



# Computer Technology

## Report for Lab 4



*Student:* Helena TEVAR HERNANDEZ

*Student ID:* ht222fd@student.lnu.se

*Student:* Jiahui LE

*Student ID:* jl224bn@student.lnu.se

*Semester:* HT2018

*Area:* Computer Technology

*Course code:* 1DV607

**Contents**

<b>1</b>	<b>Task 1</b>	<b>1</b>
1.1	Diagram . . . . .	1
1.2	Code . . . . .	2
<b>2</b>	<b>Task 2</b>	<b>5</b>
2.1	Diagram . . . . .	5
2.2	Code . . . . .	6
<b>3</b>	<b>Task 3 - 4</b>	<b>9</b>
3.1	Diagram . . . . .	9
3.2	Code . . . . .	11
<b>4</b>	<b>Task 5</b>	<b>13</b>
4.1	Diagram . . . . .	13
4.2	Code . . . . .	14

# 1 Task 1

Create a square wave, frequency 1Hz, duty 50%.

## 1.1 Diagram

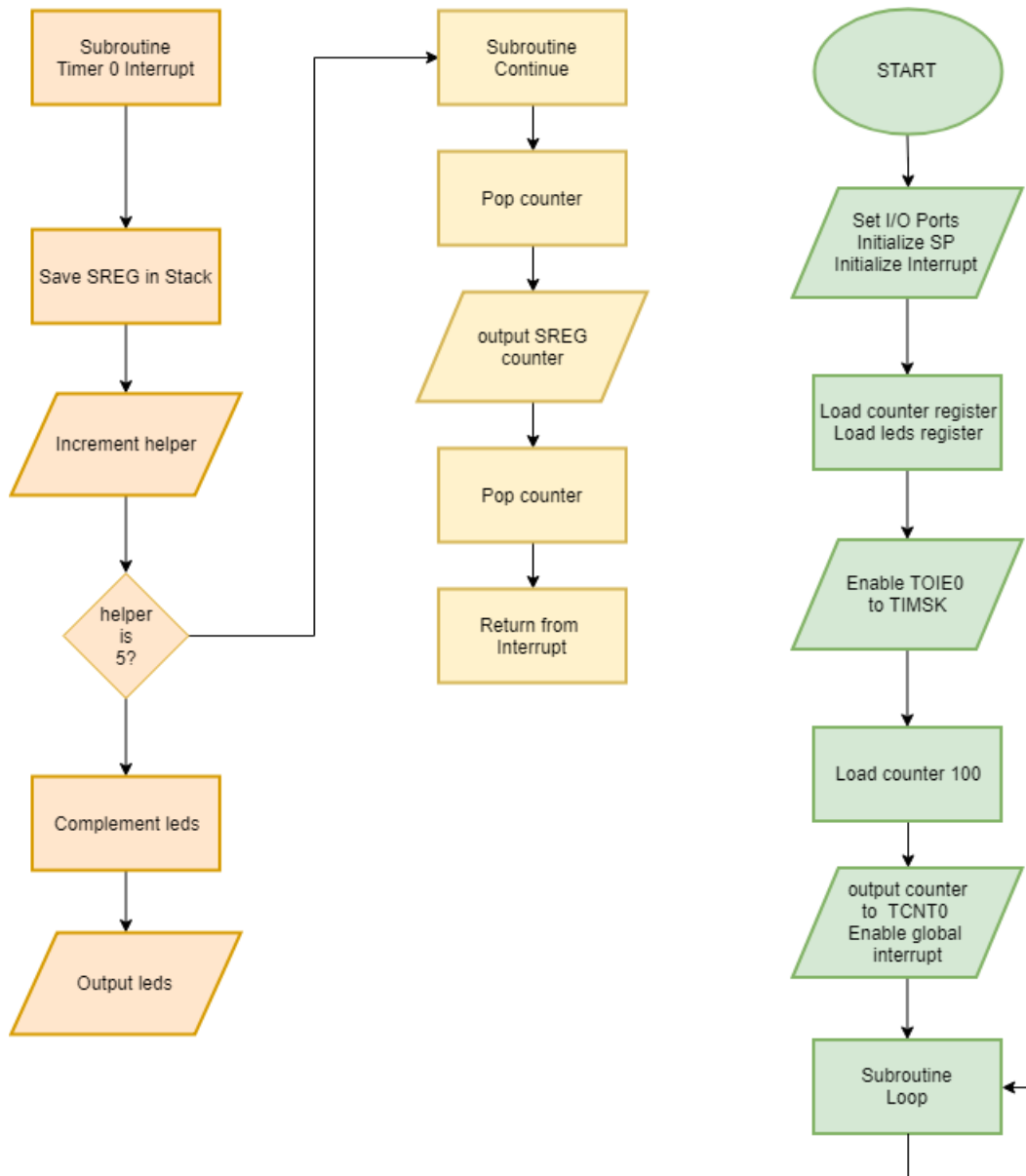


Figure 1: Task 1 diagram

## 1.2 Code

Listing 1: Task 1 code.

[illegible]



```

;_save_SREG_on_stack
;~~~~~
timer0int:
push_counter
in_counter,_SREG
push_counter

;~~~~~
;_reset_counter_value
;_increment_counter
;_check_if_"tick" is reached – when counter equals 5
; flip/invert
; push new state to PORTB
;~~~~~

ldi counter, 100
out TCNT0, counter
inc helper
cpi helper, 5
brne continue
ldi helper, 0
com led
out PORTB, led

;~~~~~
; restore SREG (overflow flag)
; restore register
; return from interrupt
;~~~~~
continue:
nop
pop counter
out SREG, counter
pop counter
reti

;~~~~~
; source: Slides from lecture 7
;~~~~~

```

## 2 Task 2

Modify Task 1 to make the duty change by 5% up and down.

### 2.1 Diagram

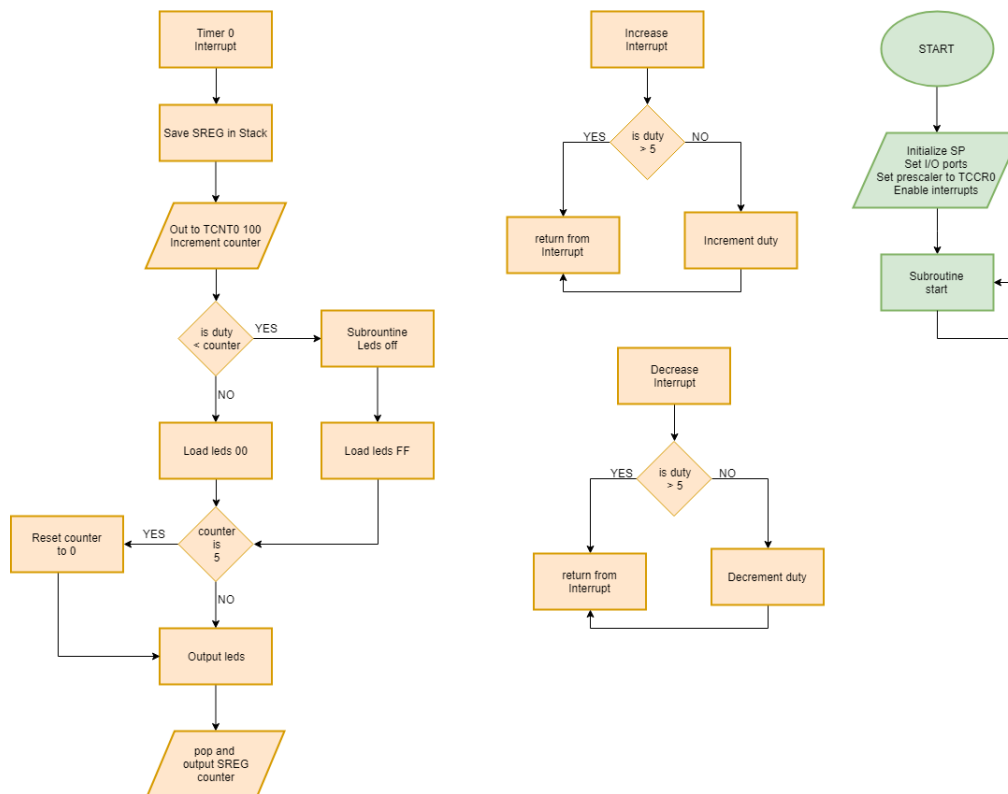


Figure 2: Task 2 diagram

## 2.2 Code

Listing 2: Task 2 code.

[illegible]



[illegible]

```

;Start loop
;The relative jump uses two cycles.;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
start:
;nop
rjmp start

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
; timer interrupt routine
; save SREG on stack
; reset counter value
; increment counter
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

timer0int:
push r16
in r16, SREG
push r16

ldi r16, 100
out TCNT0, r16

inc counter
cp duty_counter, counter
brlt led_off

ldi LED, 0x00
rjmp continue

led_off:
ldi LED, 0xFF

continue:
cpi counter, 5
brne continue2
ldi counter, 0

continue2:
out portB, LED
pop r16
out SREG, r16
pop r16
reti

increase:
cpi duty_counter, 5
brge after_inc
inc duty_counter

after_inc:
nop
reti

```

```
;
; source: Slides from lecture 7
;
```

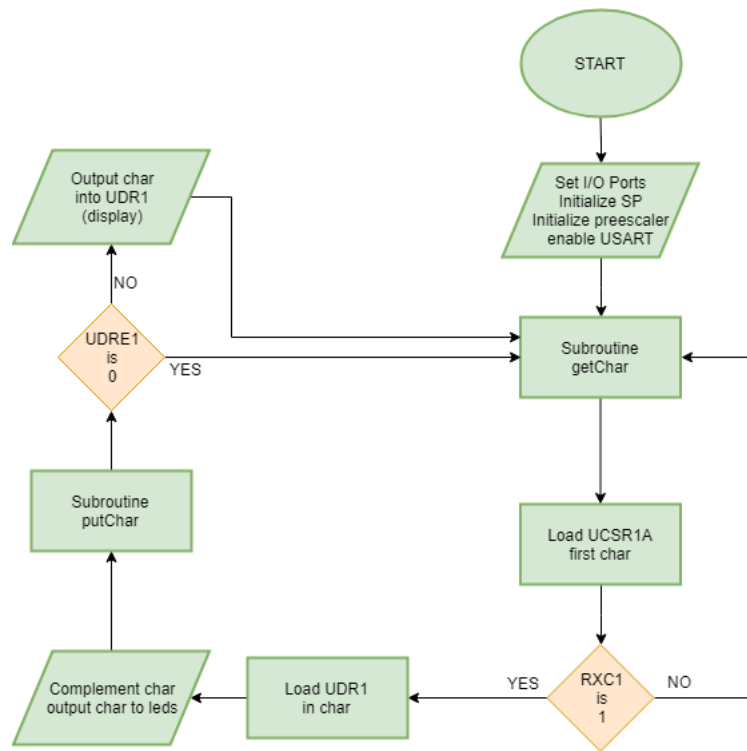


Figure 4: Task 4

### 3.2 Code

Listing 3: Task 3 code.

[illegible]

[illegible]

## 4 Task 5

Repeat task 4 with interrupt instead of UART.

### 4.1 Diagram

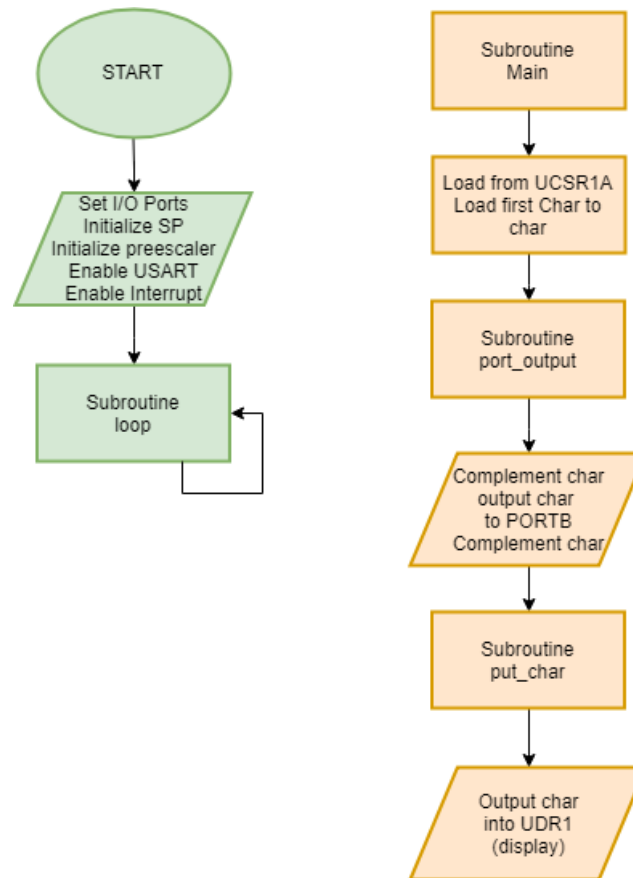


Figure 5: Task 4 Part 1 diagram

## 4.2 Code

Listing 4: Task 5 code.

[illegible]



```

out SPH ,r20
ldi r20 , LOW (RAMEND)
out SPL ,r20

ldi temp, 0xFF
out DDRB, temp
ldi temp, 0x55
out PORTB, temp

ldi temp, 12
sts UBRR1L, temp
ldi temp, 0b10011000
;ldi temp, (1<<TXEN1) | (1<<RXEN1)
sts UCSR1B, temp
sei

loop:
rjmp loop

main:
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
; read from USART to get character
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
get_char:
lds temp, UCSR1A
lds char, UDR1

port_output:
com char
out PORTB, char
com char

put_Char:
lds temp, UCSR1A
sts UDR1, char
reti

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
; source: Slides from lecture 7
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

```