Embedded systems – Exercise #2

In this exercise you will implement several interrupt based device drivers:

1. Timer
2. UART
3. LCD display
4. Input panel
5. Flash
6. Network device.

We will provide you headers files for all the drivers, and you will supply the implementation. For each header file submit a matching source file (excluding for the common_defs.h file obviously).

All your drivers should be interrupt-based, and you should implement your ISR using the "native" MetaWare compiler interrupts macros (e.g. _Interrupt1 & _Interrupt2)

You should implement the assembly code for your Interrupt Vector Table in a file named ivt.s, which will place the code in a locator segment named "INTERRUPT_TABLE".

Your make file should compile each driver to a matching object file, and pack all the resulting object files, together with any other object files that you may have generated to a single library ***ex2.a*** (use the *ar* command).

This mean that for example the for the Timer device implementation you need to submit *timer.c* that will include and implement the interface from *timer.h*, and your make file will generate *timer.o*

In the next exercise you'll be asked to implement a complete system utilizing all the above devices, but that time using ThreadX interrupts mechanisms. Planning ahead will help you port your drivers to ThreadX, but is not a requirement of this exercise.