ECE 177 Lab 10 Spring 2022

Purpose: More practice with arrays, using pointers and pointer arithmetic. Low-level code for a timer.

Assignment: For this lab you will rewrite the previous lab exercise using pointers. (Note that you must first get the previous lab to work if you haven't previously done so!) For the most part you will simply convert each array indexing operation (using "**`str[i]`**") to a pointer operation (using "**`*p_str`**"). Passing arrays to functions will be done by passing a pointer to the first element of the array. Effectively this is the same as before, but the syntax is different in the function prototype. The call to the function will be unchanged. There should not be a single square bracket ('[' or ']') in your code other than in array declarations (**`mapkey()`** can be unchanged from the previous lab). Use pointers "properly" – do not simply change x[i] to *(x+i). Also, do not use any "counters"; i.e.,, don't say, for example, "`for (i=0; i<4; i++) ...`" Instead, use your pointers to keep track.

You will also write some functions that allow you to delay more precisely. See notes at the end of this description (next page) on how to set up the timer:

`delaycycles()` will delay (fairly) precisely a number of cycles (microseconds) passed to it as an argument using the board's "`Timer 2`". This timer is essentially a "volatile" 32-bit value in memory called "**`TIM2->CNT`**" that we'll set up to auto-magically count up once per microsecond. This timer is enabled in the starter code by writing a "0x0001" to the "**`TIM2->CR1`**" register. If we write a value to **`TIM2->CNT,`** it continues counting from there. When the counter "rolls over" from 0xFFFFFFFF to 0x00000000 it auto-magically sets a "flag" in the least-significant bit of the "**`TIM2->SR`**" register.

So, **`delaycycles()`** will function as follows: Write the negative of the number of cycles to delay to **`TIM2->CNT`**, clear the least-significant bit of **`TIM2->SR`**, then wait for this bit to auto-magically go to 1 and return. (The function contains a total of maybe 3 lines of code.)

`mydelayms()` will be rewritten to include a single `for` loop. The *body* of the loop will call `delaycycles()` to delay exactly 1000 cycles (one millisecond).

Prelab: Rewrite the three CodeLab exercises using pointers. The set of Lab 9 exercises is duplicated in a "Lab 10" section of CodeLab. Note that while simply resubmitting your Lab 9 solutions may give a correct answer in CodeLab, I will check your solution by hand to make sure it has been properly rewritten.

Notes:

The above will get you a B grade. For an A, have your code play tones using the speaker: play a single friendly beep if OPEN LOCK is displayed and four error beeps if the password is entered incorrectly. Hook the speaker up as shown in Lab 7a. To play a tone of a certain duration, use a "`for`" loop whose body simply toggles the speaker and then delays using `delaycycles()`. Delay according to the tone frequency (period is 1/f, so delay half that after each toggle). The `for` loop should loop a number of times determined by the desired duration and the above period. See class notes for further information on how to accomplish this.

Setting up the timer:

We will use Timer 2 on the STM boards. For reference on the registers we are using see the documentation in Section 13.4 beginning page 353 of the reference manual (also linked on the webpage) https://www.st.com/resource/en/reference_manual/dm00096844-stm32f401xb-c-and-stm32f401xd-e-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf

After you create your project from the `.ioc` file, click on the `<projectname>.ioc` item in the project explorer to open it up. On the left click on "Timers" to open it up, then "TIM2". In the "Mode and Configuration" block, change "Clock Source" to "Internal Clock". Then down below expand "Counter Settings" and set "Prescaler (PSC – 16 bits value) to "83". "Counter Mode" should be "Up". Now on the top menu bar for the .ioc file select "Project "Manager" and then on the left select "Advanced Settings" and then expand TIM, then TIM2 and then click "HAL" and change it to "LL".