2DX4: Microprocessor System
Lab 1

Instructor: Drs.Boursalie, Doyle, Haddara

Lab TAs: Fatemeh Derakshani (derakhsf), Han Zhou (zhouh115)

Junbo Wang – L01 – wangj430

Yichen Lu – L01 – luy191

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is our own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by **[Junbo Wang wangj430 400249823]**

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is our own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by **[Yichen Lu luy191 400247938]**

1. Purpose

The objective of this lab is to gain a better understanding on how to program the microcontroller in order to detect and manipulate simple digital signals by using assembly codes.

2. BackGround

There are two milestones in this lab. Both milestones must use the assembly code to program the microcontroller. The first milestone is associated with studio 1B, which is to output a logic high to GPIO Port F Pin 4, D3 LED. However, for milestone 1 we want to blink GPIO Port N Pin 0, which is D2 LED on the microcontroller board.

In terms of milestone 2, it is associated with studio 1B and studio 2. The concept of active low push button with pull-up resistor is used in milestone 2. For the LED, we want GPIO Port N pin1, D1 LED to blink when we push the button. In studio 1B, we want to output a logic high to GPIO Port F Pin 4, D3 LED.

3. Method

Milestone 1

The first one is using assembly codes to write a program to blink one of the LEDs of Port N from the microcontroller. We chose GPIO Port N Pin 0, which is a D2 LED on the microcontroller board. So firstly, we change the address of Port F that is in template into the address of Port N. Afterwards, we change the number that sets up the pin and port for RCGPIO, GPIO DIR, GPIO DEN and GPIO DATA into the Port N and Pin 0 respectively. Finally, we copy and paste the loop code that the professor provided to us.

Milestone 2

The second one is wiring an active low push button to control the GPIO Port N pin1, D1 LED on the microcontroller. In this case, we use Port M as the input. Firstly, we fill out the address of Port M and N. Then we edit the RCGPIO, GPIO

DIR, GPIO DEN and GPIO DATA for both Port M and N respectively. Since Port M is input and digital, the number for GPIO DIR and GPIO DEN are 0*0 and 0*1 respectively. At the end, we fill out the loop code in the template. More specifically, we use an XOR function to let the output value be 1 and LSL to do a left shift to pin1 in order to turn on the LED. In the end, we load the result to Port N and store it to perform the LED D1 light up when we push the button.

4. Results

Milestone 1

Milestone 2

5. Observation and conclusion

From the milestone1, we wrote the program using the Port N address and observed that the D2 LED is blinked. For the milestone 2, when we pushed the button, the D1 LED lit up; and when we released the button, the LED would be off. Therefore, we programmed the microcontroller by using assembly codes and detected simple digital signals.

Code Appendix

Lab1_Milestone1

```
; McMaster 2DX4
; Function:
; Modify this Lab 1 Milestone 1
; Runs on MSP432E401Y
; Yichen Lu;Junbo Wang
; Student Number: 400247938 400249823
; MACID: luy191 wangj430
; Last Modified: January 30th 2022

; Original: Copyright 2014 by Jonathan W. Valvano, valvano@mail.utexas.edu
                                                ; Note: You Need to Update Port N
(used in Studio 1) to Port F (used in Lab 1). Review Studio 1 for more details.
SYSCTL_RCGCGPIO_R            EQU 0x400FE608  ; General-Purpose Input/Output Run
Mode Clock Gating Control Register
GPIO_PORTN_DIR_R             EQU 0x40064400  ; Access Port Pins 0 - 7  for Port
N
GPIO_PORTN_DEN_R             EQU 0x4006451C  ; Direction Register for Port N
GPIO_PORTN_DATA_R            EQU 0x400643FC  ; Digital Enable Register for Port
N
COUNTER                      EQU 0xFFFFF


        AREA    |.text|, CODE, READONLY, ALIGN=2
        THUMB
        EXPORT Start



;Function PortN_Init
PortN_Init
    ; Studio 1 STEP 1
    LDR R1, =SYSCTL_RCGCGPIO_R
    LDR R0, [R1]
    ORR R0,R0, #0x1000
    STR R0, [R1]
    NOP
    NOP

    ;Studio 1 STEP 5
    LDR R1, =GPIO_PORTN_DIR_R
    LDR R0, [R1]
    ORR R0,R0, #0x1                     ;Or in binary 00000001
    STR R0, [R1]

    ;Studio 1 STEP 7
    LDR R1, =GPIO_PORTN_DEN_R
    LDR R0, [R1]
    ORR R0, R0, #0x1
    STR R0, [R1]
    BX  LR                              ; return to Start
```

```
Start
      BL  PortN_Init                ; The BL instruction is like a function
call

      ;Studio 1 STEP 8
      LDR R1, =GPIO_PORTN_DATA_R
      LDR R0,[R1]
      ORR R0,R0, #0x1
loop  STR R0, [R1]
      EOR R0, R0, #0x1              ;Note: You are using an OR here you need an
XOR (What opcode should you use?)

      LDR R4, =COUNTER               ;Copy Loop Code From Avenue Here, How does
the loop code work?
loop1 NOP
      NOP
      SUB R4,R4,#0x1
      CMP R4, #0x00
      BNE loop1


      B loop
      ALIGN                         ;Make sure the end of this section is
aligned
      END
```

Lab1_Milestone2

```
;****************************************************************************
;****************************************************


; Name: Junbo Wang;Yichen Lu
; Student Number: 400249823;400247938
; MACID: Wangj430;luy191
; Lab Section: L01
; Description of Code: Turns on User LED D1 with the push of an external button


; Original: Copyright 2014 by Jonathan W. Valvano, valvano@mail.utexas.edu



;****************************************************************************
;****************************************************

;ADDRESS DEFINTIONS

;The EQU directive gives a symbolic name to a numeric constant, a
register-relative value or a program-relative value


SYSCTL_RCGCGPIO_R            EQU 0x400FE608  ;General-Purpose Input/Output Run
Mode Clock Gating Control Register (RCGCGPIO Register)
GPIO_PORTN_DIR_R             EQU 0x40064400            ;Fill in the address for
your assigned LED
GPIO_PORTN_DEN_R             EQU 0x4006451C            ;Fill in the address for
your assigned LED
GPIO_PORTN_DATA_R            EQU 0x400643FC            ;Fill in the address for
your assigned LED

GPIO_PORTM_DIR_R             EQU 0x40063400            ;GPIO Port M Direction
Register Address (Fill in these addresses)
GPIO_PORTM_DEN_R             EQU 0x4006351C            ;GPIO Port M Direction
Register Address (Fill in these addresses)
GPIO_PORTM_DATA_R            EQU 0x400633FC            ;GPIO Port M Data
Register Address      (Fill in these addresses)

;****************************************************************************
;****************************************************

;Do not alter this section

        AREA    |.text|, CODE, READONLY, ALIGN=2 ;code in flash ROM
        THUMB                                    ;specifies using Thumb
instructions
        EXPORT Start

;****************************************************************************
;****************************************************


;Function PortN_Init
```

```
PortN_Init
   ;Studio STEP 1
    LDR R1, =SYSCTL_RCGCGPIO_R
    LDR R0, [R1]
    ORR R0,R0, #0x1000
    STR R0, [R1]
    NOP
    NOP

   ;Studio STEP 5
    LDR R1, =GPIO_PORTN_DIR_R
    LDR R0, [R1]
    ORR R0,R0, #0x2                    ;Or in binary 00000010
    STR R0, [R1]

   ;Studio STEP 7
    LDR R1, =GPIO_PORTN_DEN_R
    LDR R0, [R1]
    ORR R0,R0, #0x2                    ;Or in binary 00000010
    STR R0, [R1]

PortM_Init
    ;Studio STEP 1
    LDR R1, =SYSCTL_RCGCGPIO_R
    LDR R0, [R1]
    ORR R0,R0, #0x800
    STR R0, [R1]
    NOP
    NOP

    ;Studio STEP 5 (Please note that port M is our input port)
    LDR R1, =GPIO_PORTM_DIR_R
    LDR R0, [R1]
    ORR R0,R0, #0x0                    ;input
    STR R0, [R1]

    ;Studio STEP 7
    LDR R1, =GPIO_PORTM_DEN_R
    LDR R0, [R1]
    ORR R0,R0, #0x1                    ;digital
    STR R0, [R1]
    BX  LR

Start                              ; Your assembly code starts executing here
    BL  PortM_Init                 ;The BL instruction is like a function call
to initalize Port M
    BL  PortN_Init                 ;The BL instruction is like a function call
to initalize Port N

Loop LDR R0, =GPIO_PORTM_DATA_R  ;Save the address of Input Port M to R0
     LDR R1, [R0]                ;Save the contents at Input Port M to R1

     EOR R1,R1,#0x1                      ;Flips the input bit value at R1 bit
0: 0-> 1 or 1 -> 0 (remember push button is active low, so when we push we get
0 and we need 1 to turn on LED)
     LSL R1,R1,#1                        ;Left shift R1 x bits to the left
depends on student number, just use shift of zero here if not
```

```
LDR R0, =GPIO_PORTN_DATA_R
STR R1, [R0]

B Loop

ALIGN
END
```