```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;1DT301, Computer Technology I
;Date: 2019-09-30
;
;Author:
;   Student name 1:          Einar van de Velde
;   Student name 2:          Abdulla Mehdi
;
;Lab number:                 3.
;Title:                      Task 2.
;
;Hardware:                   STK600, CPU ATmega2560.
;
;Function:                   Switch to flash 8 LEDs either in the form of a ring
;                            counter or in the form of a Johnson counter.
;                            Use the switch SW0 connected to PORTD to switch
;                            between  the two counters. Each time the button is
;                            pressed, a shift between the two counters should
;                            take place. By using interrupts we'll swap directly
;                            without a delay.
;
;Input ports:                 PORTD.
;
;Output ports:               On-board LEDs connected to PORTB.
;
;Subroutines:                INTERRUPT_0.
;
;Included files:             m2560def.inc
;
;Other information:
;Changes in program:
;         File Created (2019-09-27)
;                  Program is runnable (2019-09-29)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
.include"m2560def.inc"
.org 0x00
rjmp START

.org INT0addr
rjmp INTERRUPT_0

.org 0x72
START:
            ldi r20, HIGH(RAMEND)
            out SPH,r20
            ldi r20, LOW(RAMEND)
            out SPL, r20
```

```
        ldi r16, 0x00
        out DDRD, r16
        out DDRA, r16

        ldi r16, 0xFF
        out DDRE, r16
        out DDRB, r16

        ldi r16, 0b00000001
        out EIMSK, r16

        ldi r16, 0b00000010
        sts EICRA, r16

        sei

;-------------------
MAIN:
        ldi r16, 0xFF
        out DDRB, r16
        out PORTB, r16
        com r16
        out DDRD, r16

        ldi r16, 0

LOOP:
        cpi r16, 1
        breq RING
        cpi r16, 2
        breq JOHNSSON
rjmp LOOP

;------------------- INTERRUPT -------------------
INTERRUPT_0:
        cpi r16, 1
        breq TO_RING
        ldi r16, 1
rjmp DONE

TO_RING:
        ldi r16, 2

DONE:
        reti

;------------------- RING ————————————————
RING:
        ldi r18, 0b11111110
```

```
            out PORTB, r18

LOOP_1:
            rcall DELAY_1
            rol r18
            inc r18
            out PORTB, r18

            cpi r18, 0b11111111
            breq RING
rjmp LOOP_1
;------------------- JOHNSSON ---------------------
JOHNSSON:
            ldi r18, 0b11111110
            out PORTB, r18

STEP_1:
            rcall DELAY_3

            rol r18
            out PORTB, r18
            cpi r18, 0x00
            breq STEP_2
rjmp STEP_1
;-------
STEP_2:
            rcall DELAY_3
            ldi r18, 0b10000000
            out PORTB, r18

LOOP_2:
            rcall DELAY_3

            asr r18
            out PORTB, r18
            cpi r18, 0xFF
            breq JOHNSSON
rjmp LOOP_2

;-------------------- DELAY_1 ————————————————
DELAY_1:
            ldi r21, 255
            ldi r22, 0

DEL_1:
            rcall DELAY_2
            inc r22
            cp r21, r22
            brne DEL_1
```

```
ret

DELAY_2:
        ldi r23, 255
        ldi r24, 0

DEL_2:
        cpi r16, 1
        brne JOHNSSON

        inc r24
        cp r23, r24
        brne DEL_2
ret

;------------------- DELAY_3 -------------------
DELAY_3:
        ldi r21, 255
        ldi r22, 0

DEL_3:
        rcall DELAY_4
        inc r22
        cp r21, r22
        brne DEL_3
ret
------------------- DELAY_4 -------------------
DELAY_4:
        cpi r16, 2
        brne RING

        ldi r23, 255
        ldi r24, 0

DEL_4:
        inc r24
        cp r23, r24
        brne DEL_4
ret
```