E&CE 354 S11 RTX Project Part 1

Due at 8:30am on May 16th 2011

(a) Program the second serial port (i.e. UART1) on the MCF5307 board. Put the program source code under a directory/folder named "uart1".

The program echoes the user input on the key board to UART1 until the user hits the "q" key. When the "Return" or "Enter" key is pressed, the cursor should move to the very beginning of the next line (i.e. start a new line) on the second serial port.

(b) Program the first timer (i.e. Timer0) and UART1 on the MCF5307 board. Put the program source code under a directory/folder named "timer0".

When the program starts, it displays a 24 hour clock on the UART1 terminal screen. The clock displays time in the format of hh:mm:ss and starts from 23:59:50. Every second, the clock updates the time.

Hints:
1. How to convert an integer to a string (no C library)?
2. How to display a string to UART1?

NOTE: No polling or busy waiting strategies may be implemented.

(c) Write a program to perform simple memory management of the ColdFire board. Put the program source code under a directory/folder named "memory".

Objective: Implementing a memory pool. Complete the following functions that run in the supervisor mode.

void init_memory()
This function initializes a memory pool in the free memory region. There are 32 memory blocks in the memory pool. Each memory block has a size of *128 Bytes*.

void * s_request_memory_block()
This function returns a pointer to the starting address of a free memory block to the caller. It returns a null pointer when it fails to allocate a memory block. The requested memory block is considered as being in use (i.e. not free) by the caller until a s_release_memory_block() function call is invoked to returns it to the memory pool.

int s_release_memory_block(void* memory_block)
This function returns a memory block that is previously allocated by the s_request_memory_block() function to the memory pool. It returns 0 upon success and non-zero upon failure.

You are provided with a partially finished code in p1_c.zip downloadable from the lab website. Please read the comments in the source code carefully. We will replace the main file with another one during the demonstration. So please do not change the provided main() routine.

Hints:
1. How to determine the starting address of the free memory region?
2. What data structure do you use to organize memory blocks so that you can access memory blocks easily and efficiently?

(d) Document the pseudo code of part b) and c). Name the document p1.doc, p1.docx or p1.pdf.

Deliverables and Demo

Zip items in (**a)-(d)**, name the archived file p1_Gid.zip, where id is your two digit group ID number, and submit it to course book system.

The RTX P1 implementation will be demonstrated to project TAs in E2-2363. All group members are required to attend the demo of the RTX P1. Each group has about 20 minutes to demonstrate the RTX P1. Use course book to book a demo time. An announcement will be made to the class when the course book system is ready to accept RTX P1 demo reservation.