

## Preliminary Work

WA-1

## Experiment-5

1) Subroutine for the Full Step Mode step motor control is added as appendices.

2-4) In Fig.1, **Red** texts shows the MCU connections and **Blue** texts shows the Step Motor connections. Figure 2 shows the connection of 4 switch on the Keypad and the MCU. (B4-7 pins are the outputs of the MCU and E0-3 pins are the input of the MCU)

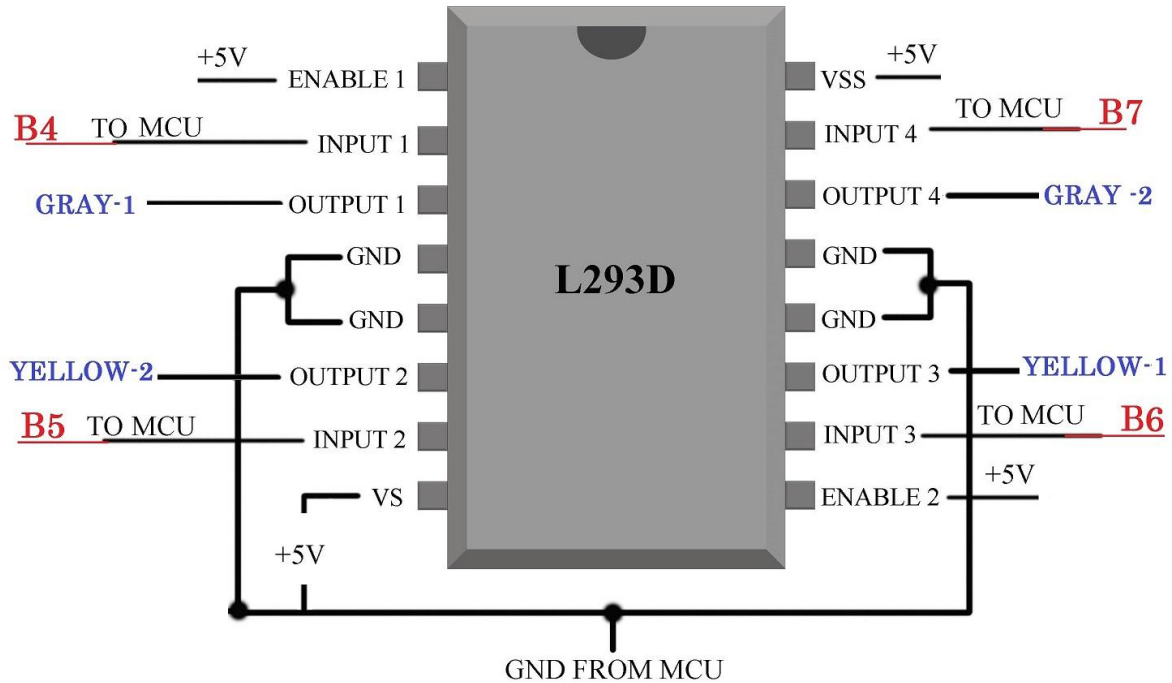


Figure 1: L293D and the MCU and the Step motor connections.

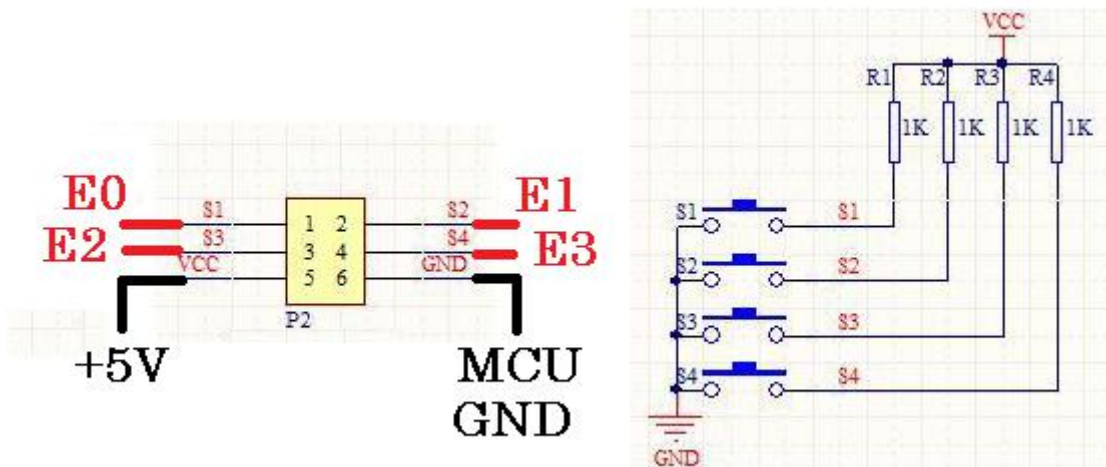


Figure 2: Switch connections of the Keypad model and the MCU.

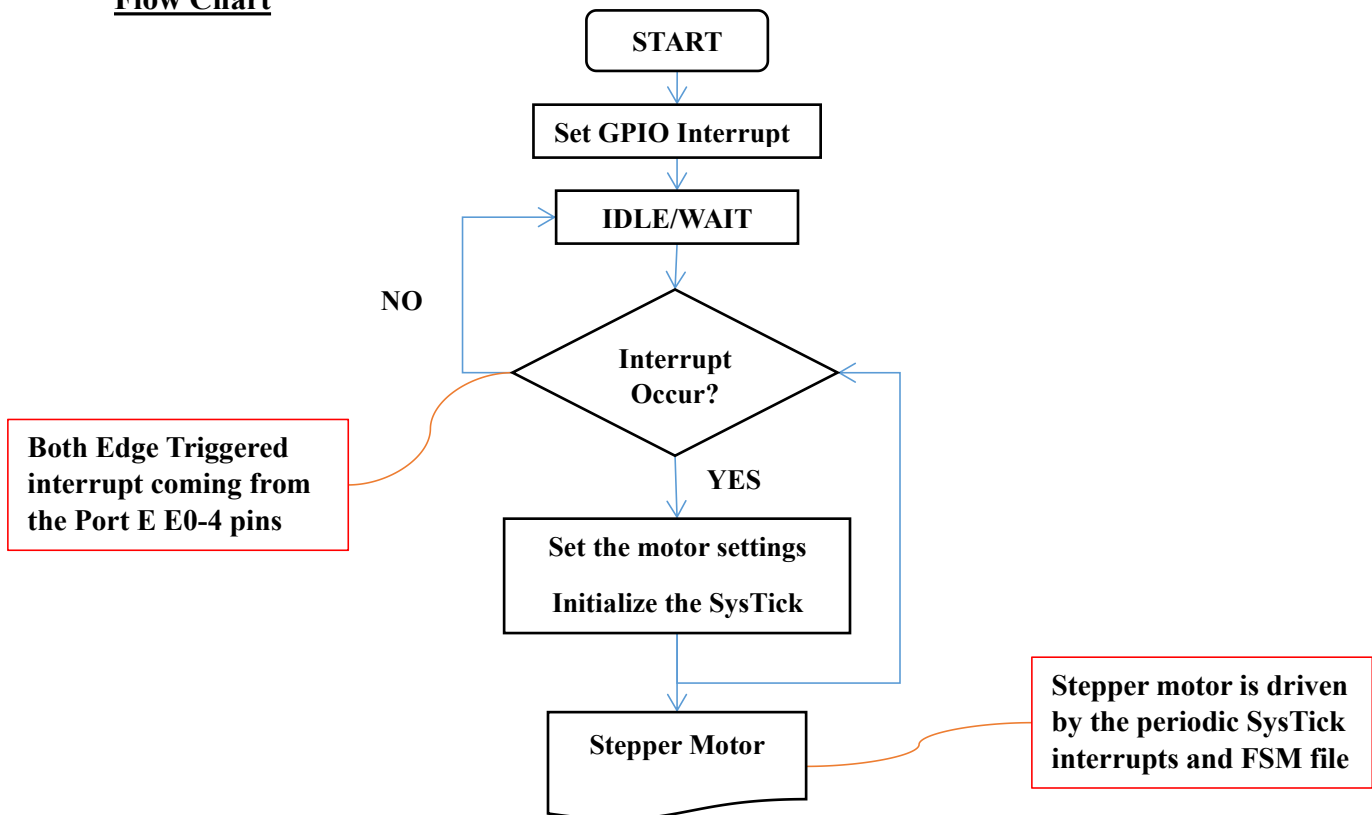
**Note:** For part 2 only E0 and E1 is used. (E0: CCW E1: CW E2: Speed Down E3: Speed Up)

3-5) Program codes for both the design in part-2 and part-4 is added to appendices.

## Preliminary Work

5) For this part, each release of the buttons generate an interrupt with higher priority and according to pressed button relative changes is done on the stepper motor driver codes that is actually controlled by SysTick interrupt. I write a subroutine that apply two input to the stepper motor continuously, which are speed and direction. Speed determined by the frequency of the SysTick subroutine and direction determined by the left or right shift in the FSM subroutine codes. R3&R6 registers is the direction input of the Stepper motor driver, which is FSM and frequency determined by R4&R7 register.

### Flow Chart



## Appendices

**Part 3)** For the coding of part 3 the Startup.s file is updated as follows:

```

334      IMPORT e5p3_ST_ISR
335      ;IMPORT e5p5_ST_ISR
336  SysTick_Handler PROC
337      EXPORT SysTick_Handler           [WEAK]
338      PUSH{LR}
339      BL      e5p3_ST_ISR
340      ;      BL      e5p5_ST_ISR
341      POP{LR}
342      BX      LR
343      ;      B       .
344      ENDP
  
```

# EE 447-Introduction to Microprocessors

## Preliminary Work

**Part 5)** For the coding of part 5 the Startup.s file is updated as follows:

```
335             IMPORT FSM      - -
336 SysTick_Handler PROC
337             EXPORT SysTick_Handler      [WEAK]
338             PUSH{LR}
339             ;BL      e5p3_ST_ISR
340             BL      FSM ; full step mode driving (R3 input)
341             POP{LR}
342             BX      LR
343 ;
344             ENDP

-
467
468             IMPORT e5p5_GPIO_ISR
469 GPIOPortE_Handler PROC
470             BL      e5p5_GPIO_ISR
471             ENDP
```

Following pages is respectively include,

- 1) Part 1 -FSM subroutine code, file name is “Exp5\_p1.s”
- 2) Part 3 -Main code of part 3, file name is “Exp5\_p3.s”
- 3) Part 3 -SysTick subroutine of the part 3, subroutine and file name is “e5p3\_ST\_ISR.s”
- 4) Part 5 -Main code of part 5, file name is “Exp5\_p5.s”
- 5) Part 5 -GPIO timer subroutine code of part 5, subroutine and file name is “e5p5\_GPIO\_ISR.s”

```

;*****
; Experiment 5 ; Preliminary Work Part 1;
; ----- FSM Subroutine -----
; GPIO Full Step Mode settings for step motor, | one step motion
;*****
; INPUT : R3 direction -| 0xF0 -> clockwise |- -| 0x0F ->counterclockwise |-
; Outputs: B4: out1      B5: out2
;          B6: out3      B7: out3
GPIO_PORTB_DATA EQU 0x400053FC

;LABEL    DIRECTIVE    VALUE        COMMENT
;AREA      routines, READONLY, CODE
;THUMB
;EXPORT    FSM

FSM PUSH{R0,R1,R2}
    MOV R2,#0xF0      ;temp
    CMP R3,R2
; get port B current data
    LDR R1,=GPIO_PORTB_DATA
    LDR R0,[R1]
    BFC R0,#0,#8 ; clear R0
    LDRB R2,[R1] ; current pos. R2
    BNE ccw
    LSR R2,#1        ;ClockWise
    CMP R2,#0x08
    BNE upd ; pass
    MOV R2,#0x80;if 0000 1000 then Reset to #0x80
    B upd

ccw LSL R2,#1        ;CounterClockWise
    CMP R2,#0x100
    BNE upd ; pass
    MOV R2,#0x10 ; if 0001 0000 0000 then Reset to #0x10
upd ORR R0,R2
    STR R0,[R1]

    POP{R0,R1,R2}
    BX LR

ALIGN
END

```

```

;*****
; Experiment 5 ; Preliminary Work Part 3;
; Step motor one step rotate control by buttons
;*****
; INPUT : 2 switch button connected to D0 CCW and D1 CW
STCTRL      EQU 0xE000E010 ; SysTick | +4 STRELOAD | +8 STCURRENT

SYSCTL_RCGCGPIO      EQU 0x400FE608 ; clock
GPIO_PORTB_DIR        EQU 0x40005400 ; 0: input | 1: output
GPIO_PORTB_AFSEL      EQU 0x40005420 ; disable AFSEL & PCTL
GPIO_PORTB_DEN        EQU 0x4000551C ; digital Enabled
GPIO_PORTB_AMSEL      EQU 0x40005528 ; analog Disabled
GPIO_PORTB_DATA       EQU 0x400053FC
GPIO_PORTE_DIR        EQU 0x40024400 ; 0: input | 1: output
GPIO_PORTE_AFSEL      EQU 0x40024420 ; disable AFSEL & PCTL
GPIO_PORTE_DEN        EQU 0x4002451C ; digital Enabled
GPIO_PORTE_AMSEL      EQU 0x40024528 ; analog Disabled

;LABEL    DIRECTIVE    VALUE          COMMENT
AREA      main, READONLY, CODE
THUMB
EXPORT    __main
__main
    BL     INIT_TIMER    ;initilization of SysTick
    BL     INIT_GPIO
    MOV    R5,#0 ; R5 is used to detect pressed Button
loop      B      loop

INIT_TIMER
    LDR    R0,=STCTRL
    MOV    R1,#0
    STR    R1,[R0] ; stop SysTick counter
    LDR    R1,=39999; 10msec'4 MHz; | 27100    for 16 MHz| trigger duration
    STR    R1,[R0,#4] ; set STRELOAD
    STR    R1,[R0,#8] ; clear COUNT
    MOV    R1,#0x3      ; 0011 PIOSC(16/4 MHz) | interrupt Enabled
    STR    R1,[R0]
    BX     LR

INIT_GPIO
    LDR    R1,=SYSCTL_RCGCGPIO
    LDR    R0,[R1]
    ORR    R0,R0, #0x12 ;0001 0010 ;Enable B and D port's clock
    STR    R0,[R1]
    NOP
    NOP
    NOP
;set direction register
    LDR    R1,=GPIO_PORTB_DIR
    LDR    R0,[R1]
    ORR    R0,R0, #0xF0 ;1111 0000 ;b4,b5,b6,b7 output
    STR    R0,[R1]
; disable alternative functions
    LDR    R1,=GPIO_PORTB_AFSEL
    MOV    R0,#0
    STR    R0,[R1]
; disable Analog & Enable Digital

```

```
LDR R1,=GPIO_PORTB_AMSEL
STR R0,[R1] ; disabled
LDR R1,=GPIO_PORTB_DEN
MOV R0,#0xFF
STR R0,[R1] ; enabled
; load default value
LDR R1,=GPIO_PORTB_DATA
LDR R0,[R1]
ORR R0,#0x80
STR R0,[R1]

;SET PORT E FOR INPUT BUTTONS
LDR R1,=GPIO_PORTE_DIR
MOV R0,#0; e0,e1,e2,e3 input
STR R0,[R1]
LDR R1,=GPIO_PORTE_AFSEL
STR R0,[R1]
LDR R1,=GPIO_PORTE_AMSEL
STR R0,[R1] ; disabled
LDR R1,=GPIO_PORTE_DEN
MOV R0,#0xFF
STR R0,[R1] ; enabled
BX LR

ALIGN
END
```

```

;*****
; Exp5_p3.s  Part-III
; Subroutine for ISR  e5p3_ST_ISR
; R5 should not be changed!  R3 is the input value from the button
;*****
GPIO_PORTE_DATA EQU 0x400243FC
;LABEL      DIRECTIVE  VALUE      COMMENT
          AREA          routines, READONLY, CODE
          THUMB
          EXTERN        FSM
          EXPORT        e5p3_ST_ISR

e5p3_ST_ISR
    PUSH{R1,R2,R3,LR}
    MOV R2,#0x03 ; 0000 0011  no pushed button
    MOV R3,#0    ; temp
    LDR R1,=GPIO_PORTE_DATA
    LDR R1,[R1]
    BFC R1,#2,#30 ;R1 current button status
    CMP R1,R2
    BEQ done
    LSR R2,#1    ; 0001
    CMP R1,R2    ; D1 check
    MOV R3,#0xF0 ; means D1,  CW
    BEQ done
    MOV R3,#0x0F ; means D0,  CCW
    ; R3  0: no change
    ;    F0: CW
    ;    0F: CCW
done    CMP R5,R3
    BEQ exit    ; if no change
    CMP R3,R2
    BHI Updt    ; R3 non-zero
    MOV R3,R5    ; get rotate direction
    MOV R5,#0    ; reset R0
    BL          FSM    ; do operation
    B          exit
Updt    MOV R5,R3    ; update R0
exit    POP{R1,R2,R3,LR}
    BX          LR

ALIGN
END

```

```

;*****
; Experiment 5 ; Preliminary Work Part 5;
; Step motor Speed and Direction control by 4 buttons
;*****
; INPUT : 2 switch button connected to D0 CCW and D1 CW
;STCTRL EQU 0xE000E010 ; SysTick | +4 STRELOAD | +8 STCURRENT

```

```

SYSCTL_RCGCGPIO EQU 0x400FE608 ; clock
GPIO_PORTB_DIR EQU 0x40005400 ; 0: input | 1: output
GPIO_PORTB_AFSEL EQU 0x40005420 ; disable AFSEL & PCTL
GPIO_PORTB_DEN EQU 0x4000551C ; digital Enabled
GPIO_PORTB_AMSEL EQU 0x40005528 ; analog Disabled
GPIO_PORTB_DATA EQU 0x400053FC
GPIO_PORTE_DIR EQU 0x40024400 ; 0: input | 1: output
GPIO_PORTE_AFSEL EQU 0x40024420 ; disable AFSEL & PCTL
GPIO_PORTE_DEN EQU 0x4002451C ; digital Enabled
GPIO_PORTE_AMSEL EQU 0x40024528 ; analog Disabled
GPIO_PORTE_IS EQU 0x40024404 ; Interrupt Sense
GPIO_PORTE_IBE EQU 0x40024408 ; Interrupt Both Edge
GPIO_PORTE_IEV EQU 0x4002440C ; Interrupt Event
GPIO_PORTE_IM EQU 0x40024410 ; Interrupt Mask
GPIO_PORTE_RIS EQU 0x40024414 ; Raw Interrupt Status
GPIO_PORTE_ICR EQU 0x4002441C ; Interrupt Clear Register

```

```

;LABEL DIRECTIVE VALUE COMMENT

```

```

AREA main, READONLY, CODE

```

```

THUMB

```

```

EXTERN e5p5_GPIO_ISR

```

```

EXPORT __main

```

```

__main

```

```

CPSID I

```

```

BL INIT_GPIO

```

```

MOV R6,#0xF0 ; clockwise

```

```

;MOV32 R7,#39999999 ; 1 second = 4x10^6 x 1/4x10^6

```

```

LDR R7,=#39999 ; 1 second = 4x10^6 x 1/4x10^6

```

```

NOP

```

```

CPSIE I

```

```

loop BL e5p5_GPIO_ISR

```

```

B loop

```

```

INIT_GPIO

```

```

LDR R1,=SYSCTL_RCGCGPIO

```

```

LDR R0,[R1]

```

```

ORR R0,R0, #0x12 ;0001 0010 ;Enable B and E port's clock

```

```

STR R0,[R1]

```

```

NOP

```

```

NOP

```

```

NOP

```

```

;set direction register

```

```

LDR R1,=GPIO_PORTB_DIR

```

```

LDR R0,[R1]

```

```

ORR R0,R0, #0xF0 ;1111 0000 ;b4,b5,b6,b7 output

```

```

STR R0,[R1]

```

```

; disable alternative functions

```

```

LDR R1,=GPIO_PORTB_AFSEL

```



```

MOV R0,#0
STR R0,[R1]
; disable Analog & Enable Digital
LDR R1,=GPIO_PORTB_AMSEL
STR R0,[R1] ; disabled
LDR R1,=GPIO_PORTB_DEN
MOV R0,#0xFF
STR R0,[R1] ; enabled
; load default value
LDR R1,=GPIO_PORTB_DATA
LDR R0,[R1]
ORR R0,#0x80
STR R0,[R1]

;SET PORT E FOR INPUT BUTTONS and INTERRUPTS
LDR R1,=GPIO_PORTE_DIR
MOV R0,#0xf0; e0,e1,e2,e3 input
STR R0,[R1]
LDR R1,=GPIO_PORTE_AFSEL
STR R0,[R1]
LDR R1,=GPIO_PORTE_AMSEL
STR R0,[R1] ; disabled
LDR R1,=GPIO_PORTE_DEN
MOV R0,#0x0F
STR R0,[R1] ; enabled

; SET INTERRUPT
;LDR R1,=GPIO_PORTE_IM
;MOV R0,#0
;STR R0,[R1] ; disable interrupt before configuration
;LDR R1,=GPIO_PORTE_IS
;STR R0,[R1]; Edge sensitive
;LDR R1,=GPIO_PORTE_IBE
;MOV R0,#0xFF
;STR R0,[R1] ;Both edge
;; Clear GPIORIS register by using GPIOICR
;LDR R1,=GPIO_PORTE_ICR
;STR R0,[R1] ; clear interrupt flags
;LDR R1,=GPIO_PORTE_IM ; Unmask
;MOV R0,#0x0F
;STR R0,[R1] ; E0,E1,E2,E3 Interrupt Activated
BX LR

ALIGN
END

```

```

;*****
; e5p5_GPIO_ISR.s  Part-V
; Subroutine for ISR  e5p5_ST_ISR
;
; R6 : Direction  E1: ClockWise  E0: CounterClockWise
; R7 : Speed      E3: Speed UP   E2: Speed DOWN
; R3 is the input value to the FSM Driver
;*****
R3ADRES EQU 0x20004000
STCTRL EQU 0xE000E010 ; SysTick | +4 STRELOAD | +8 STCURRENT
GPIO_PORTE_DATA EQU 0x400243FC
;LABEL DIRECTIVE VALUE COMMENT
AREA routines, READONLY, CODE
THUMB
EXTERN FSM
EXPORT e5p5_GPIO_ISR

e5p5_GPIO_ISR
    PUSH{R0,R1,R2,LR}
    MOV32 R0,#0xFFFF ; ~0.5 milisecond
delay SUBS R0,#1 ; debouncing
    BNE delay

    LDR R1,=GPIO_PORTE_DATA
    LDR R0,[R1]
    BFC R0,#4,#28
    MOV R2,#0x0F
    CMP R0,R2
    BEQ Apply
    MOV R2,#0x07; 0111 = 7
    CMP R0,R2
    BEQ sUP
    MOV R2,#0x0B; 1011 = 11
    CMP R0,R2
    BEQ sDOWN
    MOV R2,#0x0D; 1101 = 13
    CMP R0,R2
    BEQ CW
    MOV R2,#0x0E; 1110 = 14
    CMP R0,R2
    BEQ CCW
    B exit

sUP MOV R0,#2
    UDIV R7,R0
    ;LSR R7,#1
    B exit
sDOWN MOV R0,#2
    MUL R7,R0
    ;LSL R7,#1
    B exit
CW MOV R6,#0xF0
    B exit
CCW MOV R6,#0x0F
    B exit

Apply MOV R3,R6;#0xF0 ; clockwise
    LDR R0,=R3ADRES

```

```
STR R3,[R0]
MOV R4,R7;#3999999 ; 1 second = 4x10^6 x 1/4x10^6
BL INIT_TIMER
```

```
exit POP{R0,R1,R2,LR}
BX LR
```

```
INIT_TIMER ; Speed of the Stepper Motor
LDR R0,=STCTRL
MOV R1,#0
STR R1,[R0] ; stop SysTick counter
MOV R1,R4 ; Update Rotation speed
STR R1,[R0,#4] ; set STRELOAD
STR R1,[R0,#8] ; clear COUNT
MOV R1,#0x3 ; 0011 PIOSC(16/4 MHz) | interrupt Enabled
STR R1,[R0]
BX LR

ALIGN
END
```