Computer Technology

# Report for Lab 5

*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

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

### 1.1 Diagram

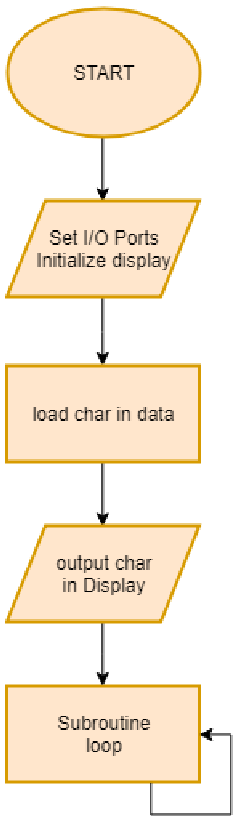

Figure 1: Task 1 diagram

## 1.2 Code

Listing 1: Task 1 code.

```
; 1DT301
; Lab 5, Initialize display JHD202A.
;
; Date: 2016-09-30
; Author, Anders Haggren
; Modified:
;
; Function
; --------
; Initialize display JHD202 connected to PORTE
;
; (run @ 1.8432 MHz clk frequency)
;
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018-10-22
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
;
; Lab number: 5
; Title: Write a program that displays a character on the display.
;
; Hardware: STK600, CPU ATmega2560
;
; Function:
; Input ports:
;
; Output ports:
; Subroutines:
;
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: 22/10/2018 - "programmed and implemented in board"
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
.include "m2560def.inc"
.def Temp = r16
.def Data = r17
.def RS = r18

.equ BITMODE4 = 0b00000010  ; 4-bit operation
.equ CLEAR = 0b00000001   ; Clear display
.equ DISPCTRL = 0b00001111  ; Display on, cursor on, blink on.

.cseg
.org 0x0000   ; Reset vector
```

```
            jmp reset

.org 0x0072

reset:

            ldi Temp, HIGH(RAMEND) ; Temp = high byte of ramend address
            out SPH, Temp   ; sph = Temp
            ldi Temp, LOW(RAMEND) ; Temp = low byte of ramend address
            out SPL, Temp   ; spl = Temp

            ser Temp    ; r16 = 0b11111111
            out DDRE, Temp   ; port E = outputs ( Display JHD202A)
            clr Temp    ; r16 = 0
            out PORTE, Temp

            rcall init_disp
            ldi Data, 0b00100101
            rcall write_char

loop: nop
            rjmp loop   ; loop forever

; **
; ** init_display
; **
init_disp:
            rcall power_up_wait  ; wait for display to power up

            ldi Data, BITMODE4  ; 4−bit operation
            rcall write_nibble  ; (in 8−bit mode)
            rcall short_wait  ; wait min. 39 us
            ldi Data, DISPCTRL  ; disp. on, blink on, curs. On
            rcall write_cmd   ; send command
            rcall short_wait  ; wait min. 39 us
clr_disp:
            ldi Data, CLEAR   ; clr display
            rcall write_cmd   ; send command
            rcall long_wait   ; wait min. 1.53 ms
            ret

; **
; ** write char/command
; **

write_char:
            ldi RS, 0b00100000  ; RS = high
            rjmp write
write_cmd:
            clr RS     ; RS = low
write:
            mov Temp, Data   ; copy Data
            andi Data, 0b11110000 ; mask out high nibble
            swap Data    ; swap nibbles
            or Data, RS    ; add register select
            rcall write_nibble  ; send high nibble
            mov Data, Temp   ; restore Data
            andi Data, 0b00001111 ; mask out low nibble
            or Data, RS    ; add register select

write_nibble:
            rcall switch_output  ; Modify for display JHD202A, port E
            nop      ; wait 542nS
            sbi PORTE, 5   ; enable high, JHD202A
            nop
            nop      ; wait 542nS
            cbi PORTE, 5   ; enable low, JHD202A
            nop
            nop      ; wait 542nS
            ret

; **
; ** busy_wait loop
; **
short_wait:
            clr zh     ; approx 50 us
            ldi zl, 30
            rjmp wait_loop
long_wait:
            ldi zh, HIGH(1000)  ; approx 2 ms
            ldi zl, LOW(1000)
            rjmp wait_loop
dbnc_wait:
            ldi zh, HIGH(4600)  ; approx 10 ms
            ldi zl, LOW(4600)
            rjmp wait_loop
power_up_wait:
            ldi zh, HIGH(9000)  ; approx 20 ms
            ldi zl, LOW(9000)

wait_loop:
            sbiw z, 1    ; 2 cycles
            brne wait_loop   ; 2 cycles
            ret

; **
; ** modify output signal to fit LCD JHD202A, connected to port E
; **

switch_output:
            push Temp
            clr Temp
            sbrc Data, 0    ; D4 = 1?
            ori Temp, 0b00000100  ; Set pin 2
```

```
sbrc Data, 1    ; D5 = 1?
ori Temp, 0b00001000  ; Set pin 3
sbrc Data, 2    ; D6 = 1?
ori Temp, 0b00000001  ; Set pin 0
sbrc Data, 3    ; D7 = 1?
ori Temp, 0b00000010  ; Set pin 1
sbrc Data, 4    ; E = 1?
ori Temp, 0b00100000  ; Set pin 5
sbrc Data, 5    ; RS = 1?
ori Temp, 0b10000000  ; Set pin 7 (wrong in previous version)
out porte, Temp
pop Temp
ret
```

## 2 Task 2

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

### 2.1 Diagram

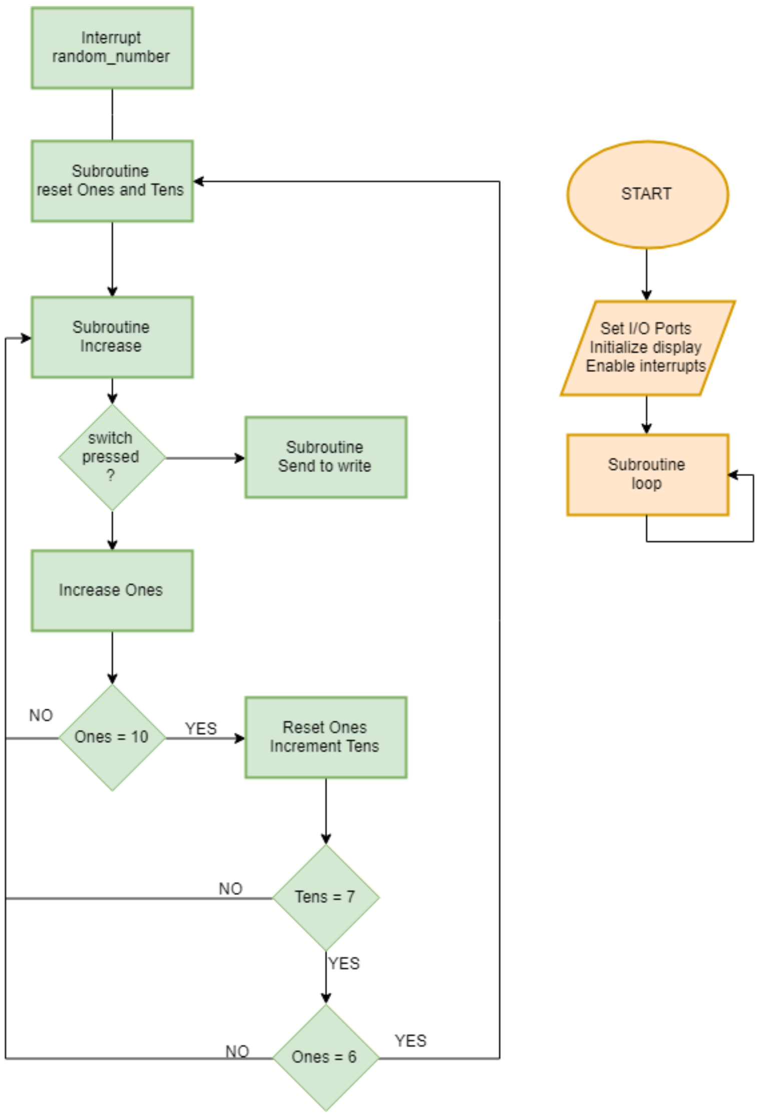

Figure 2: Task 2 diagram

## 2.2 Code

Listing 2: Task 2 code.

```
; 1DT301
; Lab 5, Initialize display JHD202A.
;
; Date: 2016-09-30
; Author, Anders Haggren
; Modified:
;
; Function
; ---------
; Initialize display JHD202 connected to PORTE
;
; (run @ 1.8432 MHz clk frequency)
;
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018-10-22
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
;
; Lab number: 5
; Title: Electronic Bingo Device
;
; Hardware: STK600, CPU ATmega2560
;
; Function:
; Input ports:
;
; Output ports:
; Subroutines:
;
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: 22/10/2018 - "programmed␣and␣implemented␣in␣board"
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include  "m2560def.inc"
.def Temp = r16
.def Data = r17
.def RS = r18
.def Tens = r19
.def Ones = r20

.equ BITMODE4 = 0b00000010  ; 4-bit operation
.equ CLEAR = 0b00000001   ; Clear display
.equ DISPCTRL = 0b00001111  ; Display on, cursor on, blink on.
```

```
.cseg
.org 0x0000    ; Reset vector
            jmp reset
.org INT1addr
rjmp random_number

.org 0x0072

reset:

            ldi Temp, HIGH(RAMEND) ; Temp = high byte of ramend address
            out SPH, Temp   ; sph = Temp
            ldi Temp, LOW(RAMEND) ; Temp = low byte of ramend address
            out SPL, Temp   ; spl = Temp

            ser Temp    ; r16 = 0b11111111
            out DDRE, Temp   ; port E = outputs ( Display JHD202A)
            clr Temp    ; r16 = 0
            out PORTE, Temp

            ldi Temp, 0b0000_0010
            out EIMSK, Temp
            ldi Temp, 0b0000_1000
            sts EICRA, Temp
            sei

rcall init_disp

loop: nop
            rjmp loop   ; loop forever
; **
; ** init_display
; **
init_disp:
            rcall power_up_wait  ; wait for display to power up

            ldi Data, BITMODE4  ; 4−bit operation
            rcall write_nibble  ; (in 8−bit mode)
            rcall short_wait  ; wait min. 39 us
            ldi Data, DISPCTRL  ; disp. on, blink on, curs. On
            rcall write_cmd   ; send command
            rcall short_wait  ; wait min. 39 us



clr_disp:
            ldi Data, CLEAR   ; clr display
            rcall write_cmd   ; send command
            rcall long_wait   ; wait min. 1.53 ms
            ret
```

```
; **
; ** write char/command
; **

write_char:
            ldi RS, 0b00100000  ; RS = high
            rjmp write
write_cmd:
            clr RS     ; RS = low
write:
            mov Temp, Data   ; copy Data
            andi Data, 0b11110000 ; mask out high nibble
            swap Data    ; swap nibbles
            or Data, RS    ; add register select
            rcall write_nibble  ; send high nibble
            mov Data, Temp   ; restore Data
            andi Data, 0b00001111 ; mask out low nibble
            or Data, RS    ; add register select

write_nibble:
            rcall switch_output  ; Modify for display JHD202A, port E
            nop      ; wait 542nS
            sbi PORTE, 5   ; enable high, JHD202A
            nop
            nop      ; wait 542nS
            cbi PORTE, 5   ; enable low, JHD202A
            nop
            nop      ; wait 542nS
            ret

; **
; ** busy_wait loop
; **
short_wait:
            clr zh     ; approx 50 us
            ldi zl, 30
            rjmp wait_loop
long_wait:
            ldi zh, HIGH(1000)  ; approx 2 ms
            ldi zl, LOW(1000)
            rjmp wait_loop
dbnc_wait:
            ldi zh, HIGH(4600)  ; approx 10 ms
            ldi zl, LOW(4600)
            rjmp wait_loop
power_up_wait:
            ldi zh, HIGH(9000)  ; approx 20 ms
            ldi zl, LOW(9000)

wait_loop:
            sbiw z, 1    ; 2 cycles
            brne wait_loop   ; 2 cycles
            ret

; **
; ** modify output signal to fit LCD JHD202A, connected to port E
; **

switch_output:
            push Temp
            clr Temp
            sbrc Data, 0    ; D4 = 1?
            ori Temp, 0b00000100  ; Set pin 2
            sbrc Data, 1    ; D5 = 1?
            ori Temp, 0b00001000  ; Set pin 3
            sbrc Data, 2    ; D6 = 1?
            ori Temp, 0b00000001  ; Set pin 0
            sbrc Data, 3    ; D7 = 1?
            ori Temp, 0b00000010  ; Set pin 1
            sbrc Data, 4    ; E = 1?
            ori Temp, 0b00100000  ; Set pin 5
            sbrc Data, 5    ; RS = 1?
            ori Temp, 0b10000000  ; Set pin 7 (wrong in previous version)
            out porte, Temp
            pop Temp
            ret

random_number:

reset_number:
ldi Tens, 0
ldi Ones, 1

increase:
in Temp,PIND
cpi Temp,0xFF
breq sendToWrite

inc Ones
cpi Ones, 10
brne check_max

inc Tens
clr Ones

check_max:
cpi Tens, 7
brne increase
cpi Ones, 6
breq reset_number
rjmp increase

sendToWrite:
rcall clr_disp
rcall short_wait

mov Data, Tens
ldi Temp, 0b00110000
add Data, Temp
rcall write_char
rcall short_wait

mov Data, Ones
ldi Temp, 0b00110000
add Data, Temp
rcall write_char
rcall short_wait

reti
```

## 3 Task 3 - 4

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

## 3.1 Diagram

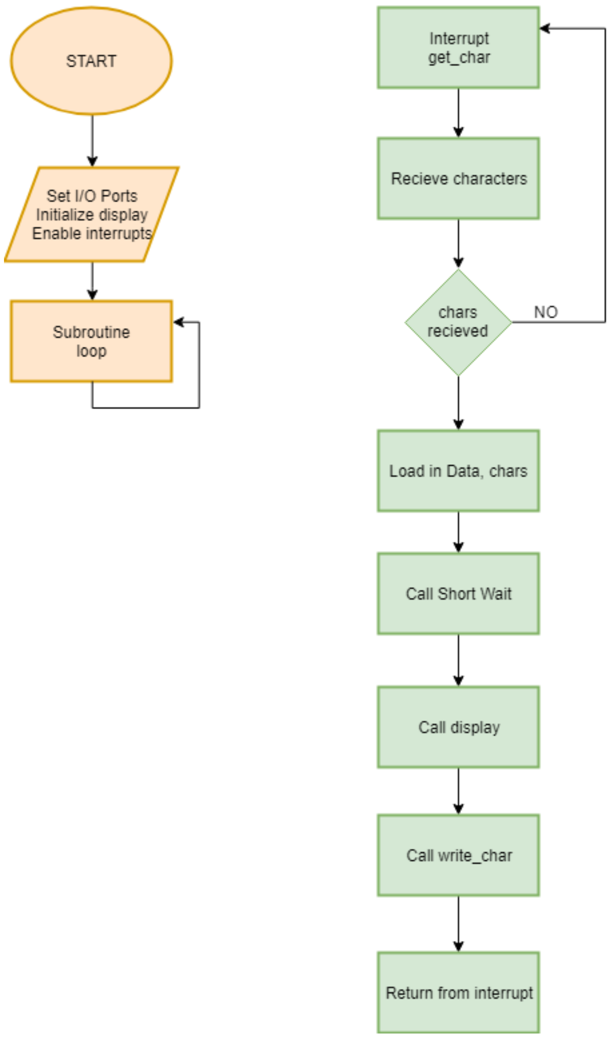

Figure 3: Task 3

## 3.2 Code

Listing 3: Task 3 code.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018−10−22
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
;
; Lab number: 5
; Title: Serial communication and display
;
; Hardware: STK600, CPU ATmega2560
;
; Function:
; Input ports:
;
; Output ports:
; Subroutines:
;
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: 22/10/2018 − "programmed␣and␣implemented␣in␣board"
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"
.def Temp = r16
.def Data = r17
.def RS = r18


.equ BITMODE4 = 0b00000010 ; 4−bit operation
.equ CLEAR = 0b00000001 ; Clear display
.equ DISPCTRL = 0b00001111 ; Display on, cursor on, blink on.




.cseg
.org 0x0000 ; Reset vector
jmp reset

.org URXC1addr ; interrupt address for USART
rjmp get_char

.org 0x0072

reset:
        ldi Temp, HIGH(RAMEND) ; Temp = high byte of ramend address
        out SPH, Temp ; sph = Temp
        ldi Temp, LOW(RAMEND) ; Temp = low byte of ramend address
        out SPL, Temp ; spl = Temp

        ser Temp ; r16 = 0b11111111
        out DDRE, Temp ; port E = outputs ( Display JHD202A)
        clr Temp ; r16 = 0
        out PORTE, Temp

        ldi Temp, 12
        sts UBRR1L, Temp

        ldi Temp, (1<<TXEN1) | (1<<RXEN1) | (1<<RXCIE1) ; enable interrupt in USART
        sts UCSR1B, Temp


                sei
                rcall init_disp
loop:
        nop
        rjmp loop ; loop forever


; **
; ** init_display
; **
init_disp:
        rcall power_up_wait ; wait for display to power up

        ldi Data, BITMODE4 ; 4-bit operation
        rcall write_nibble ; (in 8-bit mode)
        rcall short_wait ; wait min. 39 us
        ldi Data, DISPCTRL ; disp. on, blink on, curs. On
        rcall write_cmd ; send command
        rcall short_wait ; wait min. 39 us

clr_disp:
        ldi Data, CLEAR ; clr display
        rcall write_cmd ; send command
        rcall long_wait ; wait min. 1.53 ms
        ret




; **
; ** write char/command
; **

write_char:
        ldi RS, 0b00100000 ; RS = high
        rjmp write
```

```
write_cmd:
    clr RS ; RS = low
write:
    mov Temp, Data ; copy Data
    andi Data, 0b11110000 ; mask out high nibble
    swap Data ; swap nibbles
    or Data, RS ; add register select
    rcall write_nibble ; send high nibble
    mov Data, Temp ; restore Data
    andi Data, 0b00001111 ; mask out low nibble
    or Data, RS ; add register select

write_nibble:
    rcall switch_output ; Modify for display JHD202A, port E
    nop ; wait 542nS
    sbi PORTE, 5 ; enable high, JHD202A
    nop
    nop ; wait 542nS
    cbi PORTE, 5 ; enable low, JHD202A
    nop
    nop ; wait 542nS
    ret

; **
; ** busy_wait loop
; **
short_wait:
    clr zh ; approx 50 us
    ldi zl, 30
    rjmp wait_loop
long_wait:
    ldi zh, HIGH(1000) ; approx 2 ms
    ldi zl, LOW(1000)
    rjmp wait_loop
dbnc_wait:
    ldi zh, HIGH(4600) ; approx 10 ms
    ldi zl, LOW(4600)
    rjmp wait_loop
power_up_wait:
    ldi zh, HIGH(9000) ; approx 20 ms
    ldi zl, LOW(9000)

wait_loop:
    sbiw z, 1 ; 2 cycles
    brne wait_loop ; 2 cycles
    ret

; **
; ** modify output signal to fit LCD JHD202A, connected to port E
; **
switch_output:
    push Temp
```

```
        clr Temp
        sbrc Data, 0 ; D4 = 1?
        ori Temp, 0b00000100 ; Set pin 2
        sbrc Data, 1 ; D5 = 1?
        ori Temp, 0b00001000 ; Set pin 3
        sbrc Data, 2 ; D6 = 1?
        ori Temp, 0b00000001 ; Set pin 0
        sbrc Data, 3 ; D7 = 1?
        ori Temp, 0b00000010 ; Set pin 1
        sbrc Data, 4 ; E = 1?
        ori Temp, 0b00100000 ; Set pin 5
        sbrc Data, 5 ; RS = 1?
        ori Temp, 0b10000000 ; Set pin 7 (wrong in previous version)
        out porte, Temp
        pop Temp
        ret

get_char:
        lds Temp, UCSR1A ; interrupt on Serial Port
        sbrs Temp, RXC1 ; wait until characters are received
        rjmp get_char

        lds Data, UDR1 ; read character from Serial Port

        rcall short_wait
        rcall display
        reti

display:
        rcall write_char ; write the character to display.
        ret
```

## 4 Task 5

Repeat task 4 with interrupt instead of UART.

## 4.1 Diagram

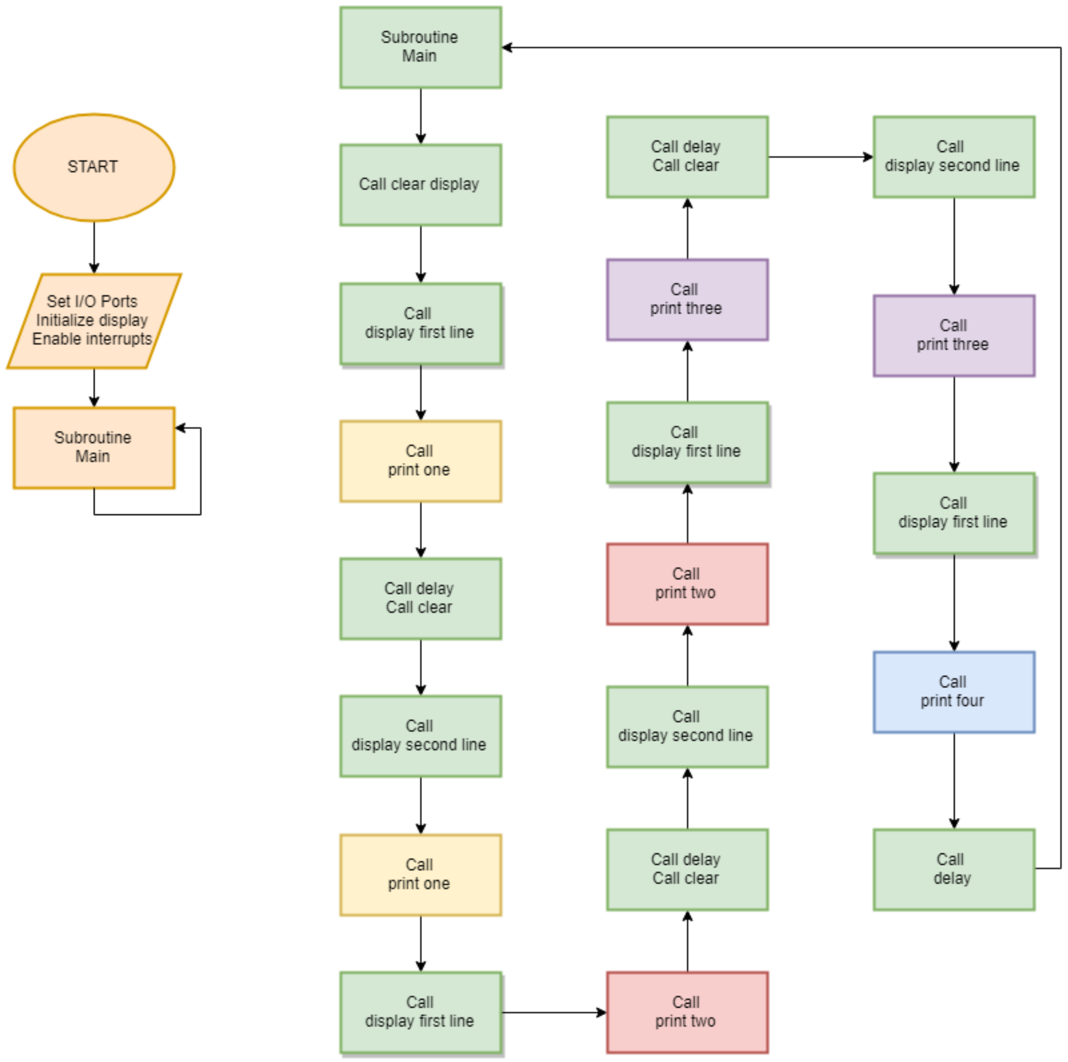

Figure 4: Task 4 Part 1 diagram

## 4.2 Code

Listing 4: Task 5 code.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018−10−22
; Author:
; Jiahui Le (jl224bn)
; Helena Tevar (ht222fd)
;
; Lab number: 5
; Title: Modify the program in task 3 so that 4 lines of text can be displayed.
;
; Hardware: STK600, CPU ATmega2560
;
; Function:
; Input ports:
;
; Output ports:
; Subroutines:
;
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: 22/10/2018 − "programmed␣and␣implemented␣in␣board"
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
.include "m2560def.inc"

.equ BITMODE4 = 0b00000010 ; 4−bit operation
.equ CLEAR = 0b00000001 ; Clear display
.equ DISPCTRL = 0b00001111 ; Display on, cursor on, blink on. //DISP_CTRL
.equ MEMORY = 0x600

.def Temp = r16
.def Data = r17
.def RS = r18
.def Temp2 = r19
.def counter = r20


.cseg
.org 0x0000 ; Reset vector
jmp reset

.org URXC1addr
rjmp get_char


.org 0x0072
```

```
reset :
; Init stack pointer
ldi Temp, HIGH(RAMEND)
out SPH, Temp
ldi Temp, LOW(RAMEND)
out SPL, Temp

ser Temp
out ddre, Temp
clr Temp
out porte, Temp

ldi Temp2, 12
sts UBRR1L, Temp2
ldi Temp2, (1<<TXEN1) | (1<<RXEN1)
sts UCSR1B, Temp2


ldi XH, HIGH(MEMORY)
ldi XL, LOW(MEMORY)

rcall init_disp



get_char:
lds Temp, UCSR1A ; interrupt on Serial Port
lds Data, UCSR1A
sbrs Temp, RXC1 ; wait until characters are received
rjmp get_char
lds Temp, UDR1

put_char:
mov Data, Temp
sbrs Data, UDRE1
rjmp put_char
st X+, Data
sts UDR1, Temp

inc counter
cpi counter, 140
brne get_char

main:
rcall clr_disp
rcall firstline
rcall print_one
rcall delay

rcall clr_disp
rcall secondline
rcall print_one
rcall firstline
rcall print_two
rcall delay

rcall clr_disp
rcall secondline
rcall print_two
rcall firstline
rcall print_three
rcall delay

rcall clr_disp
rcall secondline
rcall print_three
rcall firstline
rcall print_four
rcall delay

rjmp main
ret

print_one:
init1:
ldi counter, 0
ldi XH, HIGH(MEMORY)
ldi XL, LOW(MEMORY)
print1:
ld Data, X+
rcall write_char
rcall long_wait
inc counter
cpi counter, 40
brne print1
ret

print_two:
init2:
ldi counter, 0
ldi XH, HIGH(MEMORY+40)
ldi XL, LOW(MEMORY+40)
print2:
ld Data, X+
rcall write_char
rcall long_wait
inc counter
cpi counter, 40
brne print2
ret

print_three:
init3:
ldi counter, 0
ldi XH, HIGH(MEMORY+80)
ldi XL, LOW(MEMORY+80)
print3:
ld Data, X+
rcall write_char
rcall long_wait
inc counter
cpi counter, 40
brne print3
ret

print_four:
init4:
ldi counter, 0
ldi XH, HIGH(MEMORY+120)
ldi XL, LOW(MEMORY+120)
print4:
ld data, X+
rcall write_char
rcall long_wait
inc counter
cpi counter, 40
brne print4
ret


firstline:
ldi Data, 128
rcall write_cmd
rcall short_wait
ldi counter, 0x00
ret

secondline:
ldi Data, 168
rcall write_cmd
rcall short_wait
ldi counter, 0x00
ret

; Generated by delay loop calculator
; at http://www.bretmulvey.com/avrdelay.html
;
; Delay 5 000 000 cycles
; 5s at 1.0 MHz
delay:
    push r21
        push r22
        push r23

    ldi r21, 26
    ldi r22, 94
    ldi r23, 111
L1: dec r23
    brne L1
    dec r22
    brne L1
    dec r21
    brne L1

        pop r23
        pop r22
    pop r21
ret



init_disp:
rcall power_up_wait ; wait for display to power up
ldi Data, BITMODE4 ; 4−bit operation
rcall write_nibble ; (in 8−bit mode)
rcall short_wait ; wait min. 39 us
ldi Data, DISPCTRL ; disp. on, blink on, curs. On
rcall write_cmd ; send command
rcall short_wait ; wait min. 39 us

rcall clr_disp



clr_disp:
    ldi Data, CLEAR ; clr display
    rcall write_cmd ; send command
    rcall long_wait ; wait min. 1.53 ms
    ret

; **
; ** write char/command
; **

write_char:
    ldi RS, 0b00100000 ; RS = high
    rjmp write
write_cmd:
    clr RS ; RS = low
write:
    mov Temp, Data ; copy Data
    andi Data, 0b11110000 ; mask out high nibble
    swap Data ; swap nibbles
    or Data, RS ; add register select
    rcall write_nibble ; send high nibble
    mov Data, Temp ; restore Data
    andi Data, 0b00001111 ; mask out low nibble
    or Data, RS ; add register select
```

```
write_nibble:
      rcall switch_output ; Modify for display JHD202A, port E
      nop ; wait 542nS
      sbi PORTE, 5 ; enable high, JHD202A
      nop
      nop ; wait 542nS
      cbi PORTE, 5 ; enable low, JHD202A
      nop
      nop ; wait 542nS
      ret

; **
; ** busy_wait loop
; **
short_wait:
      clr zh ; approx 50 us
      ldi zl, 30
      rjmp wait_loop
long_wait:
      ldi zh, HIGH(1000) ; approx 2 ms
      ldi zl, LOW(1000)
      rjmp wait_loop
dbnc_wait:
      ldi zh, HIGH(4600) ; approx 10 ms
      ldi zl, LOW(4600)
      rjmp wait_loop
power_up_wait:
      ldi zh, HIGH(9000) ; approx 20 ms
      ldi zl, LOW(9000)

wait_loop:
      sbiw z, 1 ; 2 cycles
      brne wait_loop ; 2 cycles
      ret

; **
; ** modify output signal to fit LCD JHD202A, connected to port E
; **
switch_output:
      push Temp
      clr Temp
      sbrc Data, 0 ; D4 = 1?
      ori Temp, 0b00000100 ; Set pin 2
      sbrc Data, 1 ; D5 = 1?
      ori Temp, 0b00001000 ; Set pin 3
      sbrc Data, 2 ; D6 = 1?
      ori Temp, 0b00000001 ; Set pin 0
      sbrc Data, 3 ; D7 = 1?
      ori Temp, 0b00000010 ; Set pin 1
      sbrc Data, 4 ; E = 1?
      ori Temp, 0b00100000 ; Set pin 5
```

```
    sbrc Data, 5 ; RS = 1?
    ori Temp, 0b10000000 ; Set pin 7 (wrong in previous version)
    out porte, Temp
    pop Temp
    ret
```