Exercise 3

Program a timer, using the 8 switches to input the time (using binary code). The board has a 7-segment display (SSD) that must be used to show the remaining time (SSD uses HEX code). When the time reaches 0 the RGBLED must start flickering red. To start the timer, use one of the 5 buttons on the bottom right side of the board.

This exercise uses 3 new components. First, we must use the SSD. Make sure *ssd.c* is included in *main.c* file. To initialize it's functions, call `SSD_Init()` in main function.

Secondly, we must use the RBGLED. It's library is called rgbled.c and it's initializing function is `RGBLED_Init().`

Thirdly, make sure that *btn.c*. It is the buttons' library. To initialize them, use `BTN_Init().`

To display info on SSD, call `SSD_WriteDigitsGrouped(int x, int y)` function. It is a void type function that takes 2 parameters. First one is an integer with the number to be displayed. Second one is for float type information, which we won't be using and should be 0.

To turn on the RGBLED, use `RGBLED_SetValue(int r, int g, int b)` function. The 3 parameters represent 3 colors in RGB scale.

In addition, to read every switch at once, call `SWT_GetGroupValue().` It will return an integer representing the value of all the switches. Similar function for LED is `LED_SetGroupValue(int x).` The parameter equals the value of the desired LED indicators.

For example, 0 turns off every LED, 1 turns on only the rightmost LED, 2 turns on the 2. rightmost LED, 3 turns on both right most LED lights and so on. Every LED light up if the parameter equals with 255.

The function to read button's input is `BTN_GetValue(char x).` It returns 1 if button is pressed and 0 if button is not pressed. It takes 1 parameter, which represents the button you desire to read. This parameter is a char type. Every button has a letter representing it. For example, the middle button is 'c' and the upper button is 'u'.