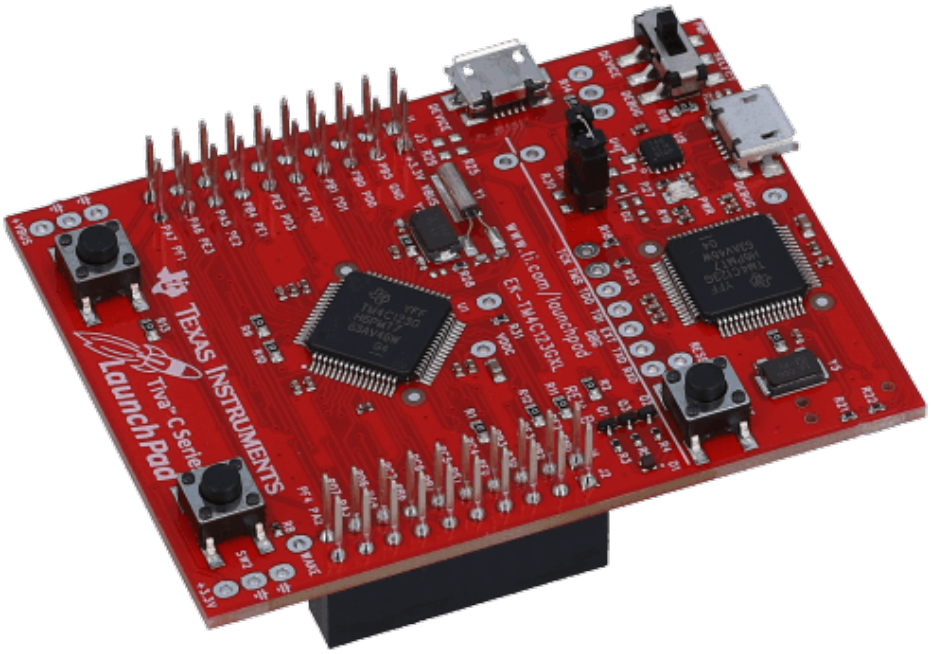
ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

EE447- Introduction to Microprocessors Laboratory With Assembly Programming

(Preliminary Work 2)

EXPERIMENTAL WORK NO: 2

1st Group Member: Uğur SAMANCI - 2398915
2nd Group Member: Barış GÜZEL - 2304764



Question 1) DELAY150 subroutine

- In this part of the preliminary work, it is asked to write a subroutine that add 150msec delay when called. Below you can find the time pass of every different line

$$BL \ = 0.250 \ \mu sec$$

$$PUSH \ \{LR, RO\} = 0.250 \ \mu sec$$

$$LDR \ R1 = \frac{0.250 \times 2}{3} \mu sec$$

$$SUBS = \frac{0.250}{3} \mu sec$$

$$BNE \ = 0.250 \ \mu sec$$

$$BX \ LR \ = 0.250 \ \mu sec$$

$$POP \ \{LR, RO\} = 0.250 \ \mu sec$$

- When we call the DELAY function from main part BL takes $0.250 \ \mu sec$ so ignore it. Then $PUSH \ \{LR, RO\}, LDR \ R1, BX \ LR, POP \ \{LR, RO\}$ these four lines will be executed ones so we will subtract them from 150msec

$$150000 \mu \ \text{sec} - \frac{0.250 \times 2}{3} \mu sec - 0.750 \ \mu sec$$

- In every loop we used 8 NOP and one SUBS these take $0.750 \ \mu sec$ and one BNE takes $0.250 \ \mu sec$ so in every loop we consume $1 \ \mu sec$ if we divide the remaining part with 1qsec then we require 149.999 times loop. One can see the consumed time in figure1. Note that main part takes $250 \ \mu sec$ so ignore it from the result.

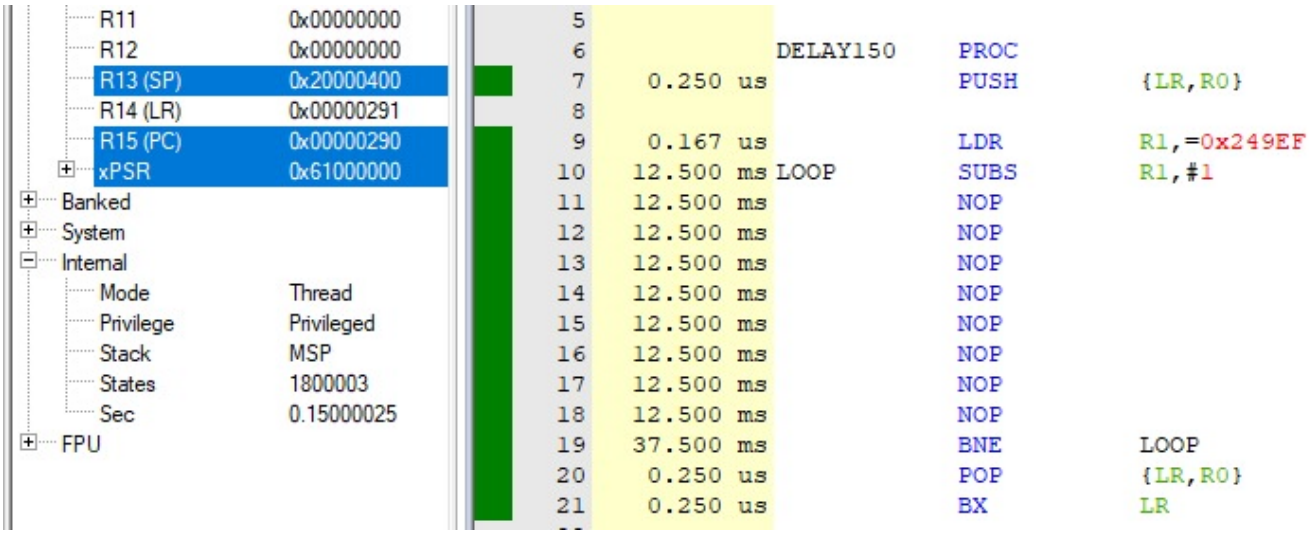


Figure 1: DELAY150 function result

DELAY150.s | mainone.s

```
1                ;LABEL      DIRECTIVE   VALUE        COMMENT
2                            AREA        sdata, DATA, READONLY
3                            THUMB
4                ;*****************************************************************
5                ; Program section
6                ;*****************************************************************
7                ;LABEL      DIRECTIVE   VALUE        COMMENT
8                            AREA        main, READONLY, CODE
9                            THUMB
10                           EXTERN      DELAY150
11                           EXPORT      __main  ; Make available
12
13               __main      PROC
14     0.250 us              BL          DELAY150
15               done        B           done
16                           ENDP
17                           END
```

Figure 2: Main part of the question 1

DELAY150.s | mainone.s

```
1                ;LABEL      DIRECTIVE       VALUE                COMMENT
2                            AREA routines , CODE, READONLY
3                            THUMB
4                            EXPORT DELAY150
5
6                DELAY150    PROC
7      0.250 us              PUSH        {LR,R0}
8
9      0.167 us              LDR         R1,=0x249EF
10    12.500 ms LOOP         SUBS        R1,#1
11    12.500 ms              NOP
12    12.500 ms              NOP
13    12.500 ms              NOP
14    12.500 ms              NOP
15    12.500 ms              NOP
16    12.500 ms              NOP
17    12.500 ms              NOP
18    12.500 ms              NOP
19    37.500 ms              BNE         LOOP
20     0.250 us              POP         {LR,R0}
21     0.250 us              BX          LR
22
23                           ENDP
24                           ALIGN
25                           END
26
27
28
```

Figure 3: DELAY150 function codes

Question 2) Simple Input LED system

maintwo.s | init_portB.s | DELAY150.s

```
; EQU Directives
; These directives do not allocate memory
;*********************************************************************
;SYMBOL             DIRECTIVE   VALUE           COMMENT
PB_INP              EQU         0x4000503C      ; data register for inputs
PB_OUT              EQU         0x400053C0      ; data register for outputs
number              EQU         20
;*********************************************************************
; Program section
;*********************************************************************
;LABEL      DIRECTIVE   VALUE           COMMENT
            AREA        main, READONLY, CODE
            THUMB
            EXTERN      DELAY150
            EXTERN      init_portB
            EXPORT      __main
__main      PROC
            BL          init_portB      ; initialize PORTB
loop
            MOV         R9,#number
            MOV32       R0, #PB_OUT
            LDR         R1, [R0]
            MVN         R1, R1
            AND         R2,R1,#0xF0
            LSR         R2, R2, #4
            MVN         R3, R2

            MOV32       R0, #PB_INP
            STR         R3, [R0]
            B           delay
delay       CMP         R9,#0
            BEQ         loop
            SUB         R9,R9,#1
            BL          DELAY150
            BL          delay

            ENDP
;*********************************************************************
```

Figure 4: Part two main

maintwo.s | init_portB.s | DELAY150.s

```
;*****************************************************************
; EQU Directives
; These directives do not allocate memory
;*****************************************************************
;SYMBOL              DIRECTIVE   VALUE           COMMENT
SYSCTL_RCGC2         EQU         0x400FE108
GPIO_PORTB_DIR       EQU         0x40005400
GPIO_PORTB_AFSEL     EQU         0x40005420
GPIO_PORTB_PUR       EQU         0x40005510
GPIO_PORTB_DEN       EQU         0x4000551C

;*****************************************************************
; Program section
;*****************************************************************
;LABEL      DIRECTIVE   VALUE           COMMENT
            AREA        main, READONLY, CODE
            THUMB
            EXPORT      init_portB
init_portB  PROC
            PUSH        {R0,R1,LR}

;Enable the clock signal
            LDR         R0,=SYSCTL_RCGC2
            LDR         R1,[R0]
            ORR         R1,#0x12
            STR         R1,[R0]
            NOP
            NOP
            NOP

;
            LDR         R0,=GPIO_PORTB_DIR
            LDR         R1,[R0]
            BIC         R1, #0xF0       ; clear bit 7-4 for inputs
            ORR         R1, #0x0F       ; clear bit 3-0 for outputs
            STR         R1,[R0]

```

Figure 5:Part two port initialization 1

maintwo.s | init_portB.s | DELAY150.s

```
;
            LDR         R0,=GPIO_PORTB_DIR
            LDR         R1,[R0]
            BIC         R1, #0xF0       ; clear bit 7-4 for inputs
            ORR         R1, #0x0F       ; clear bit 3-0 for outputs
            STR         R1,[R0]


            LDR         R0,=GPIO_PORTB_AFSEL
            LDR         R1,[R0]
            BIC         R1,#0xFF
            STR         R1,[R0]

; pull up resistors for the switches
            LDR         R0,=GPIO_PORTB_PUR
            MOV         R1,#0xF0
            STR         R1,[R0]


            LDR         R0,=GPIO_PORTB_DEN   ; Enable digital
            LDR         R1,[R0]
            ORR         R1,#0xFF
            STR         R1,[R0]

            POP         {R0,R1,LR}

            BX          LR

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

```

Figure 6:Part two port initialization 2

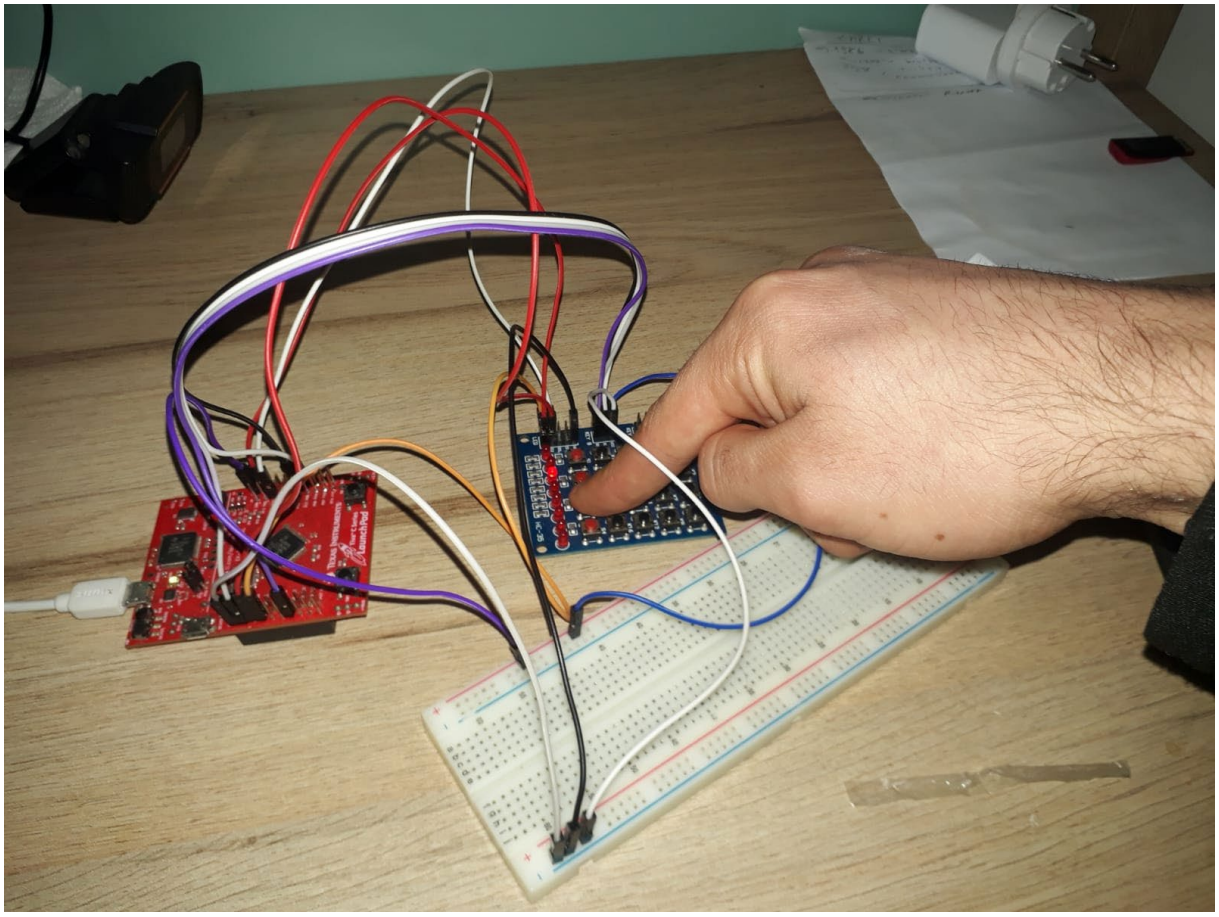


Figure 7: A sample photo that show system configuration

*Question 3) 4*4 Keypad system*

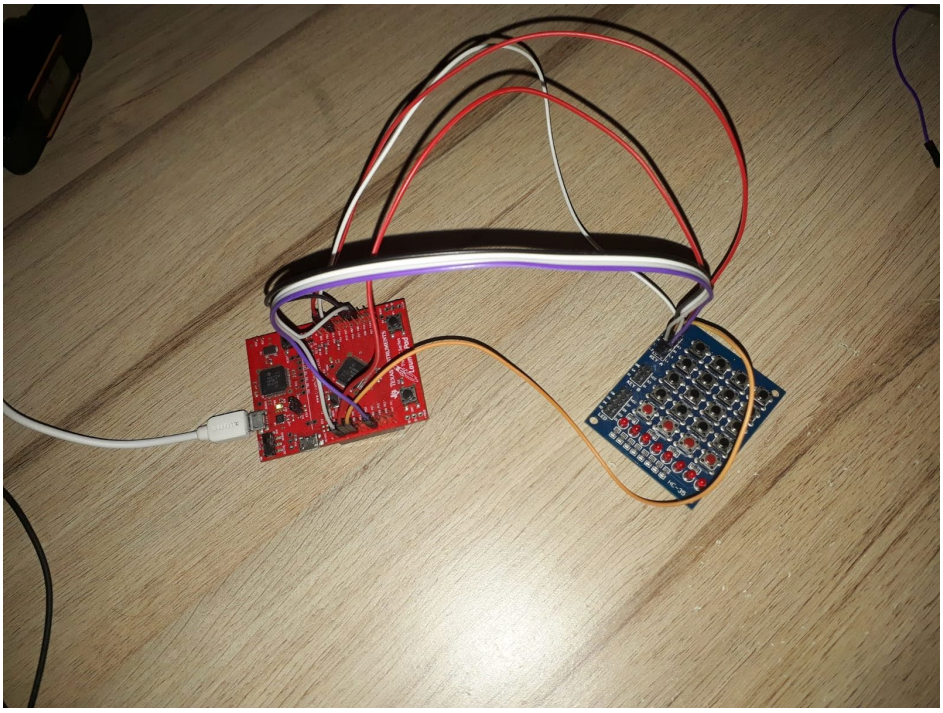


Figure 8: PART3-Circuit Connection Diagram

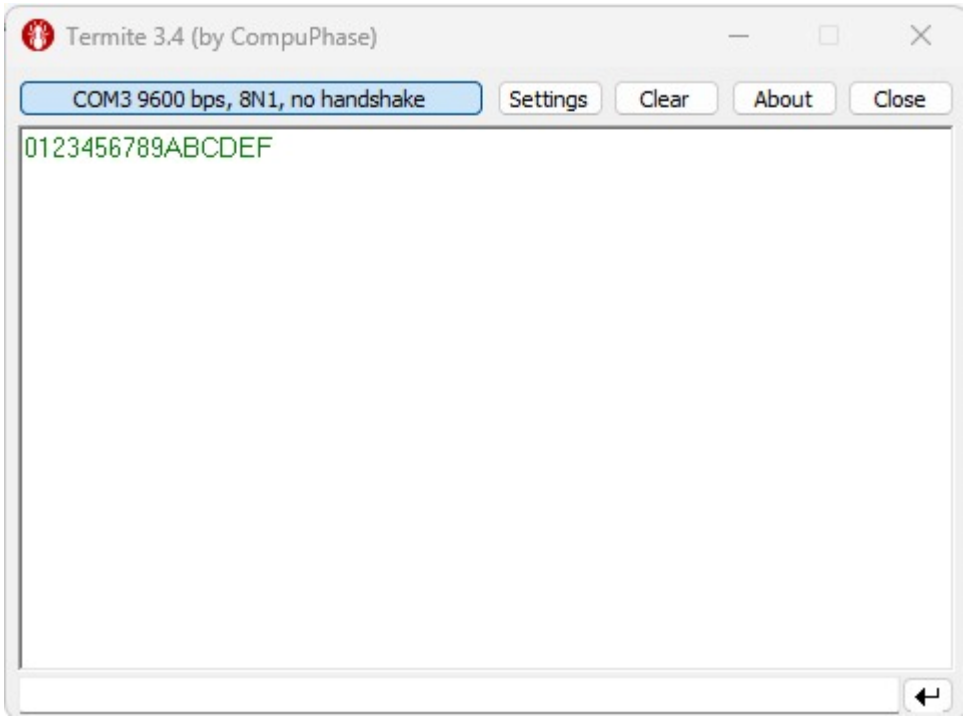


Figure 9: Termite Screen to show the results

mainthree.s | init_portB.s | DELAY150.s

```
;*****************************************************************
; mainthree.s
;*****************************************************************
; EQU Directives
; These directives do not allocate memory
;*****************************************************************
;SYMBOL              DIRECTIVE   VALUE           COMMENT
PB_INP               EQU         0x4000503C      ; data register for inputs
PB_OUT               EQU         0x400053C0      ; data register for outputs
number               EQU         20              ; DELAY150 loop time
FO                   EQU         0xF0            ; 1111_0000
OF                   EQU         0x0F            ; 0000_1111

;*****************************************************************
; Program section
;*****************************************************************
;LABEL      DIRECTIVE   VALUE           COMMENT
            AREA        main, READONLY, CODE
            THUMB
            EXTERN      init_portB
            EXTERN      DELAY150
            EXPORT      __main
            EXTERN      OutChar
__main      PROC
            BL          init_portB      ;Initialize port B
            MOV         R5,#2           ;2 for shift operations
            MOV         R10,#3          ;short time delay for debouncing
start       LDR         R1, =PB_OUT     ;ARM input
press_check LDR         R2, =PB_INP     ;KEYPAD input
            MOV         R6, #0          ;Initialization
            STR         R6, [R2]        ;R2=R6
            LDR         R0, [R1]        ;R0 = ARM input
            AND         R0, #FO
            CMP         R0, #FO         ;Comparison
            MOV         R3, R0          ;R3:Temporary data
            BEQ         press_check
            BL          DELAY150
            LDR         R0, [R1]
```

Figure 10:Main_Part_Three_1

mainthree.s | init_portB.s | DELAY150.s

```
37              BL          DELAY150
38              LDR         R0, [R1]
39              AND         R0, #F0
40              CMP         R0, #F0
41              MOV         R9, R0          ;R9:Temporary data
42              BEQ         press_check
43              CMP         R3, R9          ;DEBOUNCING
44              BEQ         row_det         ;DEBOUNCING
45              BNE         press_check     ;DEBOUNCING
46  row_det     MOV         R4, #0xEF       ; Determinin the row
47
48  shift       UDIV        R4,R5           ;Divide R4 value with 2
49              STR         R4, [R2]
50  Shortdelay  CMP         R10,#0
51              NOP
52              SUBS        R10,#1
53              BNE         exit
54              B           Shortdelay
55  exit        MOV         R10,#3
56              LDR         R0, [R1]
57              AND         R0, #F0
58              CMP         R0, #F0
59              MOV         R3, R0
60              BEQ         shift           ;According to button pressed or not
61              B           go_to
62
63  go_to       LSR         R0, #4          ;Divide by sixteen
64              MVN         R0, R0          ;Take coloumn number
65              AND         R0, #0F
66              MVN         R4, R4          ;Take row number
67              AND         R4, #0F         ;By using and reset 7-4 bits
68
69  ; Check Corresponding Row Number
70              MOV         R7, #0          ;Row counter
71  row_n       UDIV        R0, R5          ;Divide R0 value with 2
72              CMP         R0, #0x0
73              BEQ         mov_column
74              ADD         R7, #1          ;Number of the corresponding row
```

Figure 11:Main_Part_Three_2

mainthree.s | init_portB.s | DELAY150.s

```
73              BEQ         mov_column
74              ADD         R7, #1          ;Number of the corresponding row
75              BNE         row_n
76  mov_column  MOV         R8, #0          ;Column counter
77  column_n    UDIV        R4, R5          ;Divide R4 value with 2
78              CMP         R4, #0x0
79              BEQ         Conc
80              ADD         R8, #1          ;Number of the corresponding column
81              BNE         column_n
82
83  Conc        MOV         R9, #4          ; TO MULTIPLY ROW BY 4 --> BUTTON ID = ROW*4 + COLUMN
84              MUL         R7, R7, R9
85              ADD         R9, R8, R7      ; Store value of the pressed button at R8
86              CMP         R9, #0x9
87              ADDLS       R0, R9, #0x30
88              ADDHI       R0, R9, #0x37
89              BL          OutChar
90
91  check_r     LDR         R2, =PB_INP     ;Check if the pressed buton is released or not
92              MOV         R6, #0x00
93              STR         R6, [R2]
94
95              BL          DELAY150
96
97  checker1    LDR         R1, =PB_OUT
98              LDR         R0, [R1]
99              AND         R0, #F0
100             CMP         R0, #F0
101             BNE         check_r
102             BL          DELAY150
103             BEQ         checker2
104
105 checker2    LDR         R1, =PB_OUT
106             LDR         R0, [R1]
107             AND         R0, #F0
108             CMP         R0, #F0
109             BNE         checker1
110             BL          DELAY150
111             BEQ         start           ;Turn to beginning
112             ENDP
113             ALIGN
114             END
115
```

Figure 12:Main_Part_Three_3

Part - a)

- Since just one row of switches provides data when it is pressed in the case of one-by-one enabling, it is possible to identify the pressed switch's row by examining the change in the input data register.

b)

- By checking energy level of the all input pins. If they are all have high voltage level then we can say that the pressed button is relased. However if there is one pin with low level voltage which means that the pressed button is not released.

c)

- By checking all rows and inputs we can understand which key is pressed. Just check their voltage level and search for low level one.

d)

- Debouncing can be avoided by apply some delay time after pressing or double checking the input pin

e)

-

f)

-