```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;1DT301, Computer Technology I
;Date: 2019-10-09
;
;Author:
;   Student name 1:     Einar van de Velde
;   Student name 2:     Abdulla Mehdi
;
;Lab number:            5.
;Title:                 Task 3.
;
;Hardware:              STK600, CPU ATmega2560.
;
;Function:              Display serial (URAT1) input on display
;
;Input ports:           URAT1.
;
;Output ports:          PORTE.
;
;Subroutines:           None.
;
;Included files:        m2560def.inc
;
;Other information:
;Changes in program:
;       File Created (2019-10-20)
;                Program is runnable (2019-10-20)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
.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.

.equ UBRR_VALUE = 12
.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

        ldi Temp, 0xff
        out ddrb, Temp
        ldi Temp, 0x55
        out PORTB, Temp
        ldi Temp, UBRR_VALUE
        sts UBRR1L, Temp
        ldi Temp, (1<<TXEN1)|(1<<RXEN1)
        sts UCSR1B, Temp

        rcall INITIALIZE_DISPLAY
;-----------------------------------------------------------------;

GET_CHARACTER:
        lds Temp, UCSR1A
        sbrs Temp, RXC1
        rjmp GET_CHARACTER
        lds Data, UDR1
;-----------------------------------------------------------------;

OUTPUT_PORT:
        mov r24, Data
        mov r25, Temp
        rcall WRITE_CHARACTER
        com Data
        out PORTB, Data
        com Data
;-----------------------------------------------------------------;

PUT_CHARACTER:
        lds r25, UCSR1A sbrs r25, UDRE1
        rjmp PUT_CHARACTER
        sts UDR1, r24
        rjmp GET_CHARACTER
;-----------------------------------------------------------------;

INITIALIZE_DISPLAY:
        rcall WAIT_POWER_UP                 ; wait for display to
                                            ; power up
        ldi Data, BITMODE4                  ; 4-bit operation
        rcall WRITE_NIBBLE                  ; (in 8-bit mode)
        rcall WAIT_SHORT                    ; wait min. 39 us
        ldi Data, DISPCTRL                  ; disp. on, blink on,
                                            ; curs. On
        rcall WRITE_COMMAND                 ; send command
        rcall WAIT_SHORT                    ; wait min. 39 us
;-----------------------------------------------------------------;
```

```
CLEAR_DISPLAY:
        ldi Data, CLEAR                 ; clr display
        rcall WRITE_COMMAND             ; send command
        rcall WAIT_LONG                 ; wait min. 1.53 ms
        ret
;-----------------------------------------------------------------;

WRITE_CHARACTER:
        ldi RS, 0b00100000              ; RS = high
        rjmp WRITE
;-----------------------------------------------------------------;

WRITE_COMMAND:
        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
;-----------------------------------------------------------------;

WAIT_SHORT:
        clr ZH                          ; approx 50 us
        ldi ZL, 30
rjmp WAIT_LOOP
;-----------------------------------------------------------------;

WAIT_LONG:
        ldi ZH, HIGH(1000)              ; approx 2 ms
        ldi ZL, LOW(1000)
rjmp WAIT_LOOP
;-----------------------------------------------------------------;

WAIT_DBNC:
        ldi ZH, HIGH(4600)              ; approx 10 ms
```

```
        ldi ZL, LOW(4600)
rjmp WAIT_LOOP
;-----------------------------------------------------------------;

WAIT_POWER_UP:
        ldi ZH, HIGH(9000)              ; approx 20 ms
        ldi ZL, LOW(9000)
;-----------------------------------------------------------------;

WAIT_LOOP:
        sbiw z, 1                       ; 2 cycles
        brne WAIT_LOOP                  ; 2 cycles
ret
;-----------------------------------------------------------------;

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
```