

OSSES Assignment #1

Report by

De Petris Paolo - 233179

The general idea to develop the alarm system was to use:

- A flex timer, basically used to blink the onboard led
- An ADC, used to convert the voltage analog input into an output digital value

In order to satisfy the requirements the following algorithm is been implemented :

In the **main** function there are all the variables initialization and the call to:

- **setup_ftm0**, which is responsible to properly initialize the flex timer. When invoked, this function first of all enables the interrupt signals and install the interrupt service routine. Then it turns on the timer acting on the System Clock Gating Control Register 6 (SCG6) and set it in debug mode. After these instructions another mask is applied to enable the modifications on the timer. The last instruction allows to write in the registers. The return value is type void, because it is just a setup function.
- **setup_ADC0**, which is responsible to properly initialize the ADC. Similarly to the timer, also the ADC must be activated. Also in this function there is the modification of some values in the registers to get the desired behaviour and the enabling of the interrupt signals for the ISR (see comments on the code for specifics). The **ADC_offset** has been added in order to correct the analog defect (when I gave 0V input the real value was not 0).

The **AppStartupTask** has the only purpose to create the AppTask, which is the function that basically does all the work.

Inside the **AppTask**: After some hardware initializations, I created an endless loop. At each round the function **ADC_read16b** is called (to get the analog value - range 0-3.3V from pin PTB2). The proper range and color are set, corresponding to the read value. A further 'if' condition verifies in which range of frequency corresponds the read one and, if necessary, stores the desired value of MOD in a global variable.

The function to update the frequency is **update_ftm0**: first of all the counter inside the timer is reinitialized, in order to have a clear count on the next read value. Then the MOD value is updated to obtain the desired frequency (10,20 Hz).

The two handlers (**BSP_FTM0_int_hndlr** and **ADC0_int_hndlr**) are necessary to tell the OS that the ISR is going on, both when there is a read on the ADC and when there is a toggle of a color.