ECSE 426 - Microprocessor Systems
Lab Report 2: Timers, Interrupts, Multithreaded, Interrupt-Driven Readings and Peripheral Control

Harley Wiltzer (260690006)
Matthew Lesko (260692352)

March 19, 2018

Contents

List of Figures

List of Tables

1 Abstract

The purpose of experiment 3 is for the programmers to gain experience in utilizing timers and interrupts to accomplish a task which involves converting an analog pulse to digital and displaying its voltage on an LED display, effectively a voltmeter. The purpose of experiment 4 is for the programmers to gain exposure in designing a multithreaded application on a real time operating system (RTOS) running on an embedded system. The task of experiment 4 involves copying over experiment 3's program and subdiving several of its features each to its own concurrently running thread, with the goal of optimizing power usage. This report will explain in detail how the programmers implemented the problems stated below, as well as the challenges they faced, the testing they had done, and the conclusions they have made. By the end of the report, the reader shall understand how the timers available on the STM32F4 board can be used to activate peripherals and generate a pulse, and understand the implementation of multithreading on embedded systems.

2 Problem Statement

The problem is for the developers to implement a solution for generating a PWM pulse, whose voltage is set by an input on a keypad, that is fed to a rectifier and have the output fed to an ADC to be converted to a digital signal, and finally having its voltage be automatically displayed on an LED display. Furthermore, the program has to be implemented with the use of concurrently-running threads running on an RTOS. The problem can be divided into the following tasks:

- Setting up a timer to act as a PWM pulse generator
- Configuration of the ADC to be activated by a timer
- Design of a rectifier circuit component that takes the PWM pulse as input and have its output be converted to digital by an ADC
- Testing and Optimization of an FIR Filter that reduces noise from the output of the rectifier
- Setting up the alphanumeric keypad so that the user may input their desired voltage to be displayed
- Mainting the 7-segment display
- Coding a controller component that automates the changes to be made on the PWM's duty cycle so that the correct voltage appears on the LED display
- Implementation of the program's features using CMSIS-RTOS and multithreading
- Reducing the power consumption of the product when it is in sleep mode using CMSIS-RTOS

Appendix A

GPIO Configuration Parameters

This appendix lists the configuration parameters set for each of the different GPIO pins (or classes of GPIO pins).

User Input Button

Parameter	Value
Mode	GPIO_MODE_IT_RISING
Pull	GPIO_NOPULL

Display Mode LEDs (4 of these)

Parameter	Value
Mode	GPIO_MODE_OUTPUT_PP
Pull	GPIO_NOPULL
Speed	GPIO_SPEED_FREQ_LOW

Display Segment Pins (8 of these)

Parameter	Value
Mode	GPIO_MODE_OUTPUT_PP
Pull	GPIO_NOPULL
Speed	GPIO_SPEED_FREQ_LOW

Display Selector Pins (3 of these)

Parameter	Value
Mode	GPIO_MODE_OUTPUT_PP
Pull	GPIO_NOPULL
Speed	GPIO_SPEED_FREQ_LOW

Appendix B

ADC Configuration Settings

ADC Instance Parameters

Parameter	Value
Clock Prescaler	ADC_CLOCK_SYNC_PCLK_DIV2
Resolution	ADC_RESOLUTION_8B
Scan Conversion Mode	Disabled
Continuous Conversion Mode	Disabled
Discontinuous Conversion Mode	Disabled
External Trigger Conversion Edge	ADC_EXTERNALTRIGCONVEDGE_RISING
External Trigger Conversion	ADC_SOFTWARE_START
Data Alignment	ADC_DATAALIGN_RIGHT
Number of Conversions	1
DMA Continuous Requests	Disabled
EOC Selection	ADC_EOC_SINGLE_CONV

ADC Channel Parameters (Channel 1)

Parameter	Value
Rank	1
Sampling Time	ADC_SAMPLETIME_3CYCLES

Appendix C

HAL Cube MX Autogenerated Code

Appendix D

Theory References