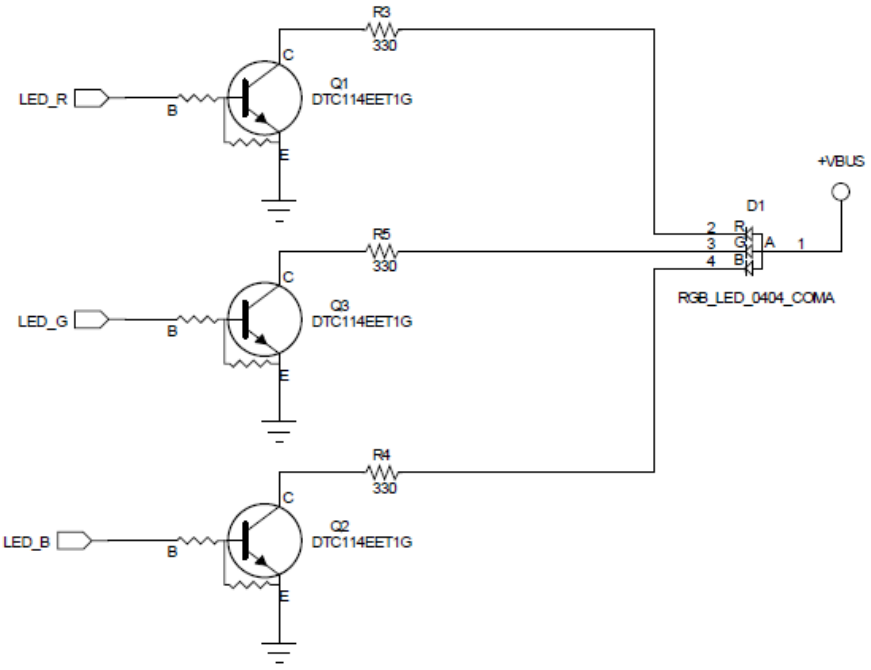
Weekly assignments ROS01

Draft 0.1 - 2013

1. Assignment week 1 (Super loop construct without an ISR)
    a. Install Code Composer Studio (CCS) and the necessary microcontroller libraries.
    b. Create the initial project and make sure you can program and debug the microcontroller.
    c. Create a C macro to dereference a pointer to a volatile unsigned long address.
    d. Set address 0x400FE060 to value 0x02400540. For now, this configures the clock to run at a frequency of 40 MHz. Later library functions will be used to accomplish this.
    e. Configure the GPIO - using the former macro and the addresses found in the datasheet - in order to be able to turn on and off each of the (RGB) leds.
    LED_R = PF1 LED_G = PF2 LED_B = PF3



    f. Now create a rotation loop which shows the following colors: Red(R), Green (G), Blue (B), Yellow (R+G) , Pink (R+B), Aqua (G+B). The colors should rotate with a frequency of about 1 hz (no interrupts necessary). Make use of an enumeration construct for the colors and a switch/case for the rotation. This switch/case will be contained in an infinite loop
    g. Now replace the macro calls with the actual names in the respective register definition header file (lm4f120h5qr.h). To be able to include this file, first add the Stellarisware directory to the list under “project properties->CCS build->Arm Compiler->Include options->Add dir..”.
    h. Save the project

2. Assignment week 2 (Super loop construct with an ISR)
    a. Create a copy of the previous project and rename it to AssignmentWeek2.
    b. Read the datasheet to figure out how to configure the microcontroller to have the main clock run at 80 Mhz. Implement this.
    c. Configure the Systick interrupt to run at a frequency of F_CPU/1000 (1 ms). Make sure the vector is properly called using the debugger.
    d. Use this ISR (Interrupt Service Routine) to replace the functionality of the delay function. You can use a (static unsigned int) software counter to wait a second and set a global flag variable to tell the main loop to continue.

e. Figure out and configure the processor to sleep while a color is showing, until the next systick interrupt. – You have just created a simple and efficient static scheduler!

3. Assignment week 3 (Cooperative scheduler)
   We will expand on the simple scheduler by implementing a list to which functions can be added with a specific period. Thus we need a c struct to combine a function pointer with a period, a counter and a possible initial delay. This struct will describe a 'task'.
   a. Create a copy of the previous project and rename it to AssignmentWeek3.
   b. If you haven't done so, use the peripheral library to replace all the existing register calls.
   c. Using the description of this assignment, define a struct for such a task to create an array of 8 tasks. Create a function to initialize and add other functions to this task list.
   d. In the systick ISR, walk through the task list and decrement each of the task counters.
   e. Think of, and expand on, the task struct to notify per task whether it is in a WAITING, READY or STOPPED state. Set the state in the ISR depending on the task counter.
   f. Create a function runReadyTasks() that will walk through the task list and execute any task in the READY state. Replace your case/switch rotation with this function.
   g. Create 3 functions to toggle each led separately and add this to the task list with periods of: 200 ticks, 500 ticks, 750 ticks for red, green, blue respectively.
   h. Enjoy the show of your 'advanced' cooperative (static) scheduler.

4. This assignment will require the student to create a simple pre-emptive scheduler and thus perform a context switch. This will form the basis for your report.

Version

Author	Comments	Date	Version
VersD	Initial assignments for week 1-3	11-2013	0.1

To-Do

Author	Comments
VersD	Assignments 4-6