

```

,*****
,* written by: Garofalo Federico
,* date: 20/1/2011
,* version: 1.0
,* file saved as: Generatore_di_segnali
,* for AVR:ATMega8515
,*****

```

```

.nolist
.include "m8515def.inc"
.list

```

```

.def all=r16
.def k2=r17
.def carattere=r17
.def k3=r18
.def temp=r18
.def reg1=r19
.def reg2=r20
.def reg3=r21
.def reg4=r22
.def k4=r23
.def temp2=r23
.def temp1=r24
.def k1=r24

```

```

;** Interrupt vector table

```

rjmp reset	; Reset
reti	; External Interrupt0
reti	; External Interrupt1
reti	; T/C1 Capture event
reti	; T/C1 Compare match A
reti	; T/C1 Compare match B
reti	; T/C1 Overflow
reti	; T/C0 Overflow
reti	; SPI Serial Transfer Complete
reti	; UART Receive complete
reti	; UART Data Register Empty
reti	; UART UART Transmit Complete
reti	; Analog Comparator Interrupt handle
reti	; External Interrupt Request 2
reti	; Timer/Counter0 Compare Match
reti	; EEPROM Ready
reti	; Store Program memory Ready

```

.CSEG

```

;------

Tabella_milioni:

.DB 0b01000000

.DB 0b01000010 ;1.000.000

.DB 0b00001111 ;MSB

.DB 0b10000000

.DB 0b10000100 ;2.000.000

.DB 0b00011110 ;MSB

.DB 0b11000000

.DB 0b11000110 ;3.000.000

.DB 0b00101101 ;MSB

.DB 0b00000000

.DB 0b00001001 ;4.000.000

.DB 0b00111101 ;MSB

.DB 0b01000000

.DB 0b01001011 ;5.000.000

.DB 0b01001100 ;MSB

.DB 0b10000000	
.DB 0b10001101	;6.000.000
.DB 0b01011011	;MSB
.DB 0b11000000	
.DB 0b11001111	;7.000.000
.DB 0b01101010	;MSB
.DB 0b00000000	
.DB 0b00010010	;8.000.000
.DB 0b01111010	;MSB
.DB 0b01000000	
.DB 0b01010100	;9.000.000
.DB 0b10001001	;MSB

;-----
 Tabella_centinaia_migliaia:

.DB 0b10100000	
.DB 0b10000110	;100.000
.DB 0b00000001	;MSB

.DB 0b01000000	
.DB 0b00001101	;200.000
.DB 0b00000011	;MSB
.DB 0b11100000	
.DB 0b10010011	;300.000
.DB 0b00000100	;MSB
.DB 0b10000000	
.DB 0b00011010	;400.000
.DB 0b00000110	;MSB
.DB 0b00100000	
.DB 0b10100001	;500.000
.DB 0b00000111	;MSB
.DB 0b11000000	
.DB 0b00100111	;600.000
.DB 0b00001001	;MSB
.DB 0b01100000	

.DB 0b10101110 ;700.000

.DB 0b00001010 ;MSB

.DB 0b00000000

.DB 0b00110101 ;800.000

.DB 0b00001100 ;MSB

.DB 0b10100000

.DB 0b10111011 ;900.000

.DB 0b00001101 ;MSB

;

Tabella_decine_migliaia:

.DB 0b00010000

.DB 0b00100111 ;10.000

.DB 0b00000000 ;MSB

.DB 0b00100000

.DB 0b01001110 ;20.000

.DB 0b00000000 ;MSB

.DB 0b00110000

.DB 0b01110101 ;30.000

.DB 0b00000000 ;MSB

.DB 0b01000000

.DB 0b10011100 ;40.000

.DB 0b00000000 ;MSB

.DB 0b01010000

.DB 0b11000011 ;50.000

.DB 0b00000000 ;MSB

.DB 0b01100000

.DB 0b11101010 ;60.000

.DB 0b00000000 ;MSB

.DB 0b01110000

.DB 0b00010001 ;70.000

.DB 0b00000001 ;MSB

.DB 0b10000000

.DB 0b00111000 ;80.000

.DB 0b00000001 ;MSB

.DB 0b10010000

.DB 0b01011111 ;90.000

.DB 0b00000001 ;MSB

;-----

Tabella_migliaia:

.DB 0b11101000

.DB 0b00000011 ;1.000

.DB 0b00000000 ;MSB

.DB 0b11010000

.DB 0b00000111 ;2.000

.DB 0b00000000 ;MSB

.DB 0b10111000

.DB 0b00001011 ;3.000

.DB 0b00000000 ;MSB

.DB 0b10100000

.DB 0b00001111 ;4.000

.DB 0b00000000 ;MSB

.DB 0b10001000

.DB 0b00010011 ;5.000

.DB 0b00000000 ;MSB

.DB 0b01110000

.DB 0b00010111 ;6.000

.DB 0b00000000 ;MSB

.DB 0b01011000

.DB 0b00011011 ;7.000

.DB 0b00000000 ;MSB

.DB 0b01000000

.DB 0b00011111 ;8.000

.DB 0b00000000 ;MSB

.DB 0b00101000

.DB 0b00100011 ;9.000

.DB 0b00000000 ;MSB

;-

Tabella_centinaia:

.DB 0b01100100	
.DB 0b00000000	;100
.DB 0b00000000	;MSB
.DB 0b11001000	
.DB 0b00000000	;200
.DB 0b00000000	;MSB
.DB 0b00101100	
.DB 0b00000001	;300
.DB 0b00000000	;MSB
.DB 0b10010000	
.DB 0b00000001	;400
.DB 0b00000000	;MSB
.DB 0b11110100	
.DB 0b00000001	;500
.DB 0b00000000	;MSB
.DB 0b01011000	

```
.DB 0b00000010      ;600

.DB 0b00000000      ;MSB

.DB 0b101111100

.DB 0b00000010      ;700

.DB 0b00000000      ;MSB

.DB 0b00100000

.DB 0b00000011      ;800

.DB 0b00000000      ;MSB

.DB 0b10000100

.DB 0b00000011      ;900

.DB 0b00000000      ;MSB
```

```
#####
```

spi:

```
in temp, spdr
out spdr, temp1
```

wait:

```
sbis spsr,spif
rjmp wait
ret
```

```
#####
```

r_risoluzione:

```

                                clr r25
                                push temp
r:                                clr r8
                                rcall tastiera
                                cpi carattere,4                        ; ++
                                brne fine4
                                cpi r25,96
                                breq r
                                add r25, carattere
                                push r25
                                cpi r25,10
                                brlo pc+4
                                subi r25,10
                                inc r8
                                rjmp pc-4
                                mov carattere,r8
                                ldi temp,8
                                rcall LCD_locate2
                                rcall scrivi_carattere
                                mov carattere,r25
                                ldi temp,9
                                rcall LCD_locate2
                                rcall scrivi_carattere
                                pop r25
                                rjmp r
fine4:                            cpi carattere,12
                                breq pc+5
                                cpi carattere,14
                                breq pc+3
                                cpi carattere,15
                                brne r
                                pop temp
                                ret
```

#####

load_frequency_value:

```

                                rcall tastiera
                                cpi carattere,10
                                breq load_frequency_value
                                cpi carattere,11
                                breq load_frequency_value
                                cpi carattere,13
                                breq load_frequency_value
                                rcall riga_2_menu_frequenza
                                clr temp
                                rjmp pc+2
loop2:                            rcall tastiera
```

```

    cpi carattere,10
    breq loop2
    cpi carattere,11
    breq loop2
    cpi carattere,12
    brne pc+10

```

loop3:

```

    tst temp
    breq pc+4
    pop carattere
    dec temp
    rjmp loop3

```

```

    mov reg1,temp
    rcall riga_2_menu_frequenza
    mov temp, reg1
    rjmp menu_invio
    cpi carattere,13
    brne pc+2
    rcall r_risoluzione           ;modificato breq loop2
    cpi carattere,12
    breq loop3
    cpi carattere,14
    brne loop7

```

```

    tst temp
    brne pc+4
    mov reg1, temp
    rcall stringa_zero
    mov temp, reg1
    cpi temp,7                   ; modificato 8
    brne menu_invio             ; scrive (out of range) se il valore supera 1,999999Mhz
    in zl,spl
    in zh,sph
    adiw zh:zl,6                 ;modificato 7
    out spl, zl
    out sph, zh
    pop carattere
    push carattere
    sbiw zh:zl,6                 ;modificato 7
    out spl, zl
    out sph, zh
    cpi carattere,1
    breq loop8

```

loop5:

```

    tst temp
    breq pc+4
    pop carattere
    dec temp
    rjmp loop5

```

```

    mov reg1, temp
    rcall out_of_range

```

```
mov temp, reg1
rjmp load_frequency_value
```

```
loop8:      rjmp menu_invio
loop7:      cpi carattere,15
            brne pc+7
```

```
loop4:      tst temp
            breq pc+4
            pop carattere
            dec temp
            rjmp loop4
```

```
            rjmp menu_waves
            cpi temp,7           ;modificato 8
            breq loop2
            inc temp
            push carattere
            rcall scrivi_carattere
            rjmp loop2
```

```
;-----
```

menu_invio:

```
            clr reg1
            clr reg2
            clr reg3
            clr reg4
            tst temp
            brne pc+2
            rjmp fine2
            dec temp
            pop carattere
            cpi carattere,0
            breq pc+2
            mov reg4,carattere

            tst temp
            breq fine3
            dec temp
            pop carattere
            cpi carattere,0
            breq pc+4
            ldi temp1,10
            mul carattere,temp1
            add reg4,r0
```

```
            ldi ZH, high(Tabella_centinaia<<1)
            ldi ZL, low(Tabella_centinaia<<1)
            rcall somma
```

```
ldi ZH, high(Tabella_migliaia<<1)
ldi ZL, low(Tabella_migliaia<<1)
rcall somma
```

```
ldi ZH, high(Tabella_decine_migliaia<<1)
ldi ZL, low(Tabella_decine_migliaia<<1)
rcall somma
```

```
ldi ZH, high(Tabella_centinaia_migliaia<<1)
ldi ZL, low(Tabella_centinaia_migliaia<<1)
rcall somma
```

```
ldi ZH, high(Tabella_milioni<<1)
ldi ZL, low(Tabella_milioni<<1)
rcall somma
```

fine3:

```
push r25
rcall Mul32
pop r25
```

```
clr r8
tst r25
breq pc+4
subi r25,4
inc r8
rjmp pc-4
```

```
add reg4, r8
adc reg3, r25
adc reg2, r25
adc reg1, r25
```

fine2:

```
rcall routine1
rcall routine2
ret
```

;-----

routine1:

```
lsl reg2
brbc 0, pc+2
set
lsl reg1
brtc pc+2
inc reg1
clt
```

```
lsl reg2
brbc 0, pc+2
set
lsl reg1
brtc pc+2
inc reg1
```

```
sbrc reg3,7
sbr reg2,2
sbrc reg3,6
sbr reg2,1
ret
```

```
routine2:    cbr reg3,128
             sbr reg3,64
             cbr reg1,128
             sbr reg1,64
             ret
```

;-----

Somma:

```
pop xl
pop xh
tst temp
breq end
dec temp
pop carattere
cpi carattere,0
breq end
dec carattere
ldi temp1,6
mul carattere,temp1
add zl,r0
clr temp1
adc zh,temp1
lpm r23, Z
add reg4, r23
clt
brbc 0, pc+2
set
adiw zh:zl,2
lpm r23, Z
clc
brbc 6, pc+2
sec
adc reg3, r23
clt
brbc 0, pc+2
set
adiw zh:zl,2
lpm r23, Z
clc
```

```

end:      brbc 6, pc+2
          sec
          adc reg2, r23
          push xh
          push xl
          ret

```

;-----

Mul32:

```

ldi k4,0b00000101
ldi k3,0b00101110
ldi k2,0b00110011
ldi k1,0b00011010

```

;K=26,201606 -> 10,245 Mhz

```

clr R16
mul    k4,reg4
movw R8,R0
clr    R10
clr    R11
clr    R12
clr    R13
clr    R14
mul    k3,reg4
add    R9,R0
adc    R10,R1
mul    k2,reg4
add    R10,R0
adc    R11,R1
mul    k1,reg4
add    R11,R0
adc    R12,R1

```

```

mul    k4,reg3
add    R9,R0
adc    R10,R1
adc    R11,R16
adc    R12,R16
adc    R13,R16
mul    k3,reg3
add    R10,R0
adc    R11,R1
adc    R12,R16
adc    R13,R16
mul    k2,reg3
add    R11,R0
adc    R12,R1
adc    R13,R16
mul    k1,reg3
add    R12,R0

```

```

adc    R13,R1

mul    k4,reg2
add    R10,R0
adc    R11,R1
adc    R12,R16
adc    R13,R16
adc    R14,R16
mul    k3,reg2
add    R11,R0
adc    R12,R1
adc    R13,R16
adc    R14,R16
mul    k2,reg2
add    R12,R0
adc    R13,R1
adc    R14,R16
mul    k1,reg2
add    R13,R0
adc    R14,R1

```

```

mov reg4,r11
mov reg3,r12
mov reg2,r13
mov reg1,r14
ret

```

;-----

Invio_istruzioni:

```

mov temp1, reg3
cbi portb, pinb4           ;lsb data
rcall spi
mov temp1, reg4
rcall spi

```

```

mov temp1, reg1
rcall spi                 ;msb data
mov temp1, reg2
rcall spi
sbi portb, pinb4

```

```

ldi temp1, 0b11000000
cbi portb, pinb4
rcall spi
ldi temp1, 0b00000000 ;lsb data phase0
rcall spi

```

```

sbi portb, pinb4

ldi temp1, 0b11100000
cbi portb, pinb4
rcall spi
ldi temp1, 0b00000000 ;lsb data phase1
rcall spi
sbi portb, pinb4
ret

```

```

;#####

```

```

;#####scrive carattere digitato sul display#####

```

scrivi_carattere:

```

mov reg1, temp
ldi temp, 48
add temp, carattere
rcall LCD_Send_Data
mov temp, reg1
ret

```

```

;#####

```

out_of_range:

```

ldi temp,0
rcall LCD_locate2

ldi temp, 'o'
rcall LCD_Send_Data

ldi temp, 'u'
rcall LCD_Send_Data

ldi temp, 't'
rcall LCD_Send_Data

ldi temp, 32
rcall LCD_Send_Data

ldi temp, 'o'
rcall LCD_Send_Data

ldi temp, 'f'
rcall LCD_Send_Data

ldi temp, 32
rcall LCD_Send_Data

```

```
ldi temp, 'r'  
rcall LCD_Send_Data
```

```
ldi temp, 'a'  
rcall LCD_Send_Data
```

```
ldi temp, 'n'  
rcall LCD_Send_Data
```

```
ldi temp, 'g'  
rcall LCD_Send_Data
```

```
ldi temp, 'e'  
rcall LCD_Send_Data
```

```
ldi temp, 32  
rcall LCD_Send_Data  
ret
```

```
#####
```

stringa_zero:

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '.'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

ret

#####

menu:

rcall LCD_clear
ldi temp,3
rcall LCD_locate1

ldi temp, 'M'
rcall LCD_Send_Data

ldi temp, 'a'
rcall LCD_Send_Data

ldi temp, 'i'
rcall LCD_Send_Data

ldi temp, 'n'
rcall LCD_Send_Data

ldi temp, '}'
rcall LCD_Send_Data

#####

ldi temp, '{'
rcall LCD_Send_Data

ldi temp, 'M'
rcall LCD_Send_Data

ldi temp, 'e'
rcall LCD_Send_Data

ldi temp, 'n'
rcall LCD_Send_Data

ldi temp, 'u'
rcall LCD_Send_Data

;#####seconda riga#####

ldi temp, 0
rcall LCD_locate2

ldi temp, 'A'
rcall LCD_Send_Data

ldi temp, '-'
rcall LCD_Send_Data

ldi temp, 'W'
rcall LCD_Send_Data

ldi temp, 'a'
rcall LCD_Send_Data

ldi temp, 'v'
rcall LCD_Send_Data

ldi temp, 'e'
rcall LCD_Send_Data

ldi temp, 's'
rcall LCD_Send_Data

ldi temp, 32
rcall LCD_Send_Data

ldi temp, 32
rcall LCD_Send_Data

ldi temp, 'S'
rcall LCD_Send_Data

ldi temp, 'w'
rcall LCD_Send_Data

```
ldi temp, 'e'
rcall LCD_Send_Data
```

```
ldi temp, 'e'
rcall LCD_Send_Data
```

```
ldi temp, 'p'
rcall LCD_Send_Data
```

```
ldi temp, '-'
rcall LCD_Send_Data
```

```
ldi temp, 'B'
rcall LCD_Send_Data
ret
```

```
#####fine menù principale#####
```

```
#####menu waves#####
```

waves:

```
rcall LCD_clear
```

```
ldi temp, 'A'
rcall LCD_Send_Data
```

```
ldi temp, '-'
rcall LCD_Send_Data
```

```
ldi temp, 's'
rcall LCD_Send_Data
```

```
ldi temp, 'i'
rcall LCD_Send_Data
```

```
ldi temp, 'n'
rcall LCD_Send_Data
```

```
ldi temp, 'e'
rcall LCD_Send_Data
```

```
ldi temp, 32  
rcall LCD_Send_Data
```

```
ldi temp, 32  
rcall LCD_Send_Data
```

```
ldi temp, 's'  
rcall LCD_Send_Data
```

```
ldi temp, 'q'  
rcall LCD_Send_Data
```

```
ldi temp, 'u'  
rcall LCD_Send_Data
```

```
ldi temp, 'a'  
rcall LCD_Send_Data
```

```
ldi temp, 'r'  
rcall LCD_Send_Data
```

```
ldi temp, 'e'  
rcall LCD_Send_Data
```

```
ldi temp, '-'  
rcall LCD_Send_Data
```

```
ldi temp, 'B'  
rcall LCD_Send_Data
```

```
#####
```

```
ldi temp, 0  
rcall LCD_locate2
```

```
ldi temp, 'C'  
rcall LCD_Send_Data
```

```
ldi temp, '-'  
rcall LCD_Send_Data
```

```
ldi temp, 't'  
rcall LCD_Send_Data
```

```
ldi temp, 'r'  
rcall LCD_Send_Data
```

```
ldi temp, 'i'  
rcall LCD_Send_Data
```

```
ldi temp, 'a'  
rcall LCD_Send_Data
```

```
ldi temp, 'n'  
rcall LCD_Send_Data
```

```
ldi temp, 'g'  
rcall LCD_Send_Data
```

```
ldi temp, 'u'  
rcall LCD_Send_Data
```

```
ldi temp, 'l'  
rcall LCD_Send_Data
```

```
ldi temp, 'a'  
rcall LCD_Send_Data
```

```
ldi temp, 'r'  
rcall LCD_Send_Data
```

```
ret
```

```
#####
```

```
sweep:
```

```
rcall LCD_clear
```

```
ldi temp, 'S'
```

rcall LCD_Send_Data

ldi temp, 'w'
rcall LCD_Send_Data

ldi temp, 'e'
rcall LCD_Send_Data

ldi temp, 'e'
rcall LCD_Send_Data

ldi temp, 'p'
rcall LCD_Send_Data

ldi temp, ':'
rcall LCD_Send_Data

ldi temp, 32
rcall LCD_Send_Data

ldi temp, 's'
rcall LCD_Send_Data

ldi temp, 'i'
rcall LCD_Send_Data

ldi temp, 'n'
rcall LCD_Send_Data

ldi temp, 'e'
rcall LCD_Send_Data

ldi temp, '-'
rcall LCD_Send_Data

ldi temp, 'A'
rcall LCD_Send_Data

#####

```
ldi temp, 0
rcall LCD_locate2
```

```
ldi temp, 'B'
rcall LCD_Send_Data
```

```
ldi temp, '-'
rcall LCD_Send_Data
```

```
ldi temp, 's'
rcall LCD_Send_Data
```

```
ldi temp, 'q'
rcall LCD_Send_Data
```

```
ldi temp, 'u'
rcall LCD_Send_Data
```

```
ldi temp, 'a'
rcall LCD_Send_Data
```

```
ldi temp, 'r'
rcall LCD_Send_Data
```

```
ldi temp, 'e'
rcall LCD_Send_Data
```

```
ldi temp, 32
rcall LCD_Send_Data
```

```
ldi temp, 't'
rcall LCD_Send_Data
```

```
ldi temp, 'r'
rcall LCD_Send_Data
```

```
ldi temp, 'i'
rcall LCD_Send_Data
```

```
ldi temp, 'a'  
rcall LCD_Send_Data
```

```
ldi temp, 'n'  
rcall LCD_Send_Data
```

```
ldi temp, '-'  
rcall LCD_Send_Data
```

```
ldi temp, 'C'  
rcall LCD_Send_Data  
ret
```

```
#####
```

```
menu_sweep_2:
```

```
ldi temp, 32  
rcall LCD_Send_Data
```

```
ldi temp, 's'  
rcall LCD_Send_Data
```

```
ldi temp, 'w'  
rcall LCD_Send_Data
```

```
ldi temp, 'e'  
rcall LCD_Send_Data
```

```
ldi temp, 'e'  
rcall LCD_Send_Data
```

```
ldi temp, 'p'  
rcall LCD_Send_Data
```

```
ldi temp, ':'  
rcall LCD_Send_Data
```

```
ldi temp, 0  
rcall LCD_locate2
```

```
ldi temp, '6'  
rcall LCD_Send_Data
```

```
ldi temp, 's'  
rcall LCD_Send_Data
```

```
ldi temp, 32  
rcall LCD_Send_Data
```

```
ldi temp, 32  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, 'H'  
rcall LCD_Send_Data
```

```
ldi temp, 'z'  
rcall LCD_Send_Data
```

```
ldi temp, '-'  
rcall LCD_Send_Data
```

```
ldi temp, '>'  
rcall LCD_Send_Data
```

```
ldi temp, '5'  
rcall LCD_Send_Data
```

```
ldi temp, '0'  
rcall LCD_Send_Data
```

```
ldi temp, 'K'  
rcall LCD_Send_Data
```

```
ldi temp, 'h'  
rcall LCD_Send_Data
```

```
ldi temp, 'z'
```

```
rcall LCD_Send_Data  
ret
```

```
#####
```

```
sine:      ldi temp, 's'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'i'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'n'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'e'  
           rcall LCD_Send_Data  
           ret
```

```
#####
```

```
square:    ldi temp, 's'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'q'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'u'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'a'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'r'  
           rcall LCD_Send_Data
```

```
           ldi temp, 'e'  
           rcall LCD_Send_Data  
           ret
```

```
#####
```

```
ramp:      ldi temp, 'r'  
           rcall LCD_Send_Data
```

```
ldi temp, 'a'
rcall LCD_Send_Data
```

```
ldi temp, 'm'
rcall LCD_Send_Data
```

```
ldi temp, 'p'
rcall LCD_Send_Data
ret
```

```
#####
```

```
trian:      ldi temp, 't'
            rcall LCD_Send_Data
```

```
ldi temp, 'r'
rcall LCD_Send_Data
```

```
ldi temp, 'i'
rcall LCD_Send_Data
```

```
ldi temp, 'a'
rcall LCD_Send_Data
```

```
ldi temp, 'n'
rcall LCD_Send_Data
```

```
ldi temp, '.'
rcall LCD_Send_Data
ret
```

```
#####
```

```
menu_frequency:
```

```
rcall LCD_clear
```

```
ldi temp, 'F'
rcall LCD_Send_Data
```

```
ldi temp, 'r'
rcall LCD_Send_Data
```

```
ldi temp, 'e'  
rcall LCD_Send_Data
```

```
ldi temp, 'q'  
rcall LCD_Send_Data
```

```
ldi temp, 'u'  
rcall LCD_Send_Data
```

```
ldi temp, 'e'  
rcall LCD_Send_Data
```

```
ldi temp, 'n'  
rcall LCD_Send_Data
```

```
ldi temp, 'c'  
rcall LCD_Send_Data
```

```
ldi temp, 'y'  
rcall LCD_Send_Data
```

```
ldi temp, 32  
rcall LCD_Send_Data
```

```
ret
```

```
#####
```

riga_2_menu_frequenza:

```
ldi temp, 0  
rcall LCD_locate2
```

```
ldi temp, '_'  
rcall LCD_Send_Data
```

```
ldi temp, '_'  
rcall LCD_Send_Data
```

```
ldi temp, '_'  
rcall LCD_Send_Data
```

```
ldi temp, '_'
rcall LCD_Send_Data
```

```
ldi temp, '_'
rcall LCD_Send_Data
```

```
ldi temp, '_'
rcall LCD_Send_Data
```

```
ldi temp, '_'
rcall LCD_Send_Data
```

```
ldi temp, '.'
rcall LCD_Send_Data
```

```
ldi temp, '_'
rcall LCD_Send_Data
```

```
ldi temp, '_'
rcall LCD_Send_Data
```

```
ldi temp, 32
rcall LCD_Send_Data
```

```
ldi temp, 'H'
rcall LCD_Send_Data
```

```
ldi temp, 'z'
rcall LCD_Send_Data
```

```
ldi temp, 32
rcall LCD_Send_Data
```

```
ldi temp, 0
rcall LCD_locate2
ret
```

```
#####
```

tastiera:

#####scansione prima riga#####

cbi porta,7

nop

sbic pina,3

rjmp pc+5

w1: sbis pina,3

rjmp w1

ldi carattere,1

rjmp fine

sbic pina,2

rjmp pc+5

w2: sbis pina,2

rjmp w2

ldi carattere,2

rjmp fine

sbic pina,1

rjmp pc+5

w3: sbis pina,1

rjmp w3

ldi carattere,3

rjmp fine

sbic pina,0

rjmp pc+5

w4: sbis pina,0

rjmp w4

ldi carattere, 10 ;A

rjmp fine

sbi porta,7

#####scansione seconda riga#####

cbi porta,6

nop

sbic pina,3

rjmp pc+5

w5: sbis pina,3

rjmp w5

ldi carattere,4

rjmp fine

sbic pina,2

rjmp pc+5

w6: sbis pina,2

rjmp w6

ldi carattere,5

```

                                rjmp fine

                                sbic pina,1
                                rjmp pc+5
w7:                             sbis pina,1
                                rjmp w7
                                ldi carattere,6
                                rjmp fine

                                sbic pina,0
                                rjmp pc+5
w8:                             sbis pina,0
                                rjmp w8
                                ldi carattere, 11      ;B
                                rjmp fine

                                sbi porta,6

;#####scansione terza riga#####

                                cbi porta,5

                                nop
                                sbic pina,3
                                rjmp pc+5
w9:                             sbis pina,3
                                rjmp w9
                                ldi carattere,7
                                rjmp fine

                                sbic pina,2
                                rjmp pc+5
w10:                           sbis pina,2
                                rjmp w10
                                ldi carattere,8
                                rjmp fine

                                sbic pina,1
                                rjmp pc+5
w11:                           sbis pina,1
                                rjmp w11
                                ldi carattere,9
                                rjmp fine

                                sbic pina,0
                                rjmp pc+5
w12:                           sbis pina,0
                                rjmp w12
                                ldi carattere, 12      ;C
                                rjmp fine

                                sbi porta,5

```

#####scansione quarta riga#####

cbi porta,4

nop

sbic pina,3

rjmp pc+5

w13: sbis pina,3

rjmp w13

ldi carattere,15 ;*

rjmp fine

sbic pina,2

rjmp pc+5

w14: sbis pina,2

rjmp w14

ldi carattere,0

rjmp fine

sbic pina,1

rjmp pc+5

w15: sbis pina,1

rjmp w15

ldi carattere,14 ;#

rjmp fine

sbic pina,0

rjmp pc+5

w16: sbis pina,0

rjmp w16

ldi carattere, 13 ;D

rjmp fine

sbi porta,4

rjmp tastiera

#####fine#####

fine: ser all
out porta,all
ret

#####

LCD_clear:

ldi temp, 0b00000001
rcall LCD_Send_Command
ret

LCD_locate1:

```

ldi all, 0b10000000
add all, temp          ;-> posizione1x
mov temp, all
rcall LCD_Send_Command
ret

```

LCD_locate2:

```

ldi all, 0b11000000
add all, temp          ;-> posizione2x
mov temp, all
rcall LCD_Send_Command
ret

```

LCD_Send_Data:

```

out portd, temp        ;->scrivi carattere ASCII
ldi temp1, 0b10100000
out portc, temp1
ldi temp1, 0b10000000
out portc, temp1
rcall ritardo
ldi temp, 0b00000110 ;-> testo fermo, cursore incrementa
rcall LCD_Send_Command
ret

```

LCD_Send_Command:

```

out portd, temp
ldi temp1, 0b00100000
out portc, temp1
ldi temp1, 0b00000000
out portc, temp1
rcall ritardo
ret

```

ritardo:

```

ldi all, (0<<pinc7)|(1<<pinc6)|(1<<pinc5)
out portc, all
cbi ddrd, 7
sbi portd, 7
nop

```

wait1:

```

sbic pind, 7
rjmp wait1
sbi ddrd, 7
ldi all, (0<<pinc7)|(1<<pinc6)|(0<<pinc5)
out portc, all
ret

```

#####

reset_DDS:

```
ldi temp1, 0b00100001
cbi portb, pinb4           ;msb ctrl
rcall spi
ldi temp1, 0b00000000
rcall spi
sbi portb, pinb4
ret
```

DDS_seno:

```
ldi temp1, 0b00100000
cbi portb, pinb4           ;msb ctrl
rcall spi
ldi temp1, 0b00000000
rcall spi
sbi portb, pinb4
ret
```

DDS_onda_quadra:

```
ldi temp1, 0b00100000
cbi portb, pinb4           ;msb ctrl
rcall spi
ldi temp1, 0b01101000
rcall spi
sbi portb, pinb4
ret
```

DDS_onda_triangolare:

```
ldi temp1, 0b00100000
cbi portb, pinb4           ;msb ctrl
rcall spi
ldi temp1, 0b00000010
rcall spi
sbi portb, pinb4
ret
```

#####

reset:

; Initialization of the Stack Pointer (obligatory on 8515)
; Load the low byte of the constant RAMEND in register temp
; Initialize SPL (Stack Pointer Low Byte)
; Load the high byte of the constant RAMEND in register temp

```
ldi    temp, LOW(RAMEND)
out     SPL, temp
ldi     temp, HIGH(RAMEND)
out     SPH, temp
```

#####interfaccia fisica tastiera#####

```
ser temp
out porta, temp          ;1,1,1,1/ pull-up attivo
ldi temp, 0b11110000     ;r0,r1,r2,r3/c0,c1,c2,c3
out ddra, temp
```

#####interfaccia fisica DDS#####

```
clr temp
ldi temp, (1<<ddb5)|(1<<ddb7)|(1<<ddb4)|(1<<ddb6) ; mosi,sck,SS;ctrl out of DDS
out ddrb, temp
sbi portb, pinb4
sbi portb, pinb7
cbi portb, pinb6          ; default: uscita =0 per triangolare e sinusoidale
```

#####interfaccia fisica display#####

```
ldi temp, 0b11100000     ; rs,r/w,e
out DDRC, temp           ;ctrl line
ser temp
out DDRD, temp           ;data line
clr temp
out portd, temp
```

#####fine configurazione porte#####

#####Menu_principale#####

#####inizializzazione del DDS#####

```
clr temp
ldi temp, (0<<spie)|(1<<spe)|(1<<mstr)|(0<<spr0)|(0<<spr1)|(0<<dord)|(1<<cpol)|(0<<cpha)
out spcr, temp
sbi spsr, 0
in temp, spsr
in temp, spdr

rcall reset_DDS
```

#####inizializzazione del display#####

```
rcall ritardo

ldi temp, 0b00110000
rcall LCD_Send_Command

ldi temp, 0b00110000
rcall LCD_Send_Command

ldi temp, 0b00110000
rcall LCD_Send_Command
```

```
ldi temp, 0b00111100      ;fine inizializzazione
rcall LCD_Send_Command
```

```
ldi temp, 0b00001000      ;spegnimento display
rcall LCD_Send_Command
```

```
rcall LCD_clear            ;pulizia display
```

```
ldi temp, 0b00001100      ;accensione display
rcall LCD_Send_Command
```

```
#####
```

```
loop1:
```

```
rcall menu
```

```
loop:
```

```
rcall tastiera
cpi carattere, 11
breq menu_sweep
cpi carattere, 10
breq menu_waves
rjmp loop
```

```
#####
```

```
menu_waves:
```

```
rcall reset_DDS

rcall waves
rcall tastiera
cpi carattere,10
breq menu_seno
cpi carattere,11
breq menu_onda_quadra
cpi carattere,12
breq menu_onda_triangolare
cpi carattere,15
breq loop1
rjmp pc-9
```

```
#####
```

```
menu_sweep:
```

```

rcall reset_DDS

rcall sweep
rcall tastiera
cpi carattere,10
breq sweep_sine
cpi carattere,11
brne pc+2
rjmp sweep_square
cpi carattere,12
brne pc+2
rjmp sweep_triangular
cpi carattere,15
breq loop1
rjmp pc-11

```

```

;#####

```

menu_seno:

```

rcall menu_frequency
rcall sine
rcall riga_2_menu_frequenza
cbi portb,6
rcall load_frequency_value
rcall Invio_istruzioni

rcall DDS_seno
rjmp pc-3

```

```

;#####

```

menu_onda_quadra:

```

rcall menu_frequency
rcall square
rcall riga_2_menu_frequenza
sbi portb,6
rcall load_frequency_value
rcall Invio_istruzioni

rcall DDS_onda_quadra
rjmp pc-3

```

```

;#####

```

menu_onda_triangolare:

```

rcall menu_frequency
rcall trian
rcall riga_2_menu_frequenza
cbi portb,6
rcall load_frequency_value

```

```
rcall Invio_istruzioni

rcall DDS_onda_triangolare
rjmp pc-3
```

```
;#####
```

sweep_sine:

```
rcall LCD_clear
ldi temp, 0
rcall LCD_locate1
rcall sine
rcall menu_sweep_2
cbi portb,6
rcall tastiera
cpi carattere,15
brne pc+2
rjmp menu_sweep
cpi carattere,14
breq loop10
rjmp pc-6
```

loop10:

```
clr reg4
clr reg3
```

loop11:

```
ldi temp2,1
add reg4, temp2
ldi temp2,0
adc reg3,temp2
clr reg2
clr reg1
```

```
push reg4
push reg3
```

```
;#####
```

```
rcall Mul32
rcall routine1
rcall routine2
```

```
;#####
```

```
rcall Invio_istruzioni
rcall DDS_seno
```

```
clr temp2
inc temp2
```

```

    cpi temp2,200
    brne pc-2

    pop reg3
    pop reg4

    cbi porta,4
    nop
    sbic pina,3
    rjmp pc+6
    sbis pina,3
    rjmp pc-1
    sbi porta,4
    nop
    rjmp sweep_sine    ;*
    sbi porta,4

```

```

    cpi reg3,0xC3
    breq pc+2
    rjmp loop11
    cpi reg4,0x50 ; cpi reg3,50Khz
    breq pc+2
    rjmp loop11
    rjmp loop10

```

```

;#####

```

sweep_square:

```

    rcall LCD_clear
    ldi temp, 0
    rcall LCD_locate1
    rcall square
    rcall menu_sweep_2
    sbi portb,6
    rcall tastiera
    cpi carattere,15
    brne pc+2
    rjmp menu_sweep
    cpi carattere,14
    breq pc+2
    rjmp pc-6

```

```

loop13:    clr reg4
           clr reg3

```

```

loop14:    ldi temp2,1
           add reg4, temp2

```

```
ldi temp2,0
adc reg3,temp2
clr reg2
clr reg1
```

```
push reg4
push reg3
```

```
;#####
```

```
rcall Mul32
rcall routine1
rcall routine2
```

```
;#####
```

```
rcall Invio_istruzioni
rcall DDS_onda_quadra
```

```
clr temp2
inc temp2
cpi temp2,200
brne pc-2
```

```
pop reg3
pop reg4
```

```
cbi porta,4
nop
sbic pina,3
rjmp pc+6
sbis pina,3
rjmp pc-1
sbi porta,4
nop
rjmp sweep_square    ;*
sbi porta,4
```

```
cpi reg3,0xC3
breq pc+2
rjmp loop14
cpi reg4,0x50 ; cpi reg3,50Khz
breq pc+2
rjmp loop14
rjmp loop13
```

```
;#####
```

sweep_triangular:

```
rcall LCD_clear
```

```

ldi temp, 0
rcall LCD_locate1
rcall trian
rcall menu_sweep_2
cbi portb,6
rcall tastiera
cpi carattere,15
brne pc+2
rjmp menu_sweep
cpi carattere,14
breq pc+2
rjmp pc-6

```

```

loop16:      clr reg4
             clr reg3

```

```

loop17:      ldi temp2,1
             add reg4, temp2
             ldi temp2,0
             adc reg3,temp2
             clr reg2
             clr reg1

```

```

push reg4
push reg3

```

```

;#####

```

```

rcall Mul32
rcall routine1
rcall routine2

```

```

;#####

```

```

rcall Invio_istruzioni
rcall DDS_onda_triangolare

```

```

clr temp2
inc temp2
cpi temp2,200
brne pc-2

```

```

pop reg3
pop reg4

```

```

cbi porta,4
nop
sbic pina,3
rjmp pc+6
sbis pina,3

```

```
rjmp pc-1
sbi porta,4
nop
rjmp sweep_triangular ;*
sbi porta,4
```

```
cpi reg3,0xC3
breq pc+2
rjmp loop17
cpi reg4,0x50 ; cpi reg3,50Khz
breq pc+2
rjmp loop17
rjmp loop16
```

```
#####
```