

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

1	Task	1	1	
	1.1	Diagram		
	1.2	Code	2	
2	Task	x 2	5	
	2.1	Diagram	5	
	2.2	Code		
3	Task 3 - 4			
	3.1	Diagram	9	
		Code		
4	Task 5			
	4.1	Diagram	3	
	4.2	Code	4	

1 Task 1

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

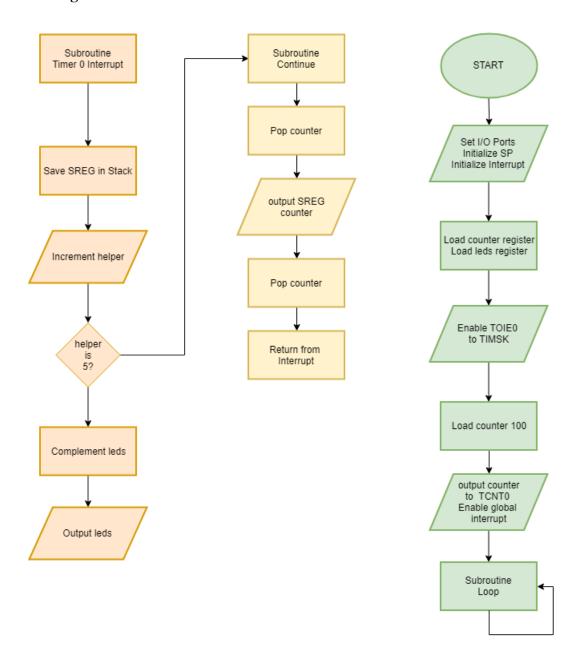


Figure 1: Task 1 diagram

Listing 1: Task 1 code.

```
; 1DT301, Computer Technology I
; Date: 2018-10-08
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
; Lab number: 4
; Title: Square wave generator
; Hardware: STK600, CPU ATmega2560
; Function: Create a square wave
      frequency 1 Hz duty 50%
; Input ports:
; Output ports: Port B
; Subroutines: delay
; Main
; changeLeds
; Included files: m2560def.inc
; Other information:
; Changes in program: 1/10/2018 – "programmed_and_implemented_in_board"
.include "m2560def.inc"
.def counter = counter
.def led = led
.def helper = helper
.org 0x00
jmp restart
.org OVF0addr ; address for Timer/Counter0 Overflow interrupt
jmp timer0int
.org 0x72
: Start subroutine
; Initialize SP, Stack Pointer
; setting output port
; setting timer
; load led values
```

```
; Global interrupt enabled
restart:
ldi r20, high (RAMEND)
out SPH, r20
ldi r20, low (RAMEND)
out SPL, r20
; set data direction registers.
; set B port as output ports
ldi counter, 0x01
out DDRB, counter
ldi led, 0x00
out PORTB, led
; setting up prescaler value to TCCR0 ""get slower""
; CS2, -, CS0, = 101, osc.clock / 1024, ->, timer counts
; every ms. (1000 times / second)
ldi counter, 0x05
out_TCCR0B,_counter
;_timer_0_enable_flag,_TOIE0_to_register_TIMSK
ldi_counter,_(1<<TOIE0)
sts, TIMSKO, counter
; starting value for counter
;_counter-from_register
; enable global interrupt
;_help, counter
ldi_counter,_100
out_TCNT0_,_counter
sei
ldi helper, 0
;Start_loop
start:
rjmp_start
;_timer_interrupt_routine
```

```
;_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.

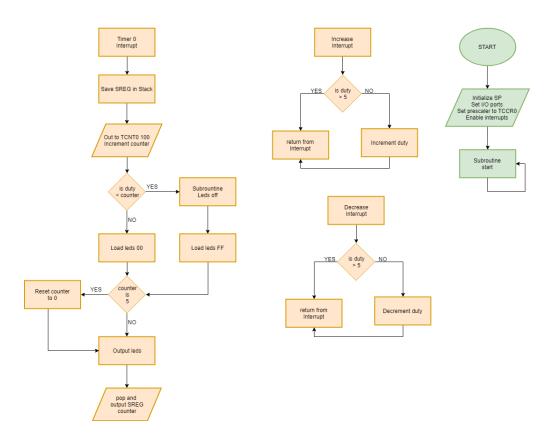


Figure 2: Task 2 diagram

Listing 2: Task 2 code.

```
; 1DT301, Computer Technology I
; Date: 2018-10-08
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
; Lab number: 4
; Title: Pulse width modulation (PWM)
; Hardware: STK600, CPU ATmega2560
; Function: Modify Task 1 to make the duty
      change by 5% up and down
; Input ports:
; Output ports: Port B
; Subroutines: timer0int
      led_off
      continue
      continue2
      increase
      after_inc
      decrease
      after dec
; Included files: m2560def.inc
; Other information:
; Changes in program: 1/10/2018 - "programmed_and_implemented_in_board"
.include "m2560def.inc"
.org 0x00
rjmp restart
.org OVF0addr
rjmp timer0int
.org INT3addr
rjmp increase
.org INT1addr
rjmp decrease
.def LED = r17
.def counter = r18
```

```
.def duty\_counter = r19
.org 0x72
; Start subroutine
; Initialize SP, Stack Pointer
; set data direction registers.
; set B port as output ports
; setting up prescaler value to TCCR0
; CS2 - CS2 = 101, osc.clock / 1024 -> timer counts every ms. (1000 times / second)
; timer 0 enable flag, TOIE0EIMSK 0 to register TIMSK
; starting value for counter
; counter register
; INTO and INT1 enabled
; falling and rising edge
; enable global interrupt
restart:
ldi r20, high (RAMEND)
out SPH, r20
ldi r20, low (RAMEND)
out SPL, r20
ldi r16, 0x01
out DDRB, r16
ldi LED, 0x00
out PORTB, LED
ldi r16, 0x04
out TCCR0B, r16
ldi r16, (1<<TOIE0)
sts TIMSK0, r16
ldi r16, 205
out TCNT0, r16
ldi r16, 0b0000_1010
out EIMSK, r16
ldi r16, 0b1100_1100
sts EICRA, r16
sei
ldi counter, 0
ldi duty_counter, 10
```

```
;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
```

3 Task 3 - 4

Read a character from terminal and show its code in leds. Echo the character back to terminal.

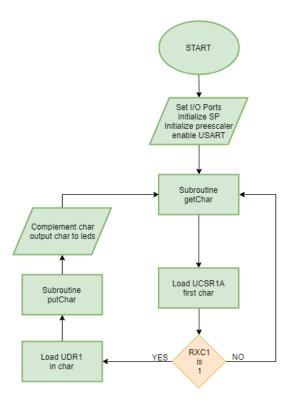


Figure 3: Task 3

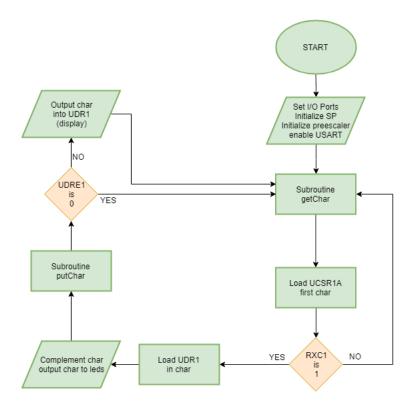


Figure 4: Task 4

Listing 3: Task 3 code.

```
; 1DT301, Computer Technology I
; Date: 2018-10-08
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
; Lab number: 4
; Title: Serial Communication with echo
; Hardware: STK600, CPU ATmega2560
; Function: Task 3
; Read a character from terminal
      show its code in leds
: Task 4
; Echo the character back to terminal
; Input ports:
; Output ports: Port B
; Subroutines: get_char
; putChar
: Included files: m2560def.inc
: Other information:
; Changes in program: 1/10/2018 — "programmed_and_implemented_in_board"
.include "m2560def.inc"
.def temp = r17
.def char = r16
.org 0x00
rjmp start
.org 0x72
; Start subroutine
; Initialize SP, Stack Pointer
; set data direction registers.
; set B port as output ports
```

```
; setting up prescaler value to TCCR0
; store Prescaler val in UBRR1L; 4800 as speed (osc.=1MHz, 4800 bps => UBBRR = 12)
; connect cable to pin 2/3 on Port D
; enable USART transmitter (set TX and RX enable flags)
start:
ldi temp, 0xFF
out DDRB, temp
out PORTB, temp
ldi temp, 12
sts UBRR1L, temp
ldi temp, (1<<TXEN1) | (1<<RXEN1)
sts UCSR1B, temp
main:
; read from USART to get character
; new character, RXC1=1
; no char received RXC1=0
; read char in UDR
get_char:
lds temp, UCSR1A
sbrs temp, RXC1
rjmp get_char
lds char, UDR1
com char
out PORTB, char
com char
;Task 4
putChar:
lds temp, UCSR1A
sbrs temp, UDRE1
rjmp putChar
sts UDR1, char
rjmp get_char
; source: Slides from lecture 7
```

4 Task 5

Repeat task 4 with interrupt instead of UART.

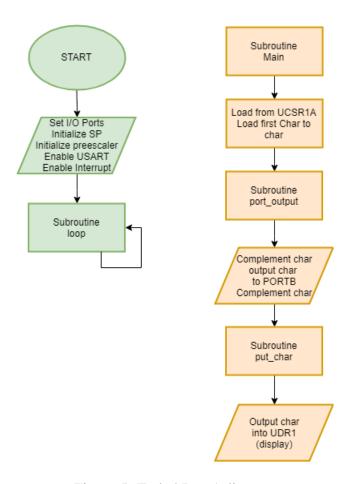


Figure 5: Task 4 Part 1 diagram

Listing 4: Task 5 code.

```
; 1DT301, Computer Technology I
; Date: 2018-10-08
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
; Lab number: 4
; Title: Serial communication using interrupt
; Hardware: STK600, CPU ATmega2560
; Function: Repeat Task 4 with interrupt instead of UART
; Input ports:
; Output ports: Port B
: Subroutines:
; Included files: m2560def.inc
: Other information:
; Changes in program: 1/10/2018 — "programmed_and_implemented_in_board"
.include "m2560def.inc"
.def temp = r17
.def char = r16
.org 0x00
rjmp start
.org URXC1addr
rjmp main
.org 0x72
; Start subroutine
; Initialize SP, Stack Pointer
; set B port as output ports
; store Prescaler val in UBRR1L; 4800 as speed
(osc.=1MHz, 4800 bps => UBBRR = 12)
; connect cable to pin 2/3 on Port D
; RXCIE =1, RXEN=1, TXEN=1
; enable USART transmitter (set TX and RX enable flags)
start:
ldi r20, HIGH (RAMEND)
```

```
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
```