Exercise 2: ADC with Keyboard Interrupt

EG-252 Group Design Exercise – Microcontroller Laboratory

Dr K. S. (Joseph) Kim Dr Chris P. Jobling

September 2020

I. Overview

For this lab exercise you are provided a sample ADC assembly program given in the appendix. An electronic version of the program is available on the Blackboard site. The program uses interrupt generated by push buttons to trigger an ADC process on the MC9S08AW60 evaluation board. You are to carry out the following two tasks with this exercise:

- Use the sample program to practice on pushbutton with interrupt mechanism and ADC process with the evaluation board.
- Design an equivalent program in C language which can perform the same keyboard interrupt and ADC processing functions as provided by the example assembly program.

This exercise is worth 8 marks. For this exercise you need only convert the provided assembly language programme to C and submit it for assessment. The assessment asks some additional questions related to the set up of the ADC and its use in the micromouse project.

You can view this document as a web page HTML, PDF or as a Word Document .docx

Appendix

Sample Program in Assembly

```
;*****************************************************************
;*        kbi_adc.asm                                            *
;*                                                               *
;*        MC9S08AW60 Evaluation board keyboard interrupt example *
;*        - Switch SW3 onboard connected to Port D pin 3, KBI pin6;  *
;*        - Switch SW4 onboard connected to Port D pin 2, KBI pin5   *
;*                                                               *
;*        Function:                                              *
```

```
;*        On reset all LEDs will light on. If SW3 or SW4 pressed,              *
;*  an interrupt is generated, which set LEDs 0:3 to light on.           *
;*        More interrupts are genereated if SW3 or SW4 are pressed.            *
;*****************************************************************************

                INCLUDE        'derivative.inc' ; Include derivative-specific definitions

FLASH        EQU                $2000
RAM                EQU                $0070
WATCH        EQU                $1802

                ORG                RAM
LED_on        DS.B        1                ; Define a variable VAR_D with a size of 1 byte

;Start program after reset
                ORG                FLASH
START_UP
                LDA                #$00
                STA                WATCH        ; Turn off the watchdog timer

;Init_GPIO init code
                LDA     #$FF
                STA     PTFDD
                MOV     #$0F, LED_on        ; Initialize VAR_D, used to control the LEDs
                LDA     #$FF
                STA     PTDPE           ; Port D is enabled with pull-up
                RSP                                                ; Reset stack pointer

;Enable interrupt for Keyboard input
                LDA     #$60
                STA     KBI1PE           ; KBI1PE: enable KBI function for pins 5 and 6 only
                BSET    $02, KBI1SC      ; KBI1SC: KBACK=1, to clear KBI flag
                BSET    $01, KBI1SC      ; KBI1SC: KBIE=1, enable KBI

                CLI                      ; Enable interrupt

MAINLOOP
                LDA     LED_on           ; Simple loop
                BRA                MAINLOOP

;Interrupt service routine for a keyboard interrupt generated upon the press of a pushbutton
;with a falling edge (transition from high logic level "1" to low logic level "0")
LED_SWITCH
                BSET    $02, KBI1SC     ; Clear KBI flag
                LDA     #8
                STA     ADC1SC1         ; ADC conversion will start after a number is writte
```

```
55 ADCLOOP
56                  TST     ADC1SC1         ; Check the COCO bit (conversion complete flag).
57                  BPL     ADCLOOP         ; if not complete, wait in the ADC loop.
58                  LDA     ADC1RL          ; if complete, read the ADC outcome (digital value)
59                  STA     PTFD            ; display over LED bar
60                  RTI
61
62 ;INT_VECTOR
63                  ORG     $FFD2
64                  DC.W    LED_SWITCH
65
66                  ORG     $FFFE
67                  DC.W    START_UP
```

View on GitHub