lab3

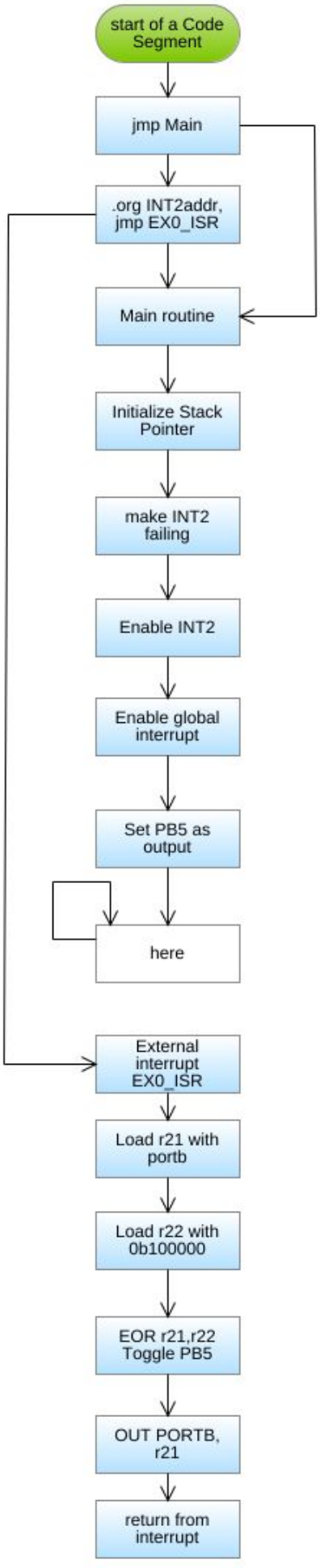
lh223ng och fe222pa

November 2021

1 task1

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2021-09-18
; Author:
; Student name Li Ang Hu
; Student name Fredric Eriksson Sep lveda
;
; Lab number: 3
; Title: task1
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Write a program that switches a LED ON/OFF when a push button pressed. To dete
button has been pressed an interrupt shall be used (use PORTD).
The program shall have a main program that runs in a loop and wait for the interrupt to be
An interrupt routine is called when the push button is pressed. Each time the button is pr
the lamp shall switch from OFF to ON, or from ON to OFF.

; Input ports: PORTD
;
; Output ports: PORTB
;
; Subroutines: If applicable.
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
;
; task1.asm
;
; Created: 2021-09-28 09:37:40
; Author : lh223ng
;


; Replace with your application code
.cseg ; Code segment to assemble to
.org 0
/*ldi r23, 0xff
out ddrd, r23
out portd, r23*/

rjmp Main
```

```
.org INT2addr ;
rjmp EX0_ISR

Main:
; Initialize the Stack Pointer
ldi r16, LOW(RAMEND)
out SPL, r16
ldi r16, HIGH(RAMEND)
out SPH, r16

LDI R16, 0b100 ; make INT2 failing
STS EICRA, r16

SBI PORTD, 2 ; pull-up enabled, input pin PIND2
LDI R16, 1<<INT2 ; enable INT2
OUT EIMSK, R16

SEI ; enable interrupt
SBI DDRB, 5 ; PORTB5 as ouput
here:
inc r25
rjmp here

EX0_ISR:
IN R21, PORTB
LDI R22, (1<<5) ; for toggling PB5
EOR R21, R22
OUT PORTB, R21
reti
```


start of a Code Segment
jmp Main
.org INT2addr, jmp EX0_ISR
Main routine
Initialize Stack Pointer
make INT2 failing
Enable INT2
Enable global interrupt
Set PB5 as output
here
External interrupt EX0_ISR
Load r21 with portb
Load r22 with 0b100000
EOR r21,r22 Toggle PB5
OUT PORTB, r21
return from interrupt

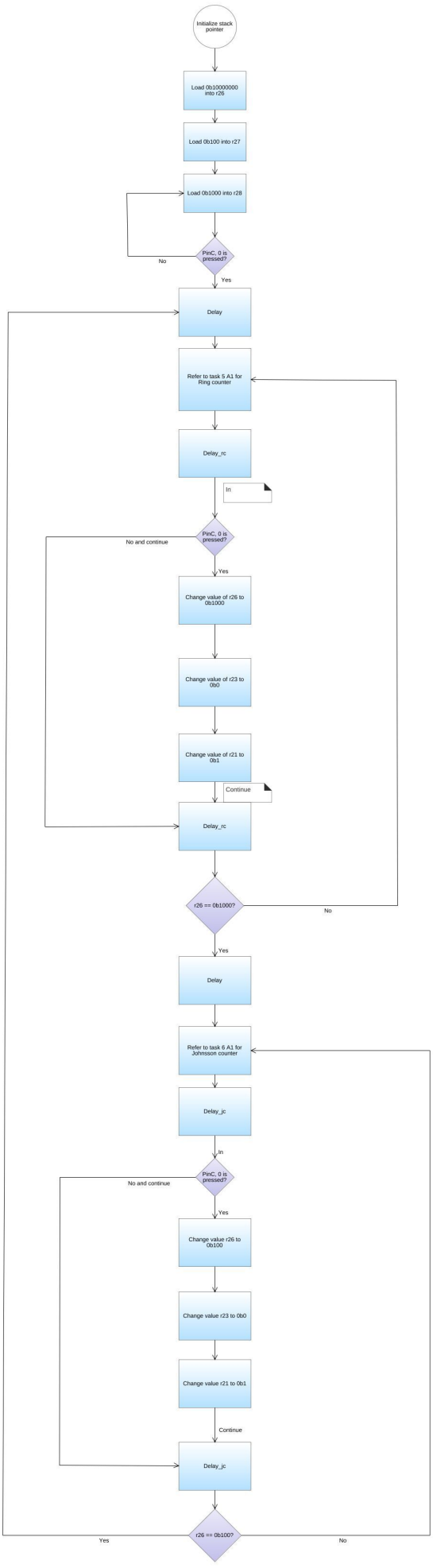
2 task2

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2021-09-18
; Author:
; Student name Li Ang Hu
; Student name Fredric Eriksson Sep lveda
;
; Lab number: 3
; Title: task2
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Write a program that by means of a switch can choose to flash 8 LEDs either in
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 y o u ll swap directly with no delay.

; Input ports: PORTD
;
; Output ports: PORTB
;
; Subroutines: If applicable.
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
;
; task2.asm
;
; Created: 2021-10-02 16:28:52
; Author : fe222pa och lh223ng
;



.org 0x00
rjmp innit
rjmp interrupt

innit:
rcall input_Delay
ldi r31, LOW(RAMEND)
out spl, r31
ldi r31, HIGH(RAMEND)
out sph, r31


LDI R16, 0x2 ; make INT0 failing
STS EICRA, r16

SBI PORTD, 2 ; pull-up enabled, input pin PIND2
LDI R16, 1<<INT0 ; enable INT0
OUT EIMSK, R16
```

```
ldi r19, 0xff

ldi r21, 0b1;
out DDRB, r19
ldi r25, 0b1; set this constant 0b1

ldi r24, 0b1; always be 1
ldi r23, 0b0; temp register
ldi r22, 0b0; always be 0

ldi r27, 0xff

sei

cpi r26, 0xff
breq jc
cpi r26, 0b0
breq rc




rjmp innit



rc:

ldi r26, 0xff
com r21 ; invert r21 to make
out PORTB, r21
com r21 ; revert r21 back

rcall Delay_1sec

lsl r21; left shift one bit

cpi r21, 0b0 ;compare with 0
breq reset ; if r21 is 0, then reset r21

rjmp rc

reset:
com r21 ; To set r21 to 1111 1111
out portB, r21 ; turn all LEDs off
rcall Delay_1sec; delay 0.5 sec
mov r21, r25; reset r21 to 0b1
;ret; make it stop permanently

rjmp rc



interrupt:

push r26

push r16
in r16, SREG
push r16
```

```
pop r16
out SREG, r16
pop r26
RETI



jc:
ldi r26, 0x0
or r23, r21
rol r21



com r23
out PORTB, r23; OUTPUT will be PORTB
com r23

rcall Delay_1sec

cpi r23, 0xff
breq decrease
rjmp jc

decrease:
lsr r23

com r23
out PORTB, r23; OUTPUT will be PORTD
com r23

rcall Delay_1sec

cpi r23, 0x0
breq reset1
rjmp decrease


reset1:
mov r23, r22 ; reset r23 to 0
mov r21, r24 ; reset r21 to 1


rjmp jc




Delay_1sec:                  ; For CLK(CPU) = 1 MHz
    LDI     r16,   4         ; One clock cycle;
Delay1:
    LDI     r17,   125       ; One clock cycle
Delay2:
    LDI     r18,   250       ; One clock cycle
```

```
Delay3:
    DEC     r18             ; One clock cycle
    NOP                     ; One clock cycle
    BRNE    Delay3          ; Two clock cycles when jumping to Delay3, 1 clock when continuin

    DEC     r17             ; One clock cycle
    BRNE    Delay2          ; Two clock cycles when jumping to Delay2, 1 clock when contin

    DEC     r16             ; One clock Cycle
    BRNE    Delay1          ; Two clock cycles when jumping to Delay1, 1 clock when contin
RET



input_Delay:                ; For CLK(CPU) = 1 MHz
    LDI     r16,    1       ; One clock cycle;
Delay1c:
    LDI     r17,    125     ; One clock cycle
Delay2c:
    LDI     r18,    250     ; One clock cycle
Delay3c:
    DEC     r18             ; One clock cycle
    NOP                     ; One clock cycle
    BRNE    Delay3c         ; Two clock cycles when jumping to Delay3, 1 clock when continu

    DEC     r17             ; One clock cycle
    BRNE    Delay2c         ; Two clock cycles when jumping to Delay2, 1 clock when conti

    DEC     r16             ; One clock Cycle
    BRNE    Delay1c         ; Two clock cycles when jumping to Delay1, 1 clock when conti
RET
```


Initialize stack pointer
Load 0b10000000 into r26
Load 0b100 into r27
Load 0b1000 into r28
PinC, 0 is pressed?
No
Yes
Delay
Refer to task 5 A1 for Ring counter
Delay_rc
In
PinC, 0 is pressed?
No and continue
Yes
Change value of r26 to 0b1000
Change value of r23 to 0b0
Change value of r21 to 0b1
Continue
Delay_rc
r26 == 0b1000?
No
Yes
Delay
Refer to task 6 A1 for Johnsson counter
Delay_jc
In
PinC, 0 is pressed?
No and continue
Yes
Change value r26 to 0b100
Change value r23 to 0b0
Change value r21 to 0b1
Continue
Delay_jc
r26 == 0b100?
Yes
No

3 task3 and task4

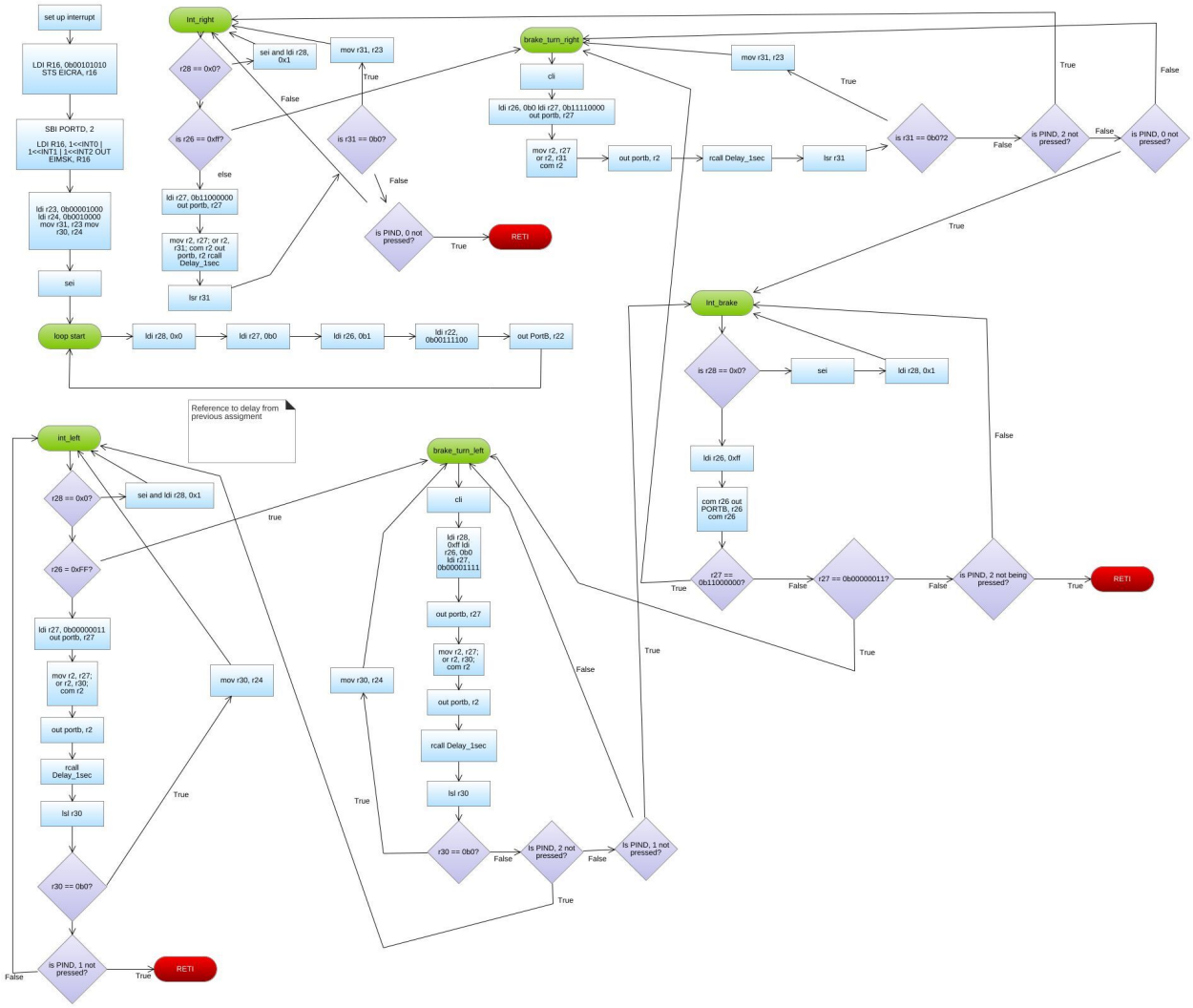
```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2021-09-18
; Author:
; Student name Li Ang Hu
; Student name Fredric Eriksson Sep lveda
;
; Lab number: 3
; Title: task3 and task4
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Add functions to the previous task for simulating stop lights when braking. Us
the braking.
Functions
1. Brake and no turning
Turn ON all LEDs (0-7).
2. Brake and turn right
LED 4    7 is ON, LED 0    3 is blinking as a Ring counter to the right..
3. Brake and turn left
LED 0    3 is ON, LED 4    7 is blinking as a Ring counter to the left..

; Input ports: PORTD
;
; Output ports: PORTB
;
; Subroutines: If applicable.
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
.org 0x00
rjmp innit

.org 0x02
rjmp int_right

.org 0x04
rjmp int_left

.org 0x06
rjmp int_brake

activate_seiC:
ldi r28, 0x1
rjmp int_right

int_right:
cpi r28, 0x0
breq activate_seiC
cpi r26, 0xff
breq brake_turn_right
ldi r27, 0b11000000
out portb, r27
```

```
mov r2, r27;
or r2, r31;
com r2
out portb, r2
rcall Delay_1sec

lsr r31
cpi r31, 0b0
breq resetL

sbis PIND, 0
rjmp int_right
reti

resetL:

mov r31, r23
rjmp int_right


brake_turn_left:
cli
ldi r28, 0xff
ldi r26, 0b0
ldi r27, 0b00001111
out portb, r27

mov r2, r27;
or r2, r30;
com r2
out portb, r2
rcall Delay_1sec

lsl r30
cpi r30, 0b0
breq resetL2


sbic PIND, 2
rjmp int_left

sbic PIND, 1
rjmp int_brake

rjmp brake_turn_left

resetL2:
mov r30, r24
rjmp brake_turn_left


brake_turn_right:
cli
ldi r26, 0b0
ldi r27, 0b11110000
out portb, r27


mov r2, r27;
or r2, r31;
com r2
```

```
out portb, r2
rcall Delay_1sec

lsr r31
cpi r31, 0b0
breq resetR2

sbic PIND, 2
rjmp int_right

sbic PIND, 0
rjmp int_brake

rjmp brake_turn_right

int_brake:
cpi r28, 0x0
breq activate_seiB
ldi r26, 0xff
com r26
out PORTB, r26
com r26

cpi r27, 0b11000000
breq brake_turn_right

cpi r27, 0b00000011
breq brake_turn_left

sbic PIND, 2
reti
rjmp int_brake


activate_seiB:
sei
ldi r28, 0x1
rjmp int_brake

resetR2:
mov r31, r23
rjmp brake_turn_right

int_left:
cpi r28, 0x0
breq activate_sei
cpi r26, 0xff
breq brake_turn_left
ldi r27, 0b00000011
out portb, r27

mov r2, r27;
or r2, r30;
com r2
out portb, r2
//rcall Delay_1sec

lsl r30
cpi r30, 0b0
breq resetR
```

```
sbis PIND, 1
rjmp int_left
reti
activate_sei:
ldi r28, 0x1
sei
rjmp int_left

resetR:
mov r30, r24
rjmp int_left



innit:
ldi r31, LOW(RAMEND)
out spl, r31
ldi r31, HIGH(RAMEND)
out sph, r31

LDI R16, 0b00101010 ; make INT0
STS EICRA, r16


ldi r19, 0xff

out DDRB, r19

SBI PORTD, 2 ; pull-up enabled, input pin PIND2
LDI R16, 1<<INT0 | 1<<INT1 | 1<<INT2
OUT EIMSK, R16

ldi r23, 0b00001000
ldi r24, 0b0010000
mov r31, r23
mov r30, r24
sei

test:
ldi r28, 0x0
ldi r27, 0b0
ldi r26, 0b1
ldi r22, 0b00111100
out PortB, r22

rjmp test



input_Delay:                    ; For CLK(CPU) = 1 MHz
    LDI     r16,    1       ; One clock cycle;
Delay1c:
    LDI     r17,    125     ; One clock cycle
Delay2c:
    LDI     r18,    250     ; One clock cycle
Delay3c:
    DEC     r18             ; One clock cycle
    NOP                      ; One clock cycle
```

```
    BRNE    Delay3c         ; Two clock cycles when jumping to Delay3, 1 clock when continu

    DEC     r17             ; One clock cycle
    BRNE    Delay2c           ; Two clock cycles when jumping to Delay2, 1 clock when conti

    DEC     r16             ; One clock Cycle
    BRNE    Delay1c           ; Two clock cycles when jumping to Delay1, 1 clock when conti
RET


Delay_1sec:                   ; For CLK(CPU) = 1 MHz
    LDI     r16,    4       ; One clock cycle;
Delay1:
    LDI     r17,    125     ; One clock cycle
Delay2:
    LDI     r18,    250     ; One clock cycle
Delay3:
    DEC     r18             ; One clock cycle
    NOP                      ; One clock cycle
    BRNE    Delay3          ; Two clock cycles when jumping to Delay3, 1 clock when continui

    DEC     r17             ; One clock cycle
    BRNE    Delay2           ; Two clock cycles when jumping to Delay2, 1 clock when contin

    DEC     r16             ; One clock Cycle
    BRNE    Delay1           ; Two clock cycles when jumping to Delay1, 1 clock when contin
RET
```


set up interrupt
LDI R16, 0b00101010
STS EICRA, r16
SBI PORTD, 2
LDI R16, 1<<INT0 | 1<<INT1 | 1<<INT2 OUT EIMSK, R16
ldi r23, 0b00001000
ldi r24, 0b0010000
mov r31, r23 mov r30, r24
sei
loop start
ldi r28, 0x0
ldi r27, 0b0
ldi r26, 0b1
ldi r22, 0b00111100
out PortB, r22
Int_right
r28 == 0x0?
sei and ldi r28, 0x1
mov r31, r23
is r26 == 0xff?
else
ldi r27, 0b11000000 out portb, r27
mov r2, r27; or r2, r31; com r2 out portb, r2 rcall Delay_1sec
lsr r31
is r31 == 0b0?
False
True
is PIND, 0 not pressed?
RETI
brake_turn_right
cli
ldi r26, 0b0 ldi r27, 0b11110000 out portb, r27
mov r2, r27 or r2, r31 com r2
out portb, r2
rcall Delay_1sec
lsr r31
is r31 == 0b072
is PIND, 2 not pressed?
is PIND, 0 not pressed?
mov r31, r23
Int_brake
is r28 == 0x0?
sei
ldi r28, 0x1
ldi r26, 0xff
com r26 out PORTB, r26 com r26
r27 == 0b11000000?
r27 == 0b00000011?
is PIND, 2 not being pressed?
RETI
Reference to delay from previous assigment
int_left
r28 == 0x0?
sei and ldi r28, 0x1
r26 = 0xFF?
ldi r27, 0b00000011 out portb, r27
mov r2, r27; or r2, r30; com r2
out portb, r2
rcall Delay_1sec
lsl r30
r30 == 0b0?
is PIND, 1 not pressed?
RETI
mov r30, r24
brake_turn_left
cli
ldi r28, 0xff ldi r26, 0b0 ldi r27, 0b00001111
out portb, r27
mov r2, r27; or r2, r30; com r2
out portb, r2
rcall Delay_1sec
lsl r30
r30 == 0b0?
Is PIND, 2 not pressed?
Is PIND, 1 not pressed?
mov r30, r24
true
False
True