;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

; ED2022, Computer Technology I

; Lab 5, task 1

; Hardware: STK600, CPU ATmega2560

; Date: 2015-11-26

; Authors: Ivan Hussein

; Function:

; --------

; The program send a character (One byte)to the LCD-display

; Used subroutine: Yes

; Global subroutines (that can be used from other programs):

; Other information: Stock Pointer

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"

.def Temp = r16

.def Data = r17

.def RS = r18

.def CharHi = r19

.def CharLo = r20

.def ASCIIbase = r21

.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: ser Temp ; r16 = 0b11111111

out DDRA, Temp ; port A = outputs

out DDRE, Temp ; port E = outputs ( Display JHD202A)

clr Temp ; r16 = 0b00000000

out PORTE, Temp

out DDRD, Temp ; port D = inputs

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

rcall init\_disp ; Initialize display

rcall long\_wait

ldi Data, 0b00100101 ; Write "%" character

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:

; out PORTA, Data ; set datalines

rcall switch\_output ; Modify for display JHD202A, port E

nop ; wait 542nS

; sbi PORTA, 4 ; enable high

sbi PORTE, 5 ; enable high, JHD202A

nop

nop ; wait 542nS

; cbi PORTA, 4 ; enable low

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

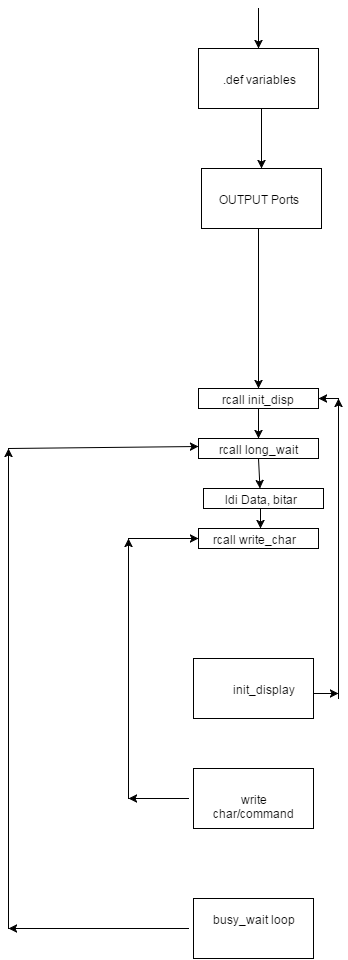
sbrc Data, 5 ; RS = 1?

ori Temp, 0b10000000 ; Set pin 7 (wrong in previous version)

out porte, Temp

pop Temp

ret



;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

; ED2022, Computer Technology I

; Lab 5, task 2

; Hardware: STK600, CPU ATmega2560

; Date: 2015-11-26

; Authors: Ivan Hussein

; Function:

; --------

; Electronic bingo generates a random number between 1 and 75. The number

; is send further on the LCD-display.

; Used subroutine: Yes

; Global subroutines (that can be used from other programs): Included

; Other information: Stock Pointer

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"

.def Temp = r16

.def Data = r17

.def RS = r18

.def CharHi = r19

.def CharLo = r20

.def ASCIIbase = r21

.def count = r22

.def num = r23

.def rest = r24

.def plus48 = r25

.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 interrupt

.org 0x0072

reset:

ser Temp ; r16 = 0b11111111

out DDRA, Temp ; port A = outputs

out DDRE, Temp ; port E = outputs ( Display JHD202A)

clr Temp ; r16 = 0b00000000

out PORTE, Temp

out DDRD, Temp ; port D = inputs

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

ldi r16,0b00000010

out EIMSK,r16 ;interrupt enabled

ldi r16,0b00001000

sts EICRA,r16

sei ;global interrupt enabled

rcall init\_disp

loop:

inc num ;increment from 0 to 76

cpi num, 76

brne loop

ldi num,0

rjmp loop

; \*\* 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:

; out PORTA, Data ; set datalines

rcall switch\_output ; Modify for display JHD202A, port E

nop ; wait 542nS

; sbi PORTA, 4 ; enable high

sbi PORTE, 5 ; enable high, JHD202A

nop

nop ; wait 542nS

; cbi PORTA, 4 ; enable low

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

interrupt:

rcall clr\_disp ; clear display

ldi count,0 ; modolus

sub10:

cpi num,10 ; less than 10 or not

brlo save ; if less than 10, save the value

subi num,10 ; if it is not subtract with 10,

inc count ; counts (value is subtracted)

rjmp sub10

save:

mov rest, num

rcall long\_wait

ldi plus48, 48

add count, plus48 ; adds 48

mov Data, count ; copies to Data

rcall write\_char ; writes out on display

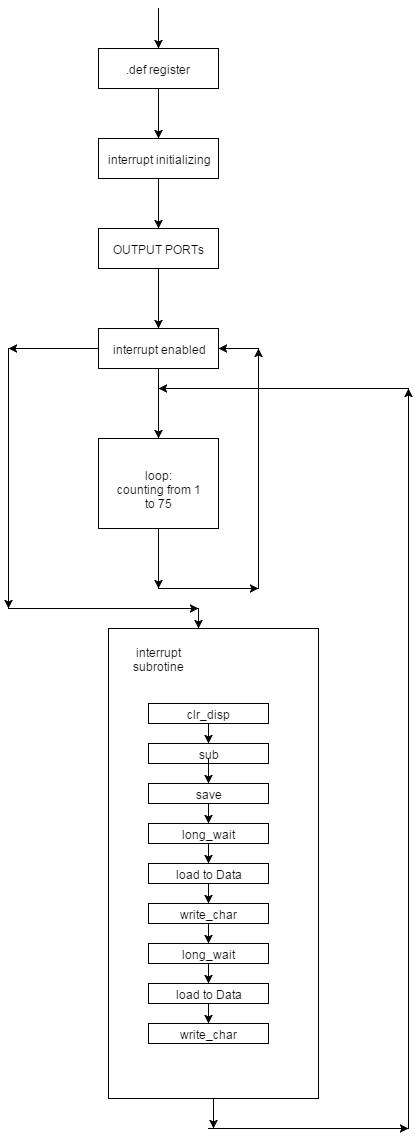
rcall long\_wait

add rest,plus48 ; adds 48

mov Data, rest ; copies to Data,

rcall write\_char ; writes out on display

reti



;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

; ED2022, Computer Technology I

; Lab 5, task 3

; Hardware: STK600, CPU ATmega2560

; Date: 2015-11-26

; Authors: Ivan Hussein

; Function:

; --------

; Serial communication and display: it receives a character on the serial ; port and displays each character on the display.

; Used subroutine: Yes

; Global subroutines (that can be used from other programs): Included

; Other information: Stock Pointer

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"

.def Temp = r16

.def char = r22

.equ UBRR\_val = 12 ; osc.aaaaaa = 1MHz, 4800 bps -=> UBBRR = 12

.def Data = r17

.def RS = r18

.def CharHi = r19

.def CharLo = r20

.def ASCIIbase = r21

.def Interrupt\_Counting = r23

.equ BITMODE4 = 0b00000010 ; 4-bit operation

.equ CLEAR = 0b00000001 ; Clear display

.equ DISPCTRL = 0b00001111 ; Display on, cursor on, blink on.

.cseg

.org 0x00

rjmp start

.org URXC1addr

rjmp GetChar ; USART interrupt

.org 0x0072

start:

ldi Interrupt\_Counting, 0

ldi Temp, 0xFF ; Set PortB outputs

out DDRB, Temp

out DDRA, Temp ; port A = outputs

out DDRE, Temp ; port E = outputs ( Display JHD202A)

clr Temp ; r16 = 0b00000000

out PORTE, Temp

out DDRD, Temp ; port D = inputs

ldi Temp, UBRR\_val ; soter Prescaler value in UBRR1L

sts UBRR1L, Temp

ldi Temp, (1<<TXEN1) | (1<<RXEN1) | (1<<RXCIE1)

sts UCSR1B, Temp ; set TX and RX and RXCIE enable flags

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

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

rcall init\_disp ; Initialize display

rcall long\_wait

rcall clr\_disp

sei ; global interrupt enable

main\_program:

rjmp main\_program

GetChar: ; receive data

inc Interrupt\_Counting

cpi Interrupt\_Counting, 21

breq break\_line

to\_break\_line:

lds Temp, UCSR1A ; read UCSR1A I/O register

lds Char, UDR1 ; character reading

Port\_output:

com Char

out PortB, char

com Char

PutChar: ; send data

lds Temp, UCSR1A ; read UCSR1A I/O register

sts UDR1, Char ; writew character

PrintChar: ; Prints the character

mov Data, char

rcall write\_char

rcall long\_wait

reti

break\_line:

ldi Data, 48

rcall write\_char ;1

rcall write\_char ;2

rcall write\_char ;3

rcall write\_char ;4

rcall write\_char ;5

rcall write\_char ;6

rcall write\_char ;7

rcall write\_char ;8

rcall write\_char ;9

rcall write\_char ;10

rcall write\_char ;11

rcall write\_char ;12

rcall write\_char ;13

rcall write\_char ;14

rcall write\_char ;15

rcall write\_char ;16

rcall write\_char ;17

rcall write\_char ;18

rcall write\_char ;19

rcall write\_char ;20

ldi Interrupt\_Counting, 1

rjmp to\_break\_line

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

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:

; out PORTA, Data ; set datalines

rcall switch\_output ; Modify for display JHD202A, port E

nop ; wait 542nS

; sbi PORTA, 4 ; enable high

sbi PORTE, 5 ; enable high, JHD202A

nop

nop ; wait 542nS

; cbi PORTA, 4 ; enable low

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

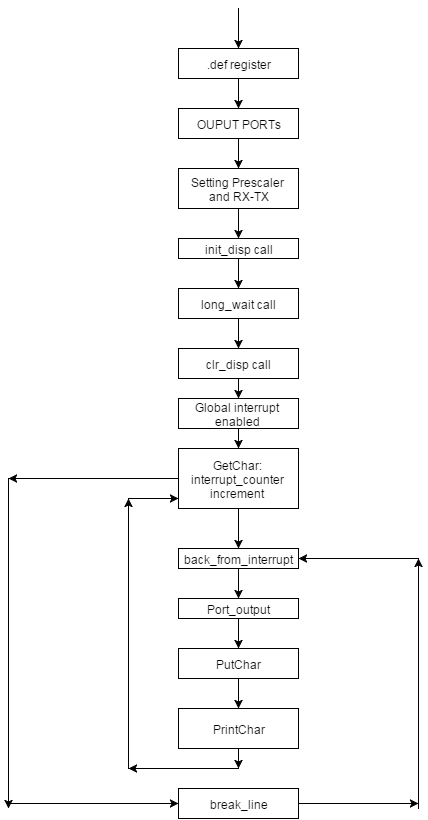
sbrc Data, 5 ; RS = 1?

ori Temp, 0b10000000 ; Set pin 7 (wrong in previous version)

out porte, Temp

pop Temp

ret



;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

; ED2022, Computer Technology I

; Lab 5, task 4

; Hardware: STK600, CPU ATmega2560

; Date: 2015-11-26

; Authors: Ivan Hussein

; Function:

; --------

; This program let 4 lines of text can be displayed. Each

; text line displays during 5 seconds, after that the text on

; line 1 should be moved to line 2 and so on. The text enters

; from the terminal program, via the serial port.

; Used subroutine: Yes

; Global subroutines (that can be used from other programs): Included

; Other information: Stock Pointer

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"

.def Temp = r16

.def char = r22

.equ UBRR\_val = 12 ; osc.aaaaaa = 1MHz, 4800 bps -=> UBBRR = 12

.def Data = r17

.def RS = r18

.def CharHi = r19

.def CharLo = r20

.def ASCIIbase = r21

.def Interrupt\_Counting = r23

.equ BITMODE4 = 0b00000010 ; 4-bit operation

.equ CLEAR = 0b00000001 ; Clear display

.equ DISPCTRL = 0b00001111 ; Display on, cursor on, blink on.

.cseg

.org 0x00

rjmp start

.org URXC1addr

rjmp GetChar ; USART interrupt

.org 0x0072

start:

ldi Interrupt\_Counting, 0

ldi Temp, 0xFF ; Set PortB outputs

out DDRB, Temp

out DDRA, Temp ; port A = outputs

out DDRE, Temp ; port E = outputs ( Display JHD202A)

clr Temp ; r16 = 0b00000000

out PORTE, Temp

out DDRD, Temp ; port D = inputs

ldi Temp, UBRR\_val ; soter Prescaler value in UBRR1L

sts UBRR1L, Temp

ldi Temp, (1<<TXEN1) | (1<<RXEN1) | (1<<RXCIE1)

sts UCSR1B, Temp ; set TX and RX and RXCIE enable flags

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

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

rcall init\_disp ; Initialize display

rcall long\_wait

rcall clr\_disp

sei ; global interrupt enabled

main:

ldi r25, 20

ldi XL, HIGH (0x1DDD) ; r31 ; pointer to row 1

ldi XH, LOW (0x1DDD) ; r30

ldi Temp,char ; first character

next:

st X+,Temp ; store character

; increment pointer

inc Temp ; increment char

dec r17 ; decrement counter

brne next

out\_text:

ldi r25, 20

ldi XL, HIGH(0x1DDD) ; pointer to row 1

ldi XH, LOW(0x1DDD)

next\_again:

ld char, X+

brne next\_again

rjmp main

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

GetChar: ; receive data

inc Interrupt\_Counting

cpi Interrupt\_Counting, 21

breq break\_line

to\_break\_line:

lds Temp, UCSR1A ; read UCSR1A I/O register

lds Char, UDR1; ; character reading

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Port\_output:

com Char

out PortB, char ; write character to PortB

com Char

PutChar: ; send data

lds Temp, UCSR1A ; read UCSR1A I/O register to r20

sts UDR1, Char ; writew character

PrintChar: ; Prints the character

mov Data, char

rcall write\_char

rcall long\_wait

reti

break\_line:

ldi Data, 48

rcall write\_char ;1

rcall write\_char ;2

rcall write\_char ;3

rcall write\_char ;4

rcall write\_char ;5

rcall write\_char ;6

rcall write\_char ;7

rcall write\_char ;8

rcall write\_char ;9

rcall write\_char ;10

rcall write\_char ;11

rcall write\_char ;12

rcall write\_char ;13

rcall write\_char ;14

rcall write\_char ;15

rcall write\_char ;16

rcall write\_char ;17

rcall write\_char ;18

rcall write\_char ;19

rcall write\_char ;20

ldi Interrupt\_Counting, 1

rjmp to\_break\_line

;---------------------------------------------------------------------------------------

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:

; out PORTA, Data ; set datalines

rcall switch\_output ; Modify for display JHD202A, port E

nop ; wait 542nS

; sbi PORTA, 4 ; enable high

sbi PORTE, 5 ; enable high, JHD202A

nop

nop ; wait 542nS

; cbi PORTA, 4 ; enable low

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

