



Computer Technology
(1DT301)

Lab 1:
(How to use the PORTs, Digital input/output, Subroutine call)

Authors:
Andrei Neagu (AN223KJ)
Konstantinos Tatsis (KT222IQ)
GitHub:

<https://github.com/neaguandrei101/1DT301?fbclid=IwAR3T9jGI4bulsvSGeOH3b5SmADzAx90pfXcW1SufpqhMfpXSa7peUJR8LoU>

Table of Contents

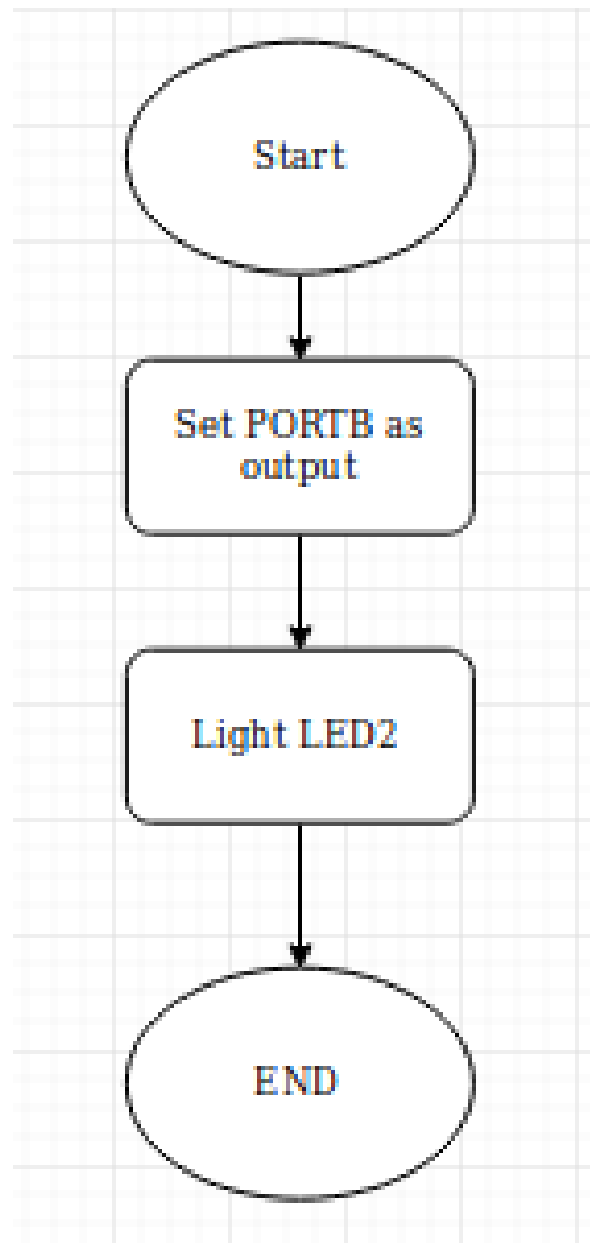
Task 1:.....	3
Task 2:.....	5
Task 3:.....	7
Task 4:.....	9
Task 5:.....	10
Task 6:.....	13

Task 1:

Write a program in Assembly language to light LED 2. You can use any of the four ports, but start with PORTB. The program should be very short! How many instructions is minimum number?

[illegible]

Flowchart 1:

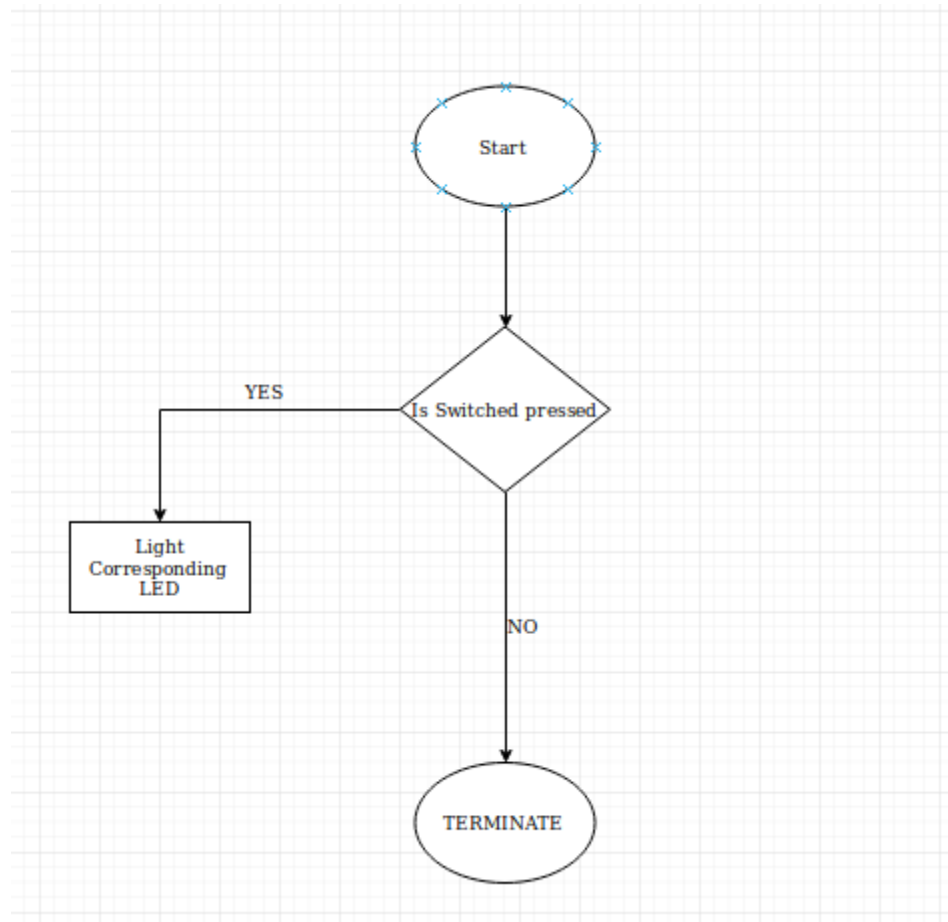


Task 2:

Write a program in Assembly language to read the switches and light the corresponding LED. Example: When you press SW5, LED5 so should light. Make an initialization part of the program and after that an infinite loop.

[illegible]

Flowchart 2:



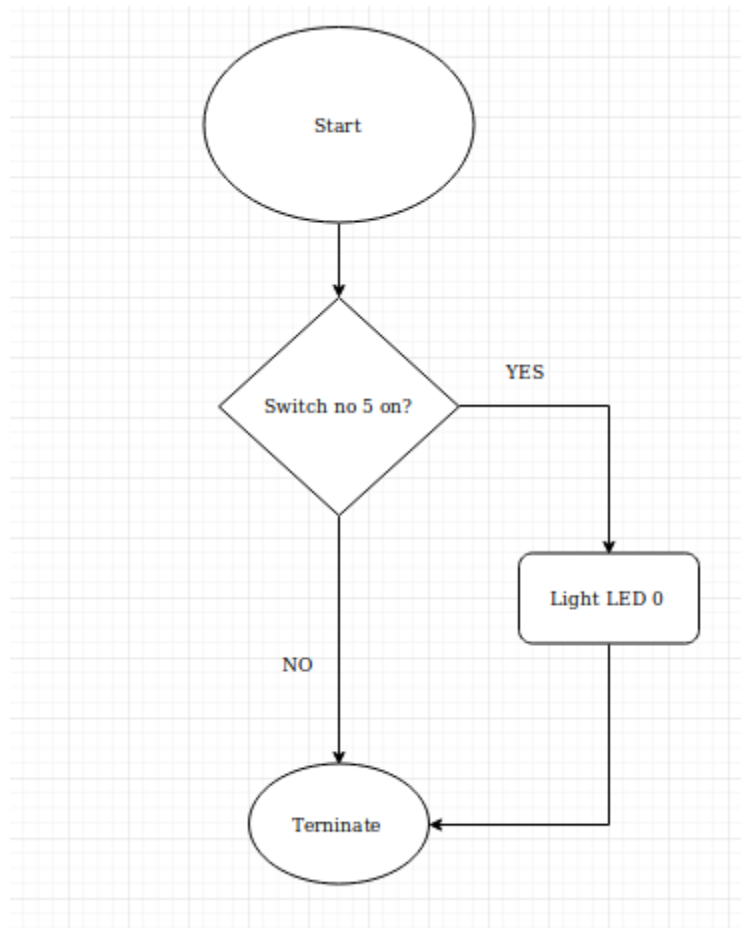
Write a program in Assembly language to read the switches and light LED0 when you press SW5. For all other switches there should be no activity.

```

2 ; IDT301, Computer Technology I
3 ; Date: 2019-09-09
4 ; Author: Andrei Neagu (an223kj)
5 ; Konstantinos Tatsis (kt222iq)
6 ;
7 ; Lab number:          1
8 ; Title:               How to use the PORTs. Digital input /output.
9 ;                     Subroutine call.
10 ;
11 ; Hardware:            STK600, CPU ATmega2560
12 ;
13 ; Function:             Light LED0 if SW5 is pressed
14 ;
15 ; Input ports:         PORTA
16 ;
17 ; Output ports:        PORTB
18 ;
19 ; Subroutines:         No subroutines used
20 ;
21 ; Included files:       m2560def.inc
22 ;
23 ; Other information:    If other switch rather than switch 5 is pressed
24 ;                       no led should light.
25 ;
26 ; Changes in program:
27 ;                               2019-09-09
28 ;
29 ;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
30 .include "m2560def.inc"
31
32 ldi r16, 0b1111_1111 ;value to set the DDRB as outputs
33 out DDRB, r16 ;sets PORTB using DDRB as a output port using the binary value stored r16
34
35 ldi r16, 0b0000_0000 ;value to set the DDRA as inputs
36 out DDRA, r16 ;sets PORTA using DDRA as a input port using the binary value stored r16
37
38 ldi r16, 0b1111_1111 ;turn off the leds
39 out portB, r16
40
41 ldi r18, 0b1101_1111 ; when sw5 is pressed PINA5 is 0
42 ldi r19, 0b1111_1110 ;code for LED0
43
44 loop:
45     in r17, PINA ; read PINA
46     cp r17, r18 ; compare r18 and r17
47     breq equal ; if r17 equal to r18 go to light
48 rjmp loop

```

Flowchart 3:



Task 4:

Task 4 must be performed on the computer cannot be displayed.

Continue to task 5.

Task 5:

Write a program in Assembly language that creates a Ring Counter. The values should be displayed with the LEDs. Use shift instructions, LSL or LSR. Make a delay of approximately 0.5 sec in between each count. Write the delay as a subroutine. For using the subroutine, you must initialize the Stack Pointer, SP. Include the following instructions in beginning of your program:

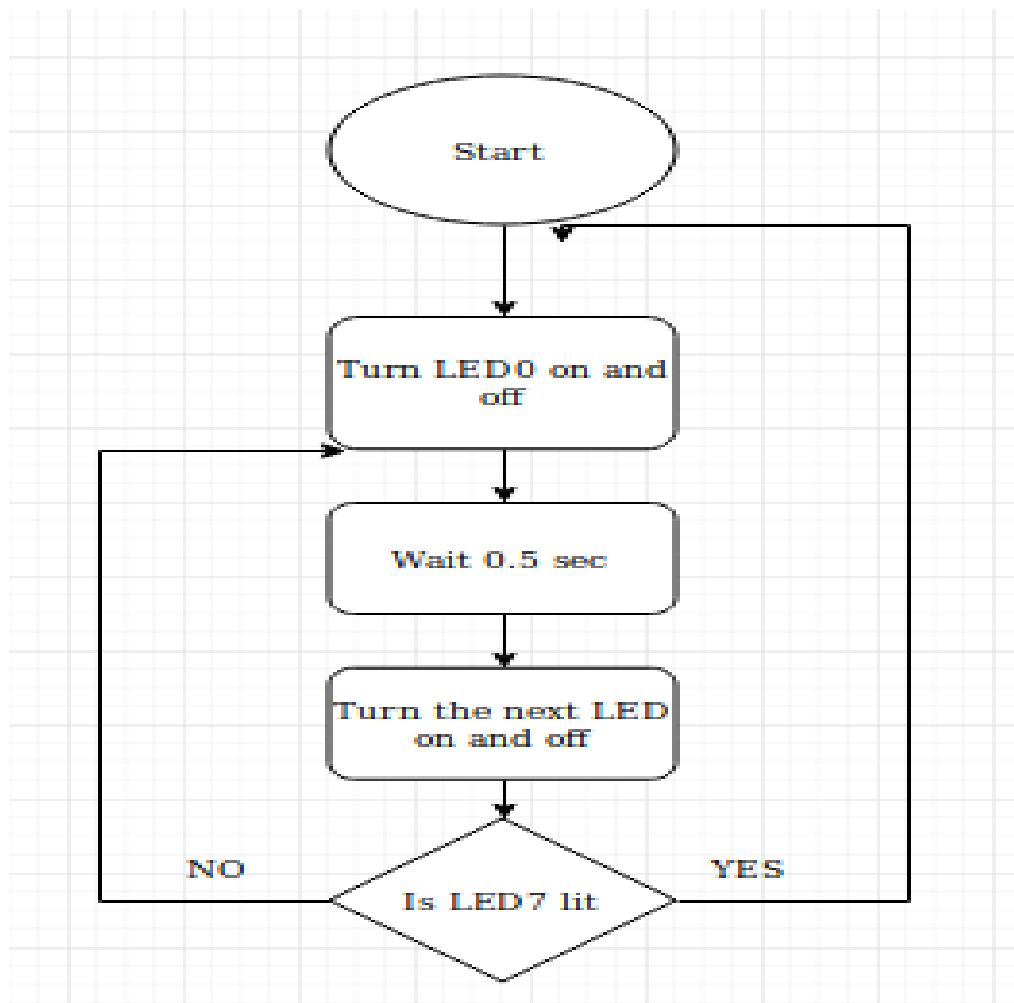
[illegible]

```

29  .include "m2560def.inc"
30
31  ldi R20, HIGH (RAMEND)
32  out SPH, R20
33
34  ldi R20, low(RAMEND)
35  out SPL, R20
36
37  start:
38  ldi r16, 0x0000_0001      ;initial led state
39  out DDRB, R16             ;initial output to LED0
40  rcall delay               ;call delay of 500ms
41
42
43  loop:
44  lsl R16                   ;shift bits to the left
45  out DDRB, R16             ;output of shifted value again
46  cpi R16, 0x00             ;if r16 equals to 0x00 then restart loop
47  breq again                ;by going to again
48  rcall delay               ;delay again
49  rjmp loop                 ;repeat if not equal to 0x00
50
51  again:
52  rjmp start
53
54
55  delay:                    ;this delay is approx 500ms
56      ldi r18, 5
57      ldi r19, 15
58      ldi r20, 242
59  L1: dec r20
60      brne L1
61      dec r19
62      brne L1
63      dec r18
64      brne L1
65
66
67  RET

```

Flowchart 5:



Task 6:

Write a program in Assembly language that creates a Johnson Counter in an infinite loop.

```

1 ; >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
2 ;   IDT381, Computer Technology I
3 ;   Date: 2019-09-09
4 ;   Author: Andrei Neagu (an223kj)
5 ;           Konstantinos Tatsis (kt222iq)
6 ;
7 ;   Lab number:          1
8 ;   Title:               How to use the PORTs. Digital input /output.
9 ;                       Subroutine call.
10 ;
11 ;   Hardware:            STK600, CPU ATmega2560
12 ;
13 ;   Function:            Program with Johnson Counter with an infinite loop
14 ;
15 ;   Input ports:
16 ;
17 ;   Output ports:        PORTB
18 ;
19 ;   Subroutines:         Delay
20 ;
21 ;   Included files:       m2560def.inc
22 ;
23 ;   Other information:    Program with Johnson Counter with an infinite loop
24 ;
25 ;   Changes in program:
26 ;                                   2019-09-09
27 ;
28 ; <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
29 .include "m2560def.inc"
30
31 ldi r20, HIGH (RAMEND)
32 out SPH, R20      ; SPH high part of the RAMEND
33 ldi R20, low (RAMEND)
34 out SPL, R20      ; SPL low part of RAMEND
35
36 ldi r16, 0xFF
37 out DDRB, r16     ; DDRB is set as an output port
38
39 ldi r21, 0b1111_1110 ; initial led state
40 ldi r22, 0xFF        ; leds off
41 ldi r23, 0x00        ; leds on
42

```

```

42
43     loop:
44         out PORTB, r21 ; initial led state
45         LSL r21        ; shift leds to left
46         CALL Delay     ; delay of 500ms
47
48         cp r21, r23    ; when r21 equals to r23 go to light
49         brsq light
50     rjmp loop
51
52
53     light:
54         out PORTB, r23 ; leds on
55         CALL Delay     ; delay of 500ms
56         ldi r21, 0b1000_0000 ; initialize the leds to go to the right
57         out PORTB, r21
58         Sec_loop:
59             out PORTB, r21 ; initialize the leds to go to the left
60             ASR r21        ; shift the bits to the right
61             CALL Delay     ; delay of 500ms
62             cp r21, r22    ; if r21 equals r22 go back to loop
63             brsq loop
64         rjmp Sec_loop
65
66     Delay:
67         ldi r18, 21
68         ldi r19, 140
69         ldi r20, 174
70
71     L1: dec r20
72         brne L1
73         dec r19
74         brne L1
75         dec r18
76         brne L1
77         rjmp PC+1
78     RET

```

Flowchart 6:

