drivers, and responsible for provide service interface to the upper layer. The service interface has a uniform data format, and the data format will not change no matter whether hardware equipments change or not. The system will initially load RTX_Config.c,RTL.c,LPC213x.c.

The main body of the program consists of the following thing.

```c
int main (void) {
    /* program execution starts here */
    os_sys_init (job1);   /* initialize and start task 1 */
}
```

Here initially in the main part of the program 1st task is initialized and then rest all the task are initialized in the task 1.

PROGRAMMING THE MICROCONTROLLER

For Loading the hex file generated in KEIL 4 MDK version, Flash Magic software is been used. And following is the screenshot of the Flash Magic software.

![Fig.3-Screen of the Software Flash Magic](image2)
In Flash Magic we need to adjust the baud rate at 9600, oscillation frequency at 12 MHz, and finally we need to select the hex file of the required program. And then click on the START button to load the program.

5. EXPERIMENT AND TEST OF SYSTEM

In order to test the multitasking functionality of the RTOS we conducted many tests were we 1st checked for the simulation results of multitasking between 2 tasks. Then we switched to multitasking of the 3 tasks. Then checked the result after changing the priority of the task. And finally we verified the result of 4 tasks being multitasked were task 1(of blinking the LEDs that are connected to GPIO 1 at pins P1.16 to P1.23 having the highest priority), Task 2 (UART sending data “PROJECT” continuously), Task 3(Printing “PROJECT” on LCD), and finally buzzer which rings on timer interrupt continuously after some fixed time.

Following is the screenshot of the simulation result of 4 task.: 1.) UART Window

![UART Window](image1)

2) Task 4 and Task 2

![Task 4 and Task 2](image2)

3) Task 1 and Task 3

![Task 1 and Task 3](image3)

4) Complete Simulation Screen

![Complete Simulation Screen](image4)

13-14 May 2011                              B.V.M. Engineering College, V.V.Nagar,Gujarat,India
6. CONCLUSION

With the development of economy and society, people set a higher request to multitasking, practicability and accuracy in any system. The system that design in this paper, has many advantages, such as small volume, lower power consumption, lower cost, safety and reliability, easy to install and to maintain, good practicality and so on. It will be an effective means in multitasking system, and get extensive applications in any automation system.

REFERENCES