Furkan Karakaya - 1937051 Experiment 4 Pre Work 01.12.2016

1) Delay Subroutine

```
;***************************************************************
; Program section
;***************************************************************
;LABEL           DIRECTIVE       VALUE           COMMENT
;LABEL DIRECTIVE VALUE COMMENT
                                 AREA rutins , CODE, READONLY
                                 THUMB
                                 EXPORT delay ;
delay            PROC

GoBack           SUBS    R0,R0,#1;
                                 BEQ             End_Delay
                                 B               GoBack
End_Delay        BX LR; end

;***************************************************************
; End of the program  section
;***************************************************************
;LABEL     DIRECTIVE      VALUE                  COMMENT
                                 ALIGN
                                 ENDP
                                 END
```

2) Simple Data Transfer

```
;***************************************************************
; Program_Directives.s
; Copies the table from one location
; to another memory location.
; Directives and Addressing modes are
; explained with this program.
;***************************************************************
;***************************************************************
; EQU Directives
; These directives do not allocate memory
;***************************************************************
;LABEL           DIRECTIVE       VALUE           COMMENT
OFFSET           EQU     0x10
FIRST            EQU             0x20000480
STORE            EQU                     0x20000410
GPIO_PORTB_DATA          EQU 0x400053FC
GPIO_PORTB_DIR                   EQU 0x40005400
GPIO_PORTB_AFSEL         EQU 0x40005420
GPIO_PORTB_DEN                   EQU 0x4000551C
GPIO_PORTB_PUR                   EQU 0x40005510
GPIO_PORTB_PDR                   EQU 0x40005514
IOB                              EQU 0x0F
```

```
SYSCTL_RCGCGPIO      EQU 0x400FE608

;************************************************************
; Directives - This Data Section is part of the code
; It is in the read only section  so values cannot be changed.
;************************************************************
;LABEL           DIRECTIVE       VALUE           COMMENT
        AREA     sdata, DATA, READONLY
        THUMB
CTR1    DCB     0x10
MSG     DCB     "Copying table..."
                        DCB                     0x0D
                        DCB                     0x04
;************************************************************
; Program section
;************************************************************
;LABEL           DIRECTIVE       VALUE           COMMENT
                        AREA    main, READONLY, CODE
                        THUMB
                        EXTERN          OutStr  ; Reference external subroutine
                        EXTERN          InChar; Serial input Added
                        EXTERN          delay;
                        EXPORT          __main ; Make available

__main
Start           LDR R1 , =SYSCTL_RCGCGPIO
                        LDR R0 , [ R1 ]
                        ORR R0 , R0 , #0x2;Port B clock enabled
                        STR R0 , [ R1 ]
                        NOP             ;Wait for clock to stabilize
                        NOP
                        NOP
                        LDR R1 , =GPIO_PORTB_DIR ; c o n f i g . o f p o r t B s t a r t s
                        LDR R0 , [ R1 ]
                        BIC R0 , #0xFF
                        ORR R0 , #IOB;00001111 1->output
                        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 ]
                        LDR R1 , =GPIO_PORTB_PUR
                        LDR R0 , [ R1 ]
                        ORR R0 , #0xF0
```

```
                    STR R0 , [ R1 ]

Begin        LDR    R1,=GPIO_PORTB_DATA; Data address in R1
                    MOV    R0,#0xFF
                    STR    R0,[R1];        All outputs OFF

InputCheck   MOV           R2,#15
                    LDR    R0,[R1];
                    LSR           R0,#4;
                    LSRS   R0,#1;
                    ANDCC R2,#0xFE;
                    LSRS   R0,#1;
                    ANDCC R2,#0xFD;
                    LSRS   R0,#1;
                    ANDCC R2,#0xFB;
                    LSRS   R0,#1;
                    ANDCC R2,#0xF7;
                    CMP           R2,#15
                    BEQ           InputCheck
                    MOV32         R0,#1600000; 100msec delay
                    BL            delay
                    MOV           R4,#15
                    LDR    R0,[R1];
                    LSR           R0,#4;
                    LSRS   R0,#1;
                    ANDCC R4,#0xFE;
                    LSRS   R0,#1;
                    ANDCC R4,#0xFD;
                    LSRS   R0,#1;
                    ANDCC R4,#0xFB;
                    LSRS   R0,#1;
                    ANDCC R4,#0xF7;
                    CMP           R2,R4 ; IF they are equal set the output
                    BNE           InputCheck
                                                     ; If an input is read
                    LDR           R1,=GPIO_PORTB_DATA; Data address in R1
                    STR           R4,[R1];        Corresponding Outputs set high
                    MOV32 R0,#25400000 ; 7Sec
                    BL            delay
                    B             Begin

;*************************************************************
; End of the program  section
;*************************************************************
;LABEL     DIRECTIVE     VALUE               COMMENT
                    ALIGN
                    END
```

3) Keypad Interface

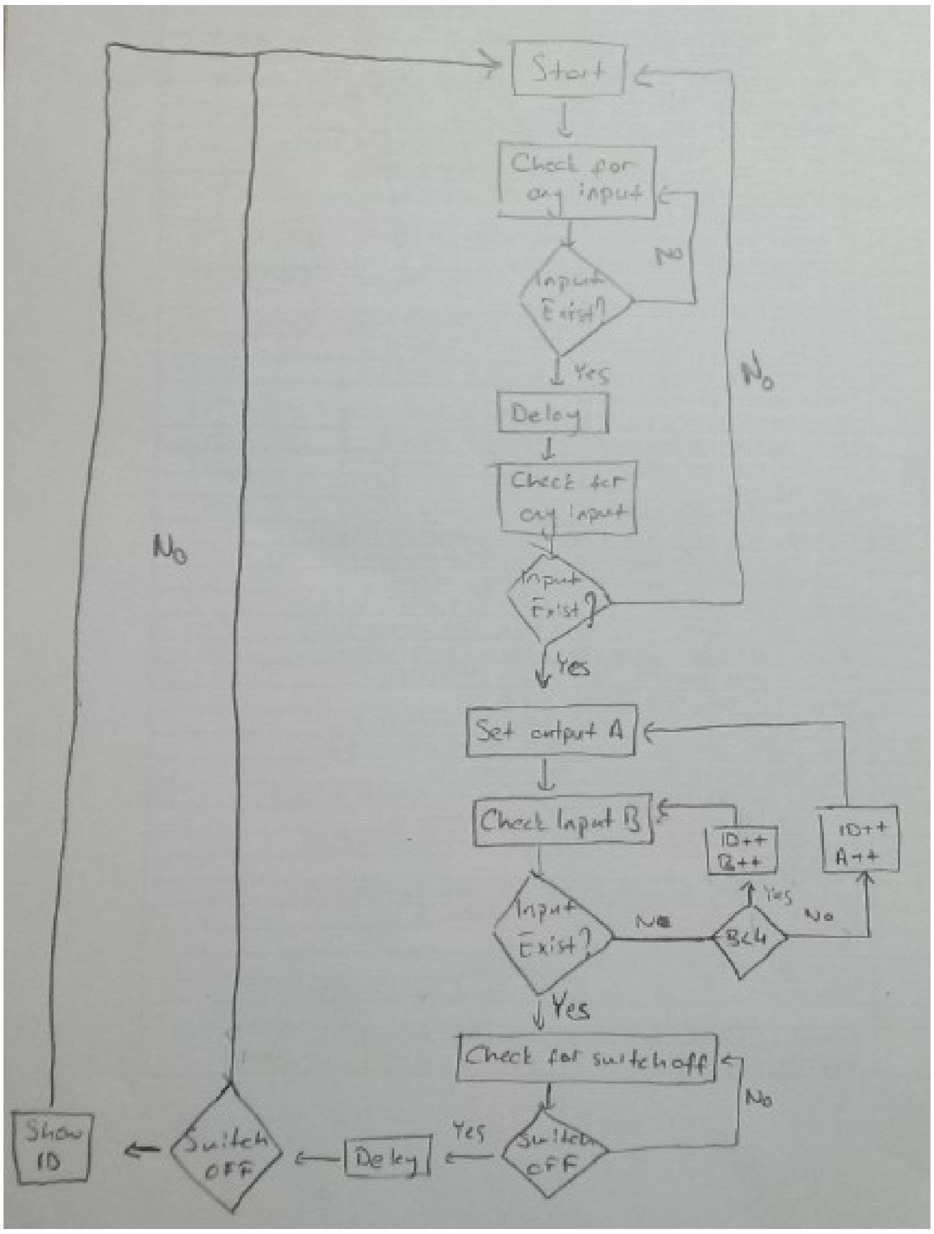
i. The input pins (4-7) should be checked continuously if any one reads an input, delay subroutine should be called and again the inputs should be checked. If an input is read again, this means one of the switches is pressed.

ii. In order to detect the switch pressed, we should set the output pins one by one. After setting one of them LOW, we should check if any one of the input pins read anything. If so that means we found the switch, else we should change the output settings by setting next output pin to LOW.

iii. While looking for the switches in part ii, a counter is increased if the switch is not detected. The counter shows the ID of the switches, so if the switch is detected, the counter shows the switch ID.

iv. Bouncing can manipulate our detectors; therefore, we use delay, that is for any change in the inputs (high->low or low->high) we should wait for a while and read the inputs again to see whether a real input has come. If we do not apply this, the bouncing effect can cheat us.

v. Flow Chart:



```
;***************************************************************
; Program_Directives.s
; Copies the table from one location
; to another memory location.
; Directives and Addressing modes are
; explained with this program.
;***************************************************************
;***************************************************************
; EQU Directives
; These directives do not allocate memory
;***************************************************************
;LABEL            DIRECTIVE       VALUE           COMMENT
OFFSET            EQU     0x10
FIRST             EQU             0x20000480
STORE             EQU                     0x20000410
GPIO_PORTB_DATA   EQU 0x400053FC
GPIO_PORTB_DIR                    EQU 0x40005400
GPIO_PORTB_AFSEL  EQU 0x40005420
GPIO_PORTB_DEN                    EQU 0x4000551C
GPIO_PORTB_PUR                    EQU 0x40005510
GPIO_PORTB_PDR                    EQU 0x40005514
IOB                               EQU 0x0F
SYSCTL_RCGCGPIO   EQU 0x400FE608

;***************************************************************
; Directives - This Data Section is part of the code
; It is in the read only section  so values cannot be changed.
;***************************************************************
;LABEL            DIRECTIVE       VALUE           COMMENT
        AREA      sdata, DATA, READONLY
        THUMB
CTR1    DCB     0x10
MSG     DCB     "Copying table..."
                  DCB                     0x0D
                  DCB                     0x04
;***************************************************************
; Program section
;***************************************************************
;LABEL            DIRECTIVE       VALUE           COMMENT
                  AREA    main, READONLY, CODE
                  THUMB
                  EXTERN          OutStr  ; Reference external subroutine
                  EXTERN          InChar; Serial input Added
                  EXTERN          delay;
                  EXPORT          __main ; Make available

__main
```

```
Start           LDR R1 , =SYSCTL_RCGCGPIO
                      LDR R0 , [ R1 ]
                      ORR R0 , R0 , #0x2;Port B clock enabled
                      STR R0 , [ R1 ]
                      NOP              ;Wait for clock to stabilize
                      NOP
                      NOP
                      LDR R1 , =GPIO_PORTB_DIR ;
                      LDR R0 , [ R1 ]
                      BIC R0 , #0xFF
                      ORR R0 , #IOB;00001111 1->output
                      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 ]
                      LDR R1 , =GPIO_PORTB_PUR
                      LDR R0 , [ R1 ]
                      ORR R0 , #0xF0
                      STR R0 , [ R1 ]

Begin           LDR   R1,=GPIO_PORTB_DATA; Data address in R1
                      MOV   R0,#0x00
                      STR   R0,[R1];        All outputs GND
                      MOV   R2,#0;          R2 is the switch ID
InputCheck      LDR   R0,[R1];Checks for any input
                      LSR          R0,#4;
                      LSRS  R0,#1;
                      BCC          Delay100
                      LSRS  R0,#1;
                      BCC          Delay100
                      LSRS  R0,#1;
                      BCC          Delay100
                      LSRS  R0,#1;
                      BCC          Delay100
                      B            InputCheck
Delay100        MOV32 R0,#1600000;If any input is detected
                      BL           delay
                      LDR   R0,[R1];        Check Again
                      LSR          R0,#4;
                      LSRS  R0,#1;
                      BCC          Detect ; If input is detected again go Detect
                      LSRS  R0,#1;
                      BCC          Detect
```

```
                        LSRS    R0,#1;
                        BCC             Detect
                        LSRS    R0,#1;
                        BCC             Detect
                        B               InputCheck
Detect          LDR     R1,=GPIO_PORTB_DATA; Lets check inputs for different outputs
                        MOV     R0,#0x0E        ;output 1110
                        STR     R0,[R1];
                        MOV32   R0,#1600000;
                        BL              delay
                        BL Which        ;Which decides which input is reading
                        MOV     R0,#0x0D        ;output 1101
                        STR     R0,[R1];
                        MOV32   R0,#1600000;
                        BL              delay
                        BL Which
                        MOV     R0,#0x0B        ;output 1011
                        STR     R0,[R1];
                        MOV32   R0,#1600000;
                        BL              delay
                        BL Which
                        MOV     R0,#0x07        ;output 0111
                        STR     R0,[R1];
                        MOV32   R0,#1600000;
                        BL              delay
                        BL Which
                        B Begin
Which           LDR     R0,[R1];
                        LSR             R0,#4;
                        LSRS    R0,#1;
                        BCC             NextStep;       If Carry is zero go NextStep
                        ADD             R2,R2,#1;
                        LSRS    R0,#1;
                        BCC             NextStep;
                        ADD             R2,R2,#1;
                        LSRS    R0,#1;
                        BCC             NextStep;
                        ADD             R2,R2,#1;
                        LSRS    R0,#1;
                        BCC             NextStep;
                        ADD             R2,R2,#1;
                        BX              LR


NextStep        LDR     R0,[R1];        It checks for if the switch is open again
                        LSR             R0,#4;
                        LSRS    R0,#1;
                        BCC             NextStep;       If it is not open
```

```
                LSRS    R0,#1;
                BCC             NextStep;
                LSRS    R0,#1;
                BCC             NextStep;
                LSRS    R0,#1;
                BCC             NextStep;
                ;If all inputs show 1 that is no input comes
                MOV32 R0,#1600000;
                BL              delay
                LDR     R0,[R1];        Checks again to be sure
                LSR             R0,#4;
                LSRS    R0,#1;
                BCC             Begin;          If any input is read again
                LSRS    R0,#1;
                BCC             Begin;
                LSRS    R0,#1;
                BCC             Begin;
                LSRS    R0,#1;
                BCC             Begin;
                ;If no input is detected then in R2 we have the switch ID
                LDR R1,=FIRST
                CMP     R2,#10
                ADDLT R2,R2,#0x30; ASCII code modified
                ADDGE R2,R2,#0x37;
                STR R2,[R1];    R2 is stored to where R0 points
                MOV R5,R1;              OutStr modification
                ADD     R1,R1,#1;
                MOV R2,#0x04;
                STR     R2,[R1];        End setup for OutStr
                BL      OutStr
                B       Begin;          Go back

;***************************************************************
; End of the program  section
;***************************************************************
;LABEL      DIRECTIVE       VALUE                   COMMENT
                ALIGN
                END
```