POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Operating Systems for Embedded Systems

Assignment Report

Gabriel GANZER
271961

A.Y. 2019/20

1. Introduction

The subsequent report documents the procedures conducted during the project of a tunable lamp using Micrium µC/OS3 operating system on an S32K144 board. Further comments regarding methods, materials, and results would be presented in the following sections.

2. Methods and Materials

2.1. I/O Devices

First and foremost, the assignment requires using the RGB LED as a digital pulse-modulated discrete output, i.e., to configure the pins associated with those LEDs to work in $ALT2$ mode. The macro expression *PORT_PCR_MUX(2)* was used to achieve that.

The lamp is initialized with the red color, moreover changing its color within a given sequence whenever switches $SW2$ and $SW3$ are pressed. Since such action occurs sporadically throughout execution, both inputs were implemented via rising edge interrupts, as polling would entail the software's constant check of their control bits, blocking the execution of further instructions.

The assignment also requires that whenever one of the switches remains pressed, the color should not change. A common practice to resolve long-pressing involves using delays, by asserting sort of a deadline to the inputs. However, such an approach is unsuitable if used inside an interrupt handler. Therefore, during switches initialization, a digital input filter was configured by writing the pins macros inside the $DFER$ register, enabling it for those pins. Then, the clock source was selected by writing into the $DFCR$, and finally, register $DFWR$ was set to filter any glitches in $PORTC$ after 384000 cycles, i.e., ignoring the interrupt if any switch is pressed for more than 3 seconds.

2.2. ADC Converter

The LED's intensity is proportionally adjusted to a potentiometer position, thus, an ADC converter is necessary to read its value and generate the corresponding response. In this project, converter was configured to work in the software triggered operation mode.

The conversion completeness could be checked either via polling or interrupts, again, to avoid endless checking and waste of resources the chosen implementation involved interrupts. Additionally, the ADC's interrupt handler uses a semaphore for synchronizing and notifying the initiator that a conversion was completed.

The ADC produces a binary value from 0 to 4095 in a $12 - bit$ conversion. The potentiometer's analog input ranges from $0V$ to $5V$, hence, a conversion from the correspondent binary value to its engineering equivalent is necessary before usage, given by equation 2.1.

$$ADC_{result} = \frac{5000 * Output_{binary}}{\texttt{0xFFF}} \tag{2.1}$$

2.3. FlexTimer Module

The LED's pins must be activated by a 10 kHz PWM signal, on that account, timer $FTM0$ that is associated with $PORTD$ was used. This feature was configured to enable its register updating from write buffers, i.e., allowing synchronization, by setting $FTMEN = 1$ into the $MODE$ register. Moreover, both $SYNCEN0$ and $SYNCEN1$ fields from $COMBINE$ register were enabled, so $C(n)V$ and $C(n+1)V$ of pair channels $n = 1$ and $n = 2$ could be updated, as a requirement from the enhanced PWM synchronization protocol.

The modulo register MOD was then set to the FTM counter value, correspondent to a 10 kHz frequency at 80 MHz clock frequency and pre-scaler equals 1, according to equation

2.2, as well as the initial value set to zero in $CNTIN$ register.

$$Modulo = \frac{f_{clock}}{Divider.f_{PWM}} - 1 = (8000 - 1) \tag{2.2}$$

The $CnSC$ register that contains channels' status and control bits was selected to operate in edge-aligned mode with high-true pulses, achieved by setting MSB and $ELSB$ fields as 1 and $ELSA$ as 0. This type of PWM signal forces the leading edges to be aligned with the beginning of the period and clearing the output on a match, as shown in figure 2.1.

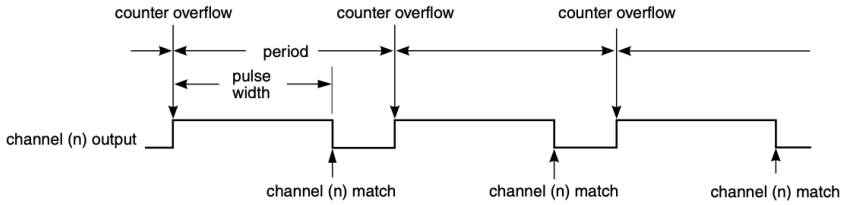

Figure 2.1: Edge-aligned PWM signal with high-true pulses.

Such an approach is correlated with the LED's and potentiometer's functionality, whereas the PWM duty cycle is 0% with a value of 0V read in the ADC converter, resulting in a continuous high pulse in the PWM signal, hence, deactivating the LED. Similarly, when the ADC's value is equal to 5V, the duty cycle would be 100% and LED is always on.

The $FTM0$ was configured to support software triggered synchronization, by setting $SYNCMODE$, $SWWRBUF$, and $SWRSTCNT$ fields to logic 1 inside $SYSCONF$ register. Moreover, the registers are updated with their buffer value by setting $SWSYNC$ bit to 1, which generates a trigger event. As $SWRSTCNT = 1$ the $SWSYNC$ bit is cleared immediately after the synchronization has ended. Additionally, the output mask software update field $SWOM$ was activated, as the mask of a channel determines if its output responds when a match occurs, forcing to its safe value set at the POL register when the mask is 0. In this project, the polarity is high since LEDs are deactivated with logic 1.

Finally, clock source, pre-scaler and PWM generation for the respective channels are enabled by writing into the fields of SC register. Counter values CnV and event trigger were implemented inside the function **BSP_LED_On**. Equation 2.3 is used to determine the counter values, taking into account the modulo range and ADC value.

$$DutyCycle = 1.6 * ADC_{result} \tag{2.3}$$

3. Results and Discussion

The Micrium µC/OS3 managed two tasks, the modified startup task and a second one designed to get the potentiometer value from the ADC converter. Since both tasks share the variable storing PWM period, the project required an additional semaphore, whose purpose was of protecting the critical section where that variable was either written or used, i.e., solving the mutual exclusion problem between those two tasks.

Bibliography

[1] NXP Semiconductors. *S32K1xx Series Reference Manual*. 2018.

[2] NXP Semiconductors. *Features and Operation Modes of FlexTimer Module on S32K*. 2016.

[3] NXP Semiconductors. *S32K1xx Series Cookbook*. 2019.