



Computer Technology I

Lab. 2 : Subroutines



Author: LOIC GALLAND,
LEONARDO PEDRO

Supervisor:

Semester: Autumn 2019

Area: Computer Science

Course code: 1DT301

Contents

1	Task 1 - Switch – Ring counter / Johnson counter	1
2	Task 2 - Electronic dice	4
3	Task 3 - Change counter	6
4	Task 4 - Delay subroutine with variable delay time	8

1 Task 1 - Switch – Ring counter / Johnson counter

Write a program which switch between Ring counter and Johnson counter. You should not use Interrupt in this lab. The pushbutton must be checked frequently, so there is no delay between the button is pressed and the change between Ring/Johnson. Use SW0 (PA0) for the button. Each time you press the button, the program should change counter.

[illegible]

```

ldi r22, 0xFF
ldi r23, 0b11111110

RC_loop:
    out portB, r17
    rol r17
    CALL Delay1
    in r25, PINA
    cp r25,r24
    breq JC
    cp r17, r22
    breq RC_light
rjmp RC_loop
RC_light:
    rol r17
    out portB, r17
    rjmp RC_loop
rjmp RC

JC:
ldi r21, 0b11111110
ldi r22, 0b11111111 ;desired one
ldi r23, 0b00000000

my_loop1:
    out portB, r21
    LSL r21
    CALL Delay1
    in r25, PINA
    cp r25,r24
    breq RC
    cp r21, r23 ;compare info with desired one
    breq light
rjmp my_loop1

light:
    out portB, r23
    CALL Delay1
    ldi r21, 0b10000000
    out portB, r21
Second_loop:
    in r25, PINA
    cp r25,r24
    breq RC
    out portB, r21
    ASR r21
    CALL Delay1
    cp r21, r22 ;compare info with desired one
    breq my_loop
rjmp Second_loop
rjmp JC

Delay1:
; Generated by delay loop calculator
; at http://www.bretmulvey.com/avrdelay.html
; Delay 1 950 500 cycles
; 500ms at 3.901 MHz

ldi r18, 10

```

```
    ldi r19, 230
    ldi r20, 22
L1: dec r20
    brne L1
    dec r19
    brne L1
    dec r18
    brne L1
RET
```

This is the flowchart of the task 1:

2 Task 2 - Electronic dice

You should create an electronic dice. Think of the LEDs placed as in the picture below. The number 1 to 6 should be generated randomly. You could use the fact that the time you press the button varies in length.

[illegible]

```

Listening_For_Switch_Release:
    inc r19
    cpi r19,7
    breq reset
    in r17,PINA
    cp r17,r25
    breq RD
rjmp Listening_For_Switch_Release

reset:
ldi r19,1
rjmp Main

RD:
    cpi r19,1
    breq ONE
    cpi r19,2
    breq TWO
    cpi r19,3
    breq THREE
    cpi r19,4
    breq FOUR
    cpi r19,5
    breq FIVE
    cpi r19,6
    breq SIX
rjmp RD

ONE:
ldi r18,0b11101111
out PortB,r18
rjmp Listening_For_Switch_Press

TWO:
ldi r18,0b10111011
out PortB,r18
rjmp Listening_For_Switch_Press

THREE:
ldi r18,0b10101011
out PortB,r18
rjmp Listening_For_Switch_Press

FOUR:
ldi r18,0b00111001
out PortB,r18
rjmp Listening_For_Switch_Press

FIVE:
ldi r18,0b00101001
out PortB,r18
rjmp Listening_For_Switch_Press

SIX:
ldi r18,0b00010001
out PortB,r18
rjmp Listening_For_Switch_Press

```

This is the flowchart of the task 1:

3 Task 3 - Change counter

Write a program that is able to count the number of changes on a switch. As a change we count when the switch SW0 goes from 0 to 1 and from 1 to 0, we expect therefore positive and negative edges. We calculate the changes in a byte variable and display its value on PORTB.

[illegible]


```

        in r19, PINA
        cp r18,r19
        breq counter
rjmp my_loop

counter:
inc r25
mov r20,r25
com r20
out portB,r20
        loop:
                in r19,PINA
                cp r19,r17
                breq counter2
        rjmp loop

counter2:
inc r25
mov r20,r25
com r20
out portB,r20
rjmp my_loop

```

This is the flowchart of the task 3:

4 Task 4 - Delay subroutine with variable delay time

[illegible]

```

Delay:
    ldi r24, low(INPUT)
    ldi r25, high(INPUT)
wait_milliseconds:
    call ms_delay
    sbiw r25:r24,1
    cpi r25, high(0)
    breq reset
    rjmp wait_milliseconds
reset:
RET

ms_delay:
    ; Generated by delay loop calculator
    ; at http://www.bretmulvey.com/avrdelay.html
    ;
    ; Delay 1 000 cycles
    ; 1ms at 1 MHz

    ldi r18, 2
    ldi r19, 75
L1: dec r19
    brne L1
    dec r18
    brne L1
    rjmp PC+1
RET

```

This is the flowchart of the task 1: