

```

;* CharTest.asm
;*
;*
;* Created: 28/06/2017 9:37 p.m.
;* Author: ob1

;ST7820 128 x 64 graphics mode character display 8 lines x 21 characters
;Modification and redistribution under provisions of GNU general public license acknowledging source.
;Provided with no warranty of fitness for purpose (actual or consequential).

;Compiler Options
.NOLIST
.INCLUDE "C:\Program Files (x86)\Atmel\AVR Tools\AvrAssembler2\Appnotes\m8515def.inc"

;Hardware Allocations
;ATMega8515 @ 8 MHz

;PA0 LCD Data LSB Output
;PA1 LCD Data Output
;PA2 LCD Data Output
;PA3 LCD Data Output
;PA4 LCD Data Output
;PA5 LCD Data Output
;PA6 LCD Data Output
;PA7 LCD Data MSB Input/Output

;PE0 LCD RS Output
;PE1 LCD RW Output
;PE2 LCD Enable Output

.*****
;
;Set Constants

.EQU ChrBuf = SRAM_START ;21 chr * 8 lines = 168 bytes

.*****
;
;Define Registers

.DEF zero = R2 ;0x00 for speed
.DEF stack = R3 ;register for SREG storage in interrupts

.DEF LCDL = R6 ;LCD write word
.DEF LCDH = R7

.DEF LCDX = R8 ;LCD address LCDX = 0 to 7
.DEF LCDY = R9 ;LCDY = 0 to 56

.DEF chrBufX = R10 ;= 0 to 20
.DEF chrBufY = R11 ;= 0 to 7

.DEF lineCtr = R12 ;font line counter 0 to 6

.DEF bufChr = R15 ;character in buffer
.DEF chr = R16 ;current character value

.DEF flags = R17 ;processing flags

.DEF temp0 = R20 ;general purpose temporary storage
.DEF temp1 = R21
.DEF temp2 = R22

```

```

,*****
clearLCDgraphics:

    mov LCDY, zero
    mov LCDL, zero
    mov LCDH, zero

CLG2:
    mov LCDX, zero
    rcall LCDsetGDRAMaddr

CLG1:
    rcall LCDwriteGDRAMword
    inc LCDX
    mov temp0, LCDX
    cpi temp0, 16
    brlo CLG1

    inc LCDY
    mov temp0, LCDY
    cpi temp0, 64
    brlo CLG2

clearChrBuf:
    ldi temp0, 8 * 21
    ldi XL, LOW(ChrBuf)
    ldi XH, HIGH(ChrBuf)
    ldi chr, ''

CCB1:
    st X+, chr
    dec temp0
    brne CCB1
    ret

,*****
LCDinit:
    rcall delay10ms           ;delay40ms
    rcall delay10ms
    rcall delay10ms
    rcall delay10ms
    rcall LCDfunc0           ;basic function set
    rcall LCDfunc0           ;basic function set

    ldi XL, 0b00001100       ;LCD on, display on, cursor off, blink off
    rcall LCDcmd
    rcall delay10ms

    ldi XL, 0b00000001       ;clear display and home
    rcall LCDcmd
    rcall delay10ms

    ldi XL, 0b00000110       ;LCD Mode, cursor right, no scroll
    rcall LCDcmd             ;cursor right, no shift

    rcall LCDfunc1           ;basic function, graphics on

    rcall clearLCDgraphics

    sbi PORTD, PD7           ;turn on the LCD backlight
    ret

```

.*****

,

LCDcmd:

```
rcall busy
cbi PORTE, PE0      ;clr RS
cbi PORTE, PE1      ;clr RW
out PORTA, XL       ;data
rcall Epulse
ret
```

.*****

,

LCDwrite:

```
rcall busy
sbi PORTE, PE0      ;set RS
cbi PORTE, PE1      ;clr RW
out PORTA, XL       ;data
rcall Epulse
ret
```

.*****

,

busy:

```
out PORTA, zero     ;no pull up
out DDRA, zero      ;PORT A input
cbi PORTE, PE0      ;clr RS
sbi PORTE, PE1      ;set RW
```

busy1:

```
sbi PORTE, PE2      ;set E
nop
sbis PINA, PIND7
rjmp busy2

cbi PORTE, PE2      ;clear E
rjmp busy1
```

busy2:

```
cbi PORTE, PE2      ;clear E
push temp0
ldi temp0, 0xFF
out DDRA, temp0     ;Port A output
pop temp0
ret
```

.*****

,

Epulse:

```
sbi PORTE, PE2      ;set E
cbi PORTE, PE2      ;clr E
ret
```

.*****

,

LCDfunc0:

```
ldi XL, 0b00110000 ;8 bit, basic function, graphics off
rcall LCDcmd
rcall delay10ms
ret
```

```

.*****
;
LCDfunc1:                                ;8 bit, basic function, graphics display on
    ldi XL, 0b00110010
    rcall LCDcmd
    ret

.*****
;
LCDsetGDRAMaddr:                          ;Function Set 2, 8 bit, extended func, graphics display on
    ldi XL, 0b00110110
    rcall LCDcmd

    mov XL, LCDY
    andi XL, 0b00011111
    ori XL, 0b10000000
    rcall LCDcmd
    mov XL, LCDX
    andi XL, 0b00001111
    ori XL, 0b10000000
    rcall LCDcmd
    rcall LCDfunc1
    ret

.*****
;
LCDwriteGDRAMword:
    mov XL, LCDH
    rcall LCDwrite
    mov XL, LCDL
    rcall LCDwrite
    ret

.*****
;
LCDstr:                                    ;Z points to ASCIIZ in Flash
    lpm chrBufX, Z+
    lpm chrBufY, Z+

LCDstr1:
    lpm chr, Z+
    cp chr, zero
    breq strExit

    push ZL
    push ZH
    rcall writeGDRAMchr
    pop ZH
    pop ZL
    rjmp LCDstr1

strExit:
    ret

.*****
;
;Write Char to GDRAM
;ASCII in chr, X and Y character locations in chrBufX and chrBufY
;chrBufX and chrBufY are incremented with line/page wrap

writeGDRAMchr:
    mov temp2, chrBufX
    rcall getBufChr                        ;changes temp0, X, R1:R0; returns chr at chrBufX,Y
    cp chr, bufChr                         ;inc chrBufX,Y with line/page wrap. Exit if no change.

```

```

    breq noUpdate

    st X, chr                ;store the new character in the character buffer at X
    clr lineCtr              ;clear the font line counter

    mov temp0, chrBufY       ;calculate the LCDY address (0, 8, 16, 24)
    andi temp0, 0b00000011
    swap temp0
    lsr temp0
    mov LCDY, temp0
    ldi temp0, 6              ;6 bits per character line
    mul chrBufX, temp0        ;calculate current X char bit location
    mov temp0, R0
    swap temp0                ;div 16
    andi temp0, 0b00000111
    sbrc chrBufY, 2
    ori temp0, 0b00001000
    mov LCDX, temp0

writeLine:
    rcall LCDsetGDRAMAddr    ;write LCD GDRAM address in LCDX, LCDY

    ldi ZL, LOW(ChWrite)     ;word address 0x0119 say
    ldi ZH, HIGH(ChWrite)
    mov temp0, chrBufX
    add ZL, temp0             ;add word offset
    adc ZH, zero
    clc
    rol ZL
    rol ZH
    lpm XL, Z+
    lpm XH, Z
    movw ZL, XL
    icall

    inc LCDY                  ;increment the LCDYaddress for the next font line
    inc lineCtr
    ldi temp0, 7
    cp lineCtr, temp0
    brlo writeLine

noUpdate:
    inc chrBufX                ;increment the character location
    ldi temp0, 21
    cp chrBufX, temp0
    brlo exit

    clr chrBufX                ;line wrap
    inc chrBufY
    sbrc chrBufY, 3
    clr chrBufY                ;page wrap

exit:
    ret

;*****
LCDbcd:                        ;Y to LCD 4 digit BCD 9999 with leading zero suppression
    clr temp0                  ;counter
    sbr flags, 1<<lead0       ;lead zero flag

thou:
    cpi YH, HIGH(1000)
    brlo dthou
    brne thou1
    cpi YI, LOW(1000)

```

```

        brlo dthou

thou1:
    subi YL, LOW(1000)
    sbci YH, HIGH(1000)
    inc temp0
    rjmp thou

dthou:
    cpi temp0, 0x00
    brne dthou1
    ldi chr, ' '           ;ascii " "
    rjmp dthou2

dthou1:
    cbr flags, 1<<lead0
    ldi chr, 0x30
    add chr, temp0

dthou2:
    rcall writeGDRAMchr
    clr temp0

hun:
    cpi YH, 0x00
    brne subhun
    cpi YL, 100
    brlo dhun

subhun:
    subi YL, LOW(100)
    sbci YH, HIGH(100)
    inc temp0
    rjmp hun

dhun:
    sbrs flags, lead0
    rjmp dhun1
    cpi temp0, 0
    brne dhun1
    ldi chr, ' '           ;ascii " "
    rjmp dhun2

dhun1:
    cbr flags, 1<<lead0
    ldi chr, 0x30
    add chr, temp0

dhun2:
    rcall writeGDRAMchr
    clr temp0

ten:
    cpi YL, 10
    brlo dten
    inc temp0
    subi YL, 10
    rjmp ten

dten:
    sbrs flags, lead0
    rjmp dten1
    cpi temp0, 0
    brne dten1
    ldi chr, ' '           ;ascii " "
    rjmp dten2

```

```

dten1:
    cbr flags, 1<<lead0
    ldi chr, 0x30
    add chr, temp0

dten2:
    rcall writeGDRAMchr

one:
    ldi chr, 0x30
    add chr, YL
    rcall writeGDRAMchr
    ret

```

```

;*****
;LCD Graphics Character Subroutines

```

```

;*****
getBufChr:
    ldi temp0, 21
    mul temp0, chrBufY
    add R0, temp2
    adc R1, zero
    ldi temp0, LOW(ChrBuf)
    add R0, temp0
    adc R1, zero
    movw XL, R0
    ld bufChr, X
    ret
;get a character from chrBuf at chrBufY, temp2 (chrBufX)
;chr per line * line number
;result is single byte in R1:R0
;add x offset (note two byte result)
;X contains chrBuf address
;bufChr contains the chr value currently in the buffer

```

```

;*****
getFontLine:
    rcall getBufChr
    mov ZL, bufChr
    subi ZL, 0x20
    ldi temp0, 4
    mul ZL, temp0
    ldi temp0, LOW(CGROM)
    add R0, temp0
    ldi temp0, HIGH(CGROM)
    adc R1, temp0
    rol R0
    rol R1

    add R0, lineCtr
    add R1, zero

    movw ZL, R0
    lpm temp0, Z
    ret
;get the font line of chr into temp0
;subtract CGROM offset
;Flash memory is stored in words
;result in R1:R0
;CGROM is a word address in Flash

```

```

;*****
cw0001:
    ldi temp2, 0
    rcall getFontLine
    lsl temp0
    lsl temp0
    lsl temp0
    mov LCDH, temp0

    ldi temp2, 1
;load chr0 line y
;load chr1 line y

```

```
rcall getFontLine
mov temp1, temp0
lsl temp0
swap temp0
andi temp0, 0b00000011
or LCDH, temp0
```

```
lsl temp1
swap temp1
andi temp1, 0b11100000
mov LCDL, temp1
```

```
ldi temp2, 2 ;load chr2 line y
rcall getFontLine
lsl temp0
or LCDL, temp0
rcall LCDwriteGDRAMword
ret
```

```
.*****
```

```
,
cw02:
rcall cw0001
rcall cw0304
ret
```

```
.*****
```

```
,
cw0304:
ldi temp2, 2 ;load chr2 line y
rcall getFontLine
ror temp0
ror temp0
andi temp0, 0b10000000
mov LCDH, temp0

ldi temp2, 3 ;load chr3 line y
rcall getFontLine
lsl temp0
or LCDH, temp0

ldi temp2, 4 ;load chr4 line y
rcall getFontLine
lsl temp0
lsl temp0
lsl temp0
mov LCDL, temp0

ldi temp2, 5 ;load chr5 line y
rcall getFontLine
lsl temp0
swap temp0
andi temp0, 0b00000011
or LCDL, temp0
rcall LCDwriteGDRAMword
ret
```

```
.*****
```

```
,
cw05:
rcall cw0304
rcall cw0607
ret
```



```

,*****
,
cw0607:
    ldi temp2, 5                ;load chr5 line y
    rcall getFontLine
    lsl temp0
    swap temp0
    andi temp0, 0b11100000
    mov LCDH, temp0

    ldi temp2, 6                ;load chr6 line y
    rcall getFontLine
    mov temp1, temp0
    lsr temp0
    or LCDH, temp0
    ror temp1
    ror temp1
    andi temp1, 0b10000000
    mov LCDL, temp1

    ldi temp2, 7                ;load chr7 line y
    rcall getFontLine
    lsl temp0
    or LCDL, temp0
    rcall LCDwriteGDRAMword
    ret

```

```

,*****
,
cw0809:
    ldi temp2, 8                ;load chr8 line y
    rcall getFontLine
    lsl temp0
    lsl temp0
    lsl temp0
    mov LCDH, temp0

    ldi temp2, 9                ;load chr9 line y
    rcall getFontLine
    mov temp1, temp0
    lsl temp0
    swap temp0
    andi temp0, 0b00000011
    or LCDH, temp0

    lsl temp1
    swap temp1
    andi temp1, 0b11100000
    mov LCDL, temp1

    ldi temp2, 10               ;load chr10 line y
    rcall getFontLine
    lsr temp0
    or LCDL, temp0
    rcall LCDwriteGDRAMword
    ret

```

```

,*****
,
cw10:
    rcall cw0809
    rcall cw1112
    ret

```

```

,*****
,
cw1112:
    ldi temp2, 10                ;load chr10 line y
    rcall getFontLine
    ror temp0
    ror temp0
    andi temp0, 0b10000000
    mov LCDH, temp0

    ldi temp2, 11                ;load chr11 line y
    rcall getFontLine
    lsl temp0
    or LCDH, temp0

    ldi temp2, 12                ;load chr12 line y
    rcall getFontLine
    lsl temp0
    lsl temp0
    lsl temp0
    mov LCDL, temp0

    ldi temp2, 13                ;load chr13 line y
    rcall getFontLine
    lsl temp0
    swap temp0
    andi temp0, 0b00000011
    or LCDL, temp0
    rcall LCDwriteGDRAMword
    ret

```

```

,*****
,
cw13:
    rcall cw1112
    rcall cw1415
    ret

```

```

,*****
,
cw1415:
    ldi temp2, 13                ;load chr13 line y
    rcall getFontLine
    lsl temp0
    swap temp0
    andi temp0, 0b11100000
    mov LCDH, temp0

    ldi temp2, 14                ;load chr14 line y
    rcall getFontLine
    mov temp1, temp0
    lsr temp0
    or LCDH, temp0

    ror temp1
    ror temp1
    andi temp1, 0b10000000
    mov LCDL, temp1

    ldi temp2, 15                ;load chr15 line y
    rcall getFontLine
    lsl temp0
    or LCDL, temp0
    rcall LCDwriteGDRAMword

```

ret

.*****
,

cw1617:

ldi temp2, 16 ;load chr16 line y
rcall getFontLine
lsl temp0
lsl temp0
lsl temp0
mov LCDH, temp0

ldi temp2, 17 ;load chr17 line y
rcall getFontLine
mov temp1, temp0
lsl temp0
swap temp0
andi temp0, 0b00000011
or LCDH, temp0

lsl temp1
swap temp1
andi temp1, 0b11100000
mov LCDL, temp1

ldi temp2, 18 ;load chr18 line y
rcall getFontLine
lsr temp0
or LCDL, temp0
rcall LCDwriteGDRAMword
ret

.*****
,

cw18:

rcall cw1617
rcall cw1920
ret

.*****
,

cw1920:

ldi temp2, 18 ;load chr18 line y
rcall getFontLine
ror temp0
ror temp0
andi temp0, 0b10000000
mov LCDH, temp0

ldi temp2, 19 ;load chr19 line y
rcall getFontLine
lsl temp0
or LCDH, temp0

ldi temp2, 20 ;load chr20 line y
rcall getFontLine
lsl temp0
lsl temp0
lsl temp0
mov LCDL, temp0
rcall LCDwriteGDRAMword
ret

```
,*****  
;FLASH Constants
```

```
;Character ICALL Table
```

```
ChWrite: ;load Z and ICALL
```

```
.DB LOW(cw0001), HIGH(cw0001)  
.DB LOW(cw0001), HIGH(cw0001)  
.DB LOW(cw02), HIGH(cw02)  
.DB LOW(cw0304), HIGH(cw0304)  
.DB LOW(cw0304), HIGH(cw0304)  
.DB LOW(cw05), HIGH(cw05)  
.DB LOW(cw0607), HIGH(cw0607)  
.DB LOW(cw0607), HIGH(cw0607)  
.DB LOW(cw0809), HIGH(cw0809)  
.DB LOW(cw0809), HIGH(cw0809)  
.DB LOW(cw10), HIGH(cw10)  
.DB LOW(cw1112), HIGH(cw1112)  
.DB LOW(cw1112), HIGH(cw1112)  
.DB LOW(cw13), HIGH(cw13)  
.DB LOW(cw1415), HIGH(cw1415)  
.DB LOW(cw1415), HIGH(cw1415)  
.DB LOW(cw1617), HIGH(cw1617)  
.DB LOW(cw1617), HIGH(cw1617)  
.DB LOW(cw18), HIGH(cw18)  
.DB LOW(cw1920), HIGH(cw1920)  
.DB LOW(cw1920), HIGH(cw1920)
```

```
;Charactor Generator ROM
```

```
CGROM:
```

```
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ;0x20, Space  
.DB 0x04, 0x04, 0x04, 0x04, 0x04, 0x00, 0x04, 0x00 ;0x21, !  
.DB 0x09, 0x09, 0x12, 0x00, 0x00, 0x00, 0x00, 0x00 ;0x22, "  
.DB 0x0a, 0x0a, 0x1f, 0x0a, 0x1f, 0x0a, 0x0a, 0x00 ;0x23, #  
.DB 0x04, 0x0f, 0x14, 0x0e, 0x05, 0x1e, 0x04, 0x00 ;0x24, $  
.DB 0x19, 0x19, 0x02, 0x04, 0x08, 0x13, 0x13, 0x00 ;0x25, %  
.DB 0x04, 0x0a, 0x0a, 0x0a, 0x15, 0x12, 0x0d, 0x00 ;0x26, &  
.DB 0x04, 0x04, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00 ;0x27, '  
.DB 0x02, 0x04, 0x08, 0x08, 0x08, 0x04, 0x02, 0x00 ;0x28, (  
.DB 0x08, 0x04, 0x02, 0x02, 0x02, 0x04, 0x08, 0x00 ;0x29, )  
.DB 0x04, 0x15, 0x0e, 0x1f, 0x0e, 0x15, 0x04, 0x00 ;0x2a, *  
.DB 0x00, 0x04, 0x04, 0x1f, 0x04, 0x04, 0x00, 0x00 ;0x2b, +  
.DB 0x00, 0x00, 0x00, 0x00, 0x04, 0x04, 0x08, 0x00 ;0x2c, ,  
.DB 0x00, 0x00, 0x00, 0x1f, 0x00, 0x00, 0x00, 0x00 ;0x2d, -  
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x0c, 0x00 ;0x2e, .  
.DB 0x01, 0x01, 0x02, 0x04, 0x08, 0x10, 0x10, 0x00 ;0x2f, /  
.DB 0x0e, 0x11, 0x13, 0x15, 0x19, 0x11, 0x0e, 0x00 ;0x30, 0  
.DB 0x04, 0x0c, 0x04, 0x04, 0x04, 0x04, 0x0e, 0x00 ;0x31, 1  
.DB 0x0e, 0x11, 0x01, 0x02, 0x04, 0x08, 0x1f, 0x00 ;0x32, 2  
.DB 0x0e, 0x11, 0x01, 0x06, 0x01, 0x11, 0x0e, 0x00 ;0x33, 3  
.DB 0x02, 0x06, 0x0a, 0x12, 0x1f, 0x02, 0x02, 0x00 ;0x34, 4  
.DB 0x1f, 0x10, 0x1e, 0x01, 0x01, 0x11, 0x0e, 0x00 ;0x35, 5  
.DB 0x06, 0x08, 0x10, 0x1e, 0x11, 0x11, 0x0e, 0x00 ;0x36, 6  
.DB 0x1f, 0x01, 0x02, 0x04, 0x08, 0x08, 0x08, 0x00 ;0x37, 7  
.DB 0x0e, 0x11, 0x11, 0x0e, 0x11, 0x11, 0x0e, 0x00 ;0x38, 8  
.DB 0x0e, 0x11, 0x11, 0x0f, 0x01, 0x02, 0x0c, 0x00 ;0x39, 9  
.DB 0x00, 0x0c, 0x0c, 0x00, 0x0c, 0x0c, 0x00, 0x00 ;0x3a, :  
.DB 0x00, 0x0c, 0x0c, 0x00, 0x0c, 0x04, 0x08, 0x00 ;0x3b, ;  
.DB 0x02, 0x04, 0x08, 0x10, 0x08, 0x04, 0x02, 0x00 ;0x3c, <  
.DB 0x00, 0x00, 0x1f, 0x00, 0x1f, 0x00, 0x00, 0x00 ;0x3d, =  
.DB 0x08, 0x04, 0x02, 0x01, 0x02, 0x04, 0x08, 0x00 ;0x3e, >
```

