Preliminary work

EE 447: Lab #3

Introduction to Interrupts through Stepper Motor Driving

Berkay İPEK

2304814 – Sec.2

Question 1)

D:\OKUL\ee 4ün 1i\Lab-447\LabFour\QuestionOne\portb_init.s

```
1   GPIO_PORTB_DIR      EQU 0x40005400
2   GPIO_PORTB_AFSEL    EQU 0x40005420
3   GPIO_PORTB_DEN      EQU 0x4000551C
4   SYSCTL_RCGCGPIO     EQU 0x400FE608
5
6
7                       AREA        main, READONLY, CODE
8                       THUMB
9                       EXPORT      portb_init
10
11  portb_init          PROC;
12                      PUSH        {R0,R1}
13  Start               LDR         R1,=SYSCTL_RCGCGPIO
14                      LDR         R0,[R1]
15                      ORR         R0,R0,#0x02
16                      STR         R0,[R1]
17                      NOP
18                      NOP
19                      NOP
20                      LDR         R1,=GPIO_PORTB_DIR
21                      LDR         R0,[R1]
22                      MOV         R0,#0x0F
23                      STR         R0,[R1]
24                      LDR         R1,=GPIO_PORTB_AFSEL
25                      LDR         R0,[R1]
26                      BIC         R0,#0xFF
27                      STR         R0,[R1]
28                      LDR         R1,=GPIO_PORTB_DEN
29                      LDR         R0,[R1]
30                      ORR         R0,#0xFF
31                      STR         R0,[R1]
32                      POP         {R0,R1}
33                      BX          LR
```

Port B Configuration for step motor

It should be noted that it is assumed PB0-3 is connected to IN1-4 pins in step motor. Also, I connected VBUS to 5V in step motor (GND to – side in step motor)

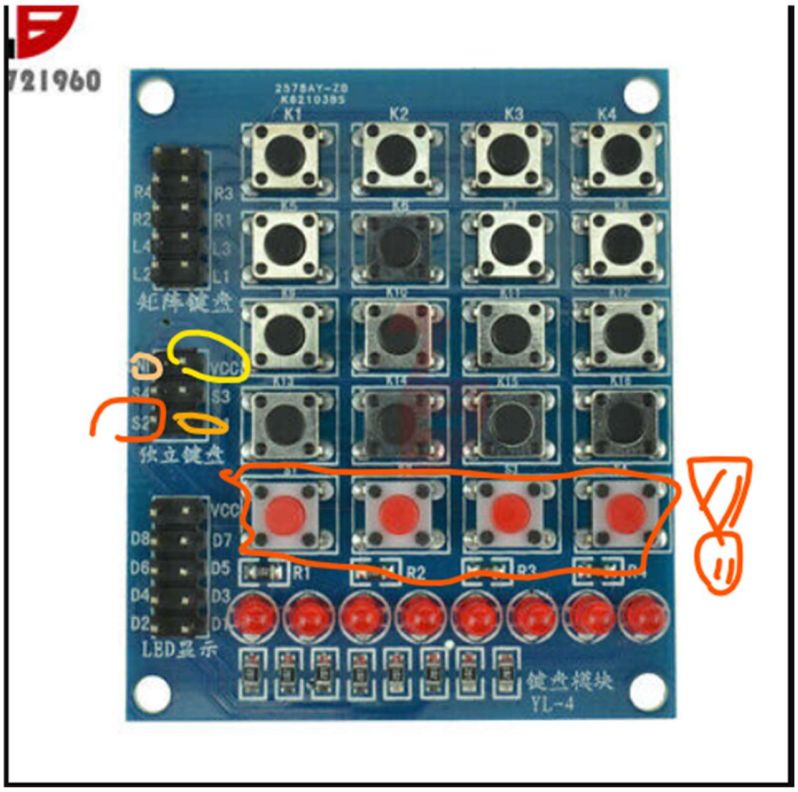
Question 2)

GND

3.3V

P64

P65



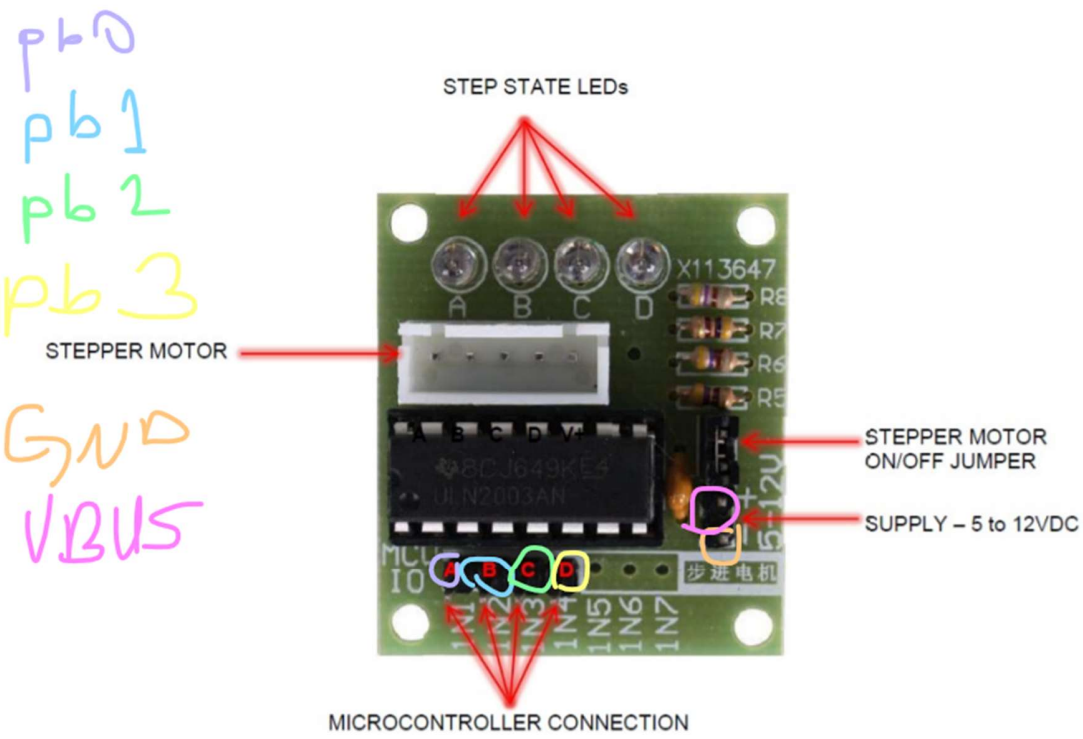
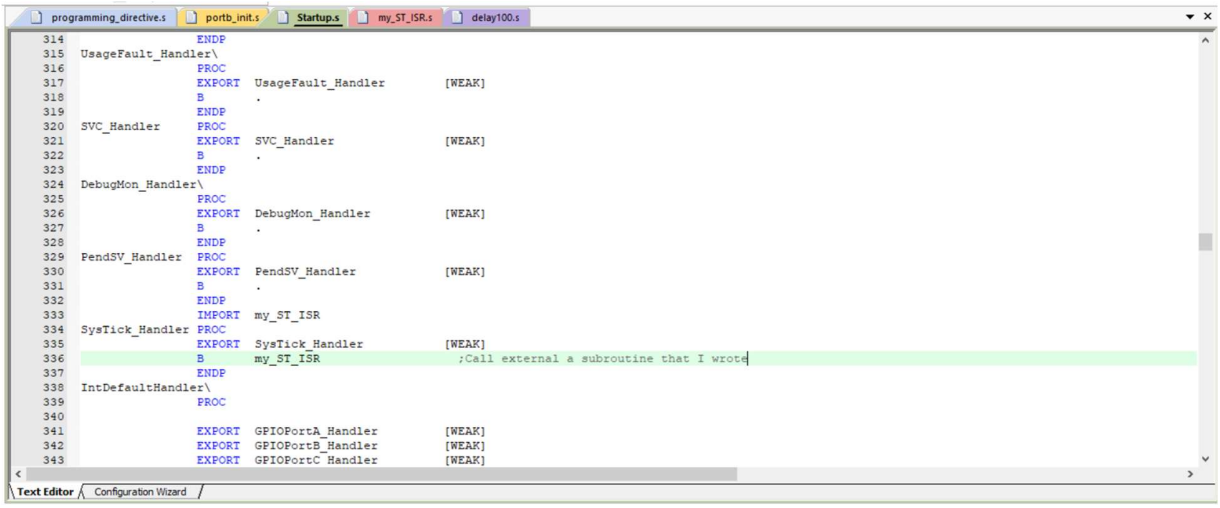


Figure 6: ULN2003A PCB connections

Connections are given in figures. In these figures, connections are given in different colors so that it can be easily follow. In this report, SW1-4 will be used. Outputs of Stepper Motor are connected to the Stepper Motor Port of ULN2003A.

Question 3)

In the startup.s file, there is a change in SysTick_Handler part. It is calling external subroutine my_ST_ISR, that I wrote.



```
1   PB_INP              EQU 0x400050C0
2   SYSCTRL             EQU 0xE000E010
3                       AREA        main, READONLY, CODE
4                       THUMB
5                       EXTERN      DELAY100        ;Delay Subroutine from previous experiment (For
    debouncing)
6                       EXTERN      portb_init      ;PortB Initialize
7                       EXPORT      __main
8
9   __main              PROC;
10
11      ;Rotation Type will be determined by R10 (1: Clockwise Rotation(Default), 2: Counter Clockwise
    Rotation)
12                      MOV         R10,#0x01
13      ;R9 begins with 1. Look in my_ST_ISR.file
14                      MOV         R9,#0x01
15                      BL          portb_init
16      ;SysTimer Settings
17                      LDR         R0,=SYSCTRL     ;set the address of systemctrl
18                      MOV         R1,#0
19                      STR         R1,[R0]         ;Reseting
20                      MOV         R1,#9000        ;GIVEN R8 VALUE, ROTATION(Number of Cycle) SPEED CAN
    BE ADJUSTED
21                      STR         R1,[R0,#4]      ;Reload value
22                      STR         R1,[R0,#8]      ;Current value
23                      MOV         R1,#0x03        ;(enable, interrupt, use PIOSC as clock)
24                      STR         R1,[R0]         ;Start timer
25
26                      LDR         R0,=PB_INP
27  re                  LDR         R1,[R0]
28                      CMP         R1,#0x30        ;
29                      BEQ         re              ;If there is no pushed switch
30                      BL          DELAY100        ; To put a barrier for debouncing
31                      LDR         R2,[R0]
32                      CMP         R1,R2
33                      BNE         re
34
35                      CMP         R1,#0x30        ;If there is no pushed switched
36                      BEQ         re
37
38                      CMP         R1,#0x20        ;If SW1 is pressed, Rotation will be in
    CounterClockWise
39                      BEQ         ccw
40
41                      CMP         R1,#0x10        ;If SW2 is pressed, Rotation will be in ClockWise
42                      BEQ         cw
43                      B           re
44
45  ccw                 LDR         R2,[R0]
46                      CMP         R2,R1
47                      BEQ         ccw             ;Wait untill switch is released
48                      MOV         R10,#0x02       ;Make R10 0x02 so that it can rotate in ccw (see
    my_ST_ISR.s)
49                      B           re
50
51  cw                  LDR         R2,[R0]
52                      CMP         R2,R1
53                      BEQ         cw              ;Wait untill switch is released
54                      MOV         R10,#0x01       ;Make R10 0x02 so that it can rotate in ccw (see
    my_ST_ISR.s)
55                      B           re
56                      ALIGN
57                      ENDP
58                      END
59
```

D:\OKUL\ee 4ün 1i\Lab-447\LabFour\QuestionTwo\my_ST_ISR.s

```
PB_OUT              EQU 0x4000503C
                    AREA        main, READONLY, CODE
                    THUMB
                    EXPORT      my_ST_ISR
;This script is called when sysTimer is triggered
;According to value in R10, it will be determined which rotation will be occured,
;CounterClockWise(cCW)      ClockWise(cW)
;R10 value is changing in main function
my_ST_ISR           PROC
                    LDR         R5,=PB_OUT
                    STR         R9,[R5]     ;Write values in IN1-4
                    CMP         R10,#0x02   ;Deciding which rotation
                    BEQ         cCW
                    CMP         R10,#0x01
                    BEQ         cW
                    BX          LR

cW                  LSR         R9,#1       ;Out1-4 changing as desired in manual
                    CMP         R9,#0x00    ;Checking boundaries
                    MOVEQ       R9,#0x08
                    BX          LR

cCW                 LSL         R9,#1       ;Out1-4 changing as desired in manual
                    CMP         R9,#0x10    ;Checking boundaries
                    MOVEQ       R9,#0x01
                    BX          LR

                    ALIGN
                    ENDP
                    END
```

D:\OKUL\ee 4ün 1i\Lab-447\LabFour\QuestionTwo\portb_init.s

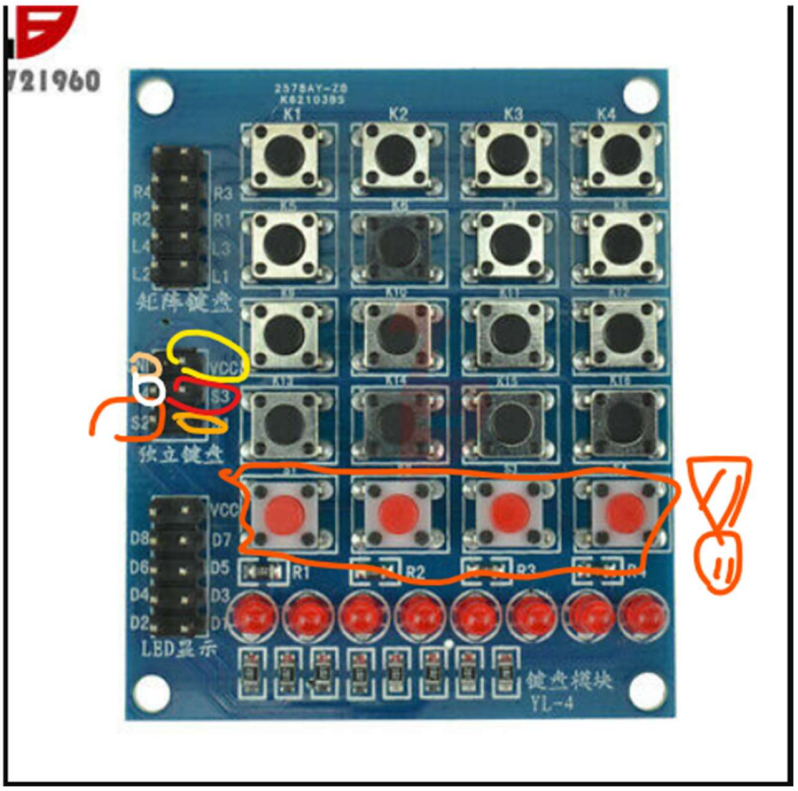
```
GPIO_PORTB_DIR      EQU 0x40005400
GPIO_PORTB_AFSEL    EQU 0x40005420
GPIO_PORTB_DEN      EQU 0x4000551C
SYSCTL_RCGCGPIO     EQU 0x400FE608

;PORTS ARE CONNECTED AS
;IN1 =>pb0 , IN2 =>pb1 , IN3 =>pb2 , IN4 =>pb3
;S1 => pb4, S2=> pb5
                    AREA        main, READONLY, CODE
                    THUMB
                    EXPORT      portb_init

portb_init          PROC;
                    PUSH        {R0,R1}
Start               LDR         R1,=SYSCTL_RCGCGPIO
                    LDR         R0,[R1]
                    ORR         R0,R0,#0x2F
                    STR         R0,[R1]
                    NOP
                    NOP
                    NOP
                    LDR         R1,=GPIO_PORTB_DIR
                    LDR         R0,[R1]
                    MOV         R0,#0x0F
                    STR         R0,[R1]
                    LDR         R1,=GPIO_PORTB_AFSEL
                    LDR         R0,[R1]
                    BIC         R0,#0xFF
                    STR         R0,[R1]
                    LDR         R1,=GPIO_PORTB_DEN
                    LDR         R0,[R1]
                    ORR         R0,#0xFF
                    STR         R0,[R1]
                    POP         {R0,R1}
                    BX          LR
```

D:\OKUL\ee 4ün 1i\Lab-447\LabFour\QuestionTwo\delay100.s

```
1
2                AREA        main, READONLY, CODE
3                THUMB
4                EXPORT      DELAY100
5
6
7
8   DELAY100     PROC;
9                PUSH        {R0}
10               MOV32       R0,#400000
11  berkay       SUBS        R0,#1
12               NOP
13               BNE         berkay
14               POP         {R0}
15               BX          LR
16               ENDP
17               END
```

Question 4)

GND
3.3V
P b 4
P b 5
P b b



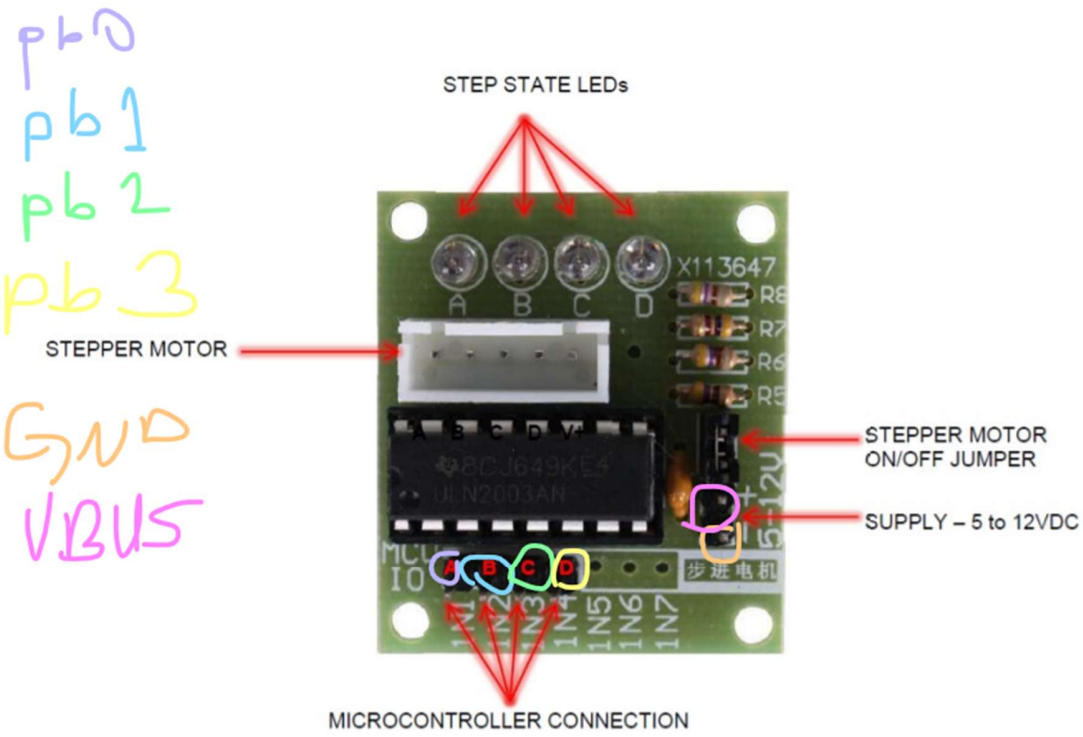


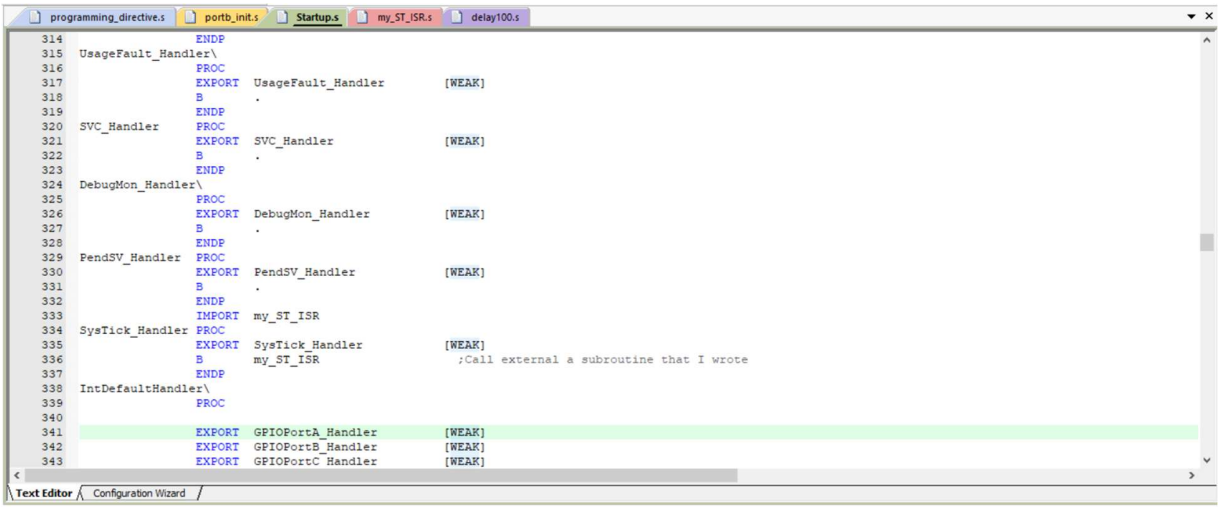
Figure 6: ULN2003A PCB connections

Connections are given in figures. In these figures, connections are given in different colors so that it can be easily follow. In this report, SW1-4 will be used. Outputs of Stepper Motor are connected to the Stepper Motor Port of ULN2003A. Also, it should be noted that there is only one change in connections. That is pb6 and pb7 are connected to SW3 and SW4.

Question 5)

Delaying subroutine, DELAY100, is the same in question 3.

In the startup.s file, there is a change in SysTick_Handler part. It is calling external subroutine my_ST_ISR, that I wrote.



D:\OKUL\ee 4ün 1i\Lab-447\LabFour\QuestionFive\programming_directive.s

```
PB_INP              EQU 0x400053C0          ;To take input

                    AREA        main, READONLY, CODE
                    THUMB
                    EXTERN      DELAY100        ;Delay for Buttons
                    EXTERN      portb_init      ;PortB Initialize
                    EXTERN      IntStart        ;Interrupt settings (Given value in the R8)
                    EXPORT      __main

__main              PROC

    ;Rotation Type will be determined by R10 (1: Clockwise Rotation(Default), 2: Counter Clockwise Rotation)
                    MOV         R10,#0x01
    ;R9 begins with 1. Look in my_ST_ISR.file
                    MOV         R9,#0x01
                    BL          portb_init    ;PORT_B initializer
                    MOV         R8,#10000      ;Default Rotation Speed Value
                    BL          IntStart    ;Create SysTime
re                  LDR         R0,=PB_INP  ;Taking input
                    LDR         R1,[R0]
                    CMP         R1,#0xF0
                    BEQ         re
                    BL          DELAY100    ; To put a barrier for debouncing
                    LDR         R2,[R0]
                    CMP         R1,R2
                    BNE         re          ; To put a barrier for debouncing

                    CMP         R1,#0xF0    ; F0 == No button is pressed. Keep going
                    BEQ         re

                    CMP         R1,#0xE0    ; E0 == SW1 is pressed => Rotate Counter Clockwise
                    BEQ         ccw

                    CMP         R1,#0xD0    ; D0 == SW2 is pressed => Rotate Clockwise
                    BEQ         cw

                    CMP         R1,#0xB0    ; B0 == SW3 is pressed => Rotation Speed is changed to fast
                    BEQ         speedup

                    CMP         R1,#0x70    ; 70 == SW4 is pressed => Rotation Speed is changed to slow
                    BEQ         speeddown

                    B           re          ; Other cases => dont do anything

ccw                 LDR         R2,[R0]
                    CMP         R2,R1
                    BEQ         ccw         ;Wait until key is released
                    MOV         R10,#0x02   ;Change R10 to 0x02 so that rotation can be in ccw (See my_STR_ISR.s)
                    B           re

cw                  LDR         R2,[R0]
                    CMP         R2,R1
                    BEQ         cw          ;Wait until key is released
                    MOV         R10,#0x01   ;Change R10 to 0x01 so that rotation can be in cw (See my_STR_ISR.s)
                    B           re

speedup             LDR         R2,[R0]
                    CMP         R2,R1
                    BEQ         speedup     ;Wait until key is released
                    MOV         R8,#9000    ; Set R8 to fast speed value. See InterruptStarter.s
                    BL          IntStart
                    B           re

speeddown           LDR         R2,[R0]
                    CMP         R2,R1
                    BEQ         speeddown   ;Wait until key is released
                    MOV         R8,#30000   ; Set R8 to slow speed value. See InterruptStarter.s
                    BL          IntStart
                    B           re

                    ALIGN
                    ENDP
```


NOTE: There is an END command at the end

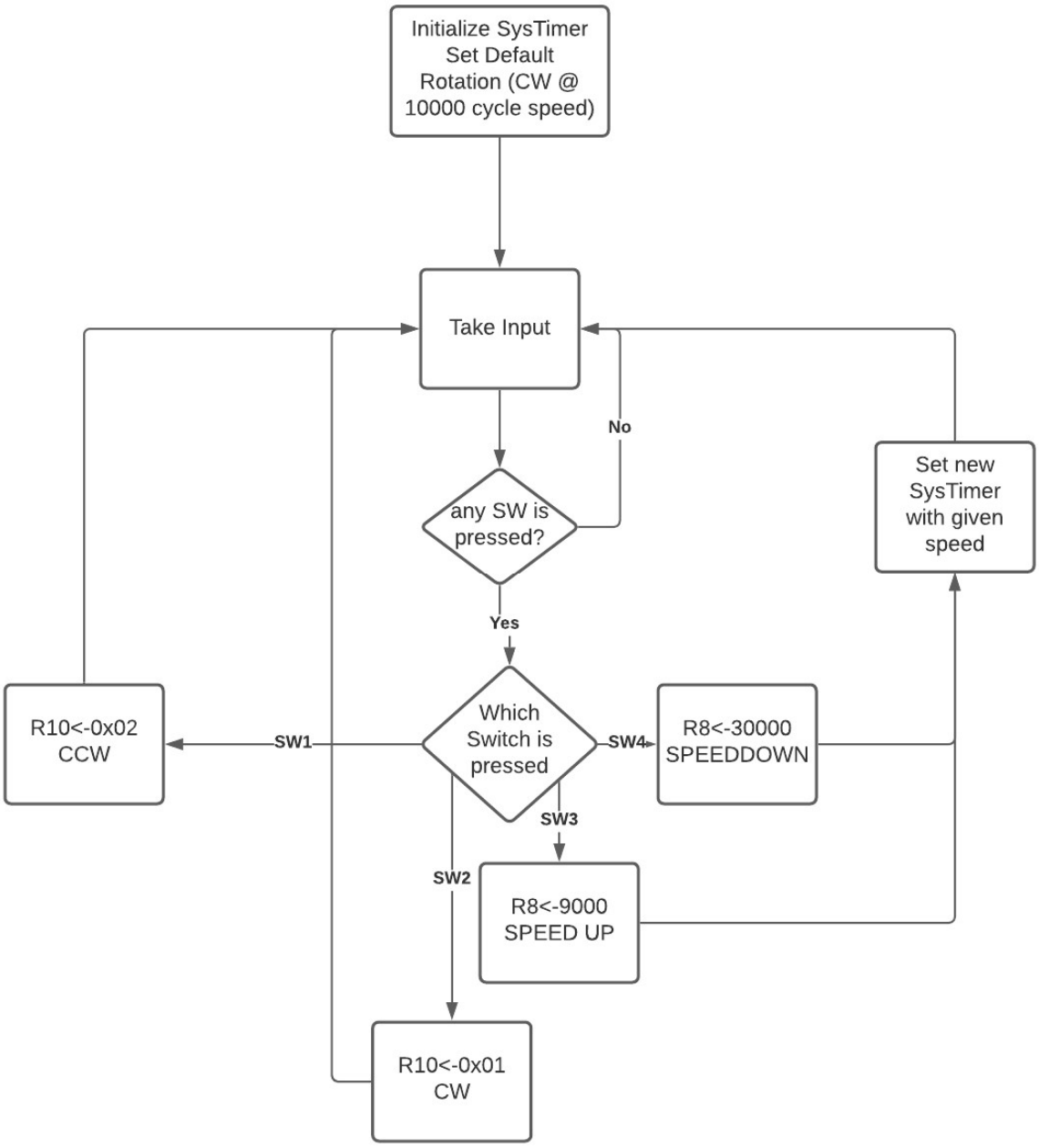
D:\OKUL\ee 4ün 1i\Lab-447\LabFour\QuestionFive\portb_init.s

```
1   GPIO_PORTB_DIR      EQU 0x40005400
2   GPIO_PORTB_AFSEL    EQU 0x40005420
3   GPIO_PORTB_DEN      EQU 0x4000551C
4   SYSCTL_RCGCGPIO     EQU 0x400FE608
5
6   ;PORTS ARE CONNECTED AS
7   ;IN1 =>pb0 , IN2=>pb1 , IN3=>pb2 , IN4=>pb3
8   ;SW1 =>pb4 , SW2=>pb5 , SW3=>pb6 , SW4=>pb7
9                       AREA        main, READONLY, CODE
10                      THUMB
11                      EXPORT      portb_init
12
13  portb_init          PROC;
14                      PUSH        {R0,R1}
15  Start               LDR         R1,=SYSCTL_RCGCGPIO
16                      LDR         R0,[R1]
17                      ORR         R0,R0,#0x2F
18                      STR         R0,[R1]
19                      NOP
20                      NOP
21                      NOP
22                      LDR         R1,=GPIO_PORTB_DIR
23                      LDR         R0,[R1]
24                      MOV         R0,#0x0F
25                      STR         R0,[R1]
26                      LDR         R1,=GPIO_PORTB_AFSEL
27                      LDR         R0,[R1]
28                      BIC         R0,#0xFF
29                      STR         R0,[R1]
30                      LDR         R1,=GPIO_PORTB_DEN
31                      LDR         R0,[R1]
32                      ORR         R0,#0xFF
33                      STR         R0,[R1]
34                      POP         {R0,R1}
35                      BX          LR
```

D:\OKUL\ee 4ün 1i\Lab-447\LabFour\QuestionFive\InterruptStarter.s

```
1   SYSCTRL             EQU         0xE000E010
2                       AREA        main, READONLY, CODE
3                       THUMB
4                       EXPORT      IntStart
5   ;This script is to create system timer with given R8 Value
6   IntStart            PROC
7                       LDR         R0,=SYSCTRL     ;set the address of systemctrl
8                       MOV         R1,#0
9                       STR         R1,[R0]         ;Reseting
10                      MOV         R1,R8           ;GIVEN R8 VALUE, ROTATION(Number of Cycle) SPEED CAN
    BE ADJUSTED
11                      STR         R1,[R0,#4]      ;Reload value
12                      STR         R1,[R0,#8]      ;Current value
13                      MOV         R1,#0x03        ;(enable, interrupt, use PIOSC as clock)
14                      STR         R1,[R0]         ;Start timer
15                      BX          LR
```

```
1   PB_OUT              EQU 0x4000503C
2                       AREA        main, READONLY, CODE
3                       THUMB
4                       EXPORT      my_ST_ISR
5   ;This script is called when sysTimer is triggered
6   ;According to value in R10, it will be determined which rotation will be occured,
7   ;CounterClockWise(cCW)      ClockWise(cW)
8   ;R10 value is changing in main function
9   my_ST_ISR           PROC
10                      LDR         R5,=PB_OUT
11                      STR         R9,[R5]     ;Write values in IN1-4
12                      CMP         R10,#0x02   ;Deciding which rotation
13                      BEQ         cCW
14                      CMP         R10,#0x01
15                      BEQ         cW
16                      BX          LR
17
18  cW                  LSR         R9,#1       ;Out1-4 changing as desired in manual
19                      CMP         R9,#0x00    ;Checking boundaries
20                      MOVEQ       R9,#0x08
21                      BX          LR
22
23  cCW                 LSL         R9,#1       ;Out1-4 changing as desired in manual
24                      CMP         R9,#0x10    ;Checking boundaries
25                      MOVEQ       R9,#0x01
26                      BX          LR
27
28                      ALIGN
29                      ENDP
30                      END
```


Initialize SysTimer
Set Default
Rotation (CW @
10000 cycle speed)
Take Input
any SW is
pressed?
No
Yes
Which
Switch is
pressed
SW1
SW2
SW3
SW4
R10<-0x02
CCW
R10<-0x01
CW
R8<-9000
SPEED UP
R8<-30000
SPEEDDOWN
Set new
SysTimer
with given
speed

Flow Chart of ISR Subroutine