

Anexo A: Código para realização do trabalho

/*****

Projecto : Monitoramento dos parametros do solo usando o atmega8535

Data : 24/6/2008

Autor : Angelina R. Sambo

Curso : Engenharia Electronica

Tipo de Chip : ATmega8535

Tipo de Programa : Aplicacao

frequencia do Clock : 4000000Hz

Tamanho da RAM Externa : 0

*****/

.include "m8535def.inc"

.cseg

.org 0

//Definicao e inicializacao das variaveis

.def temp = r16

.def backupL = r17

.def backupH = r18

.def output_result = r19

;r20 = delay_10ms

;r21 = delay_10ms

;r22 = nao usado

;r23 = delay_10s

;r24 = delay_10s

;r25 = delay_10s

.def Lrefa = r26

.def Hrefa = r27

.def Lrefb = r28

.def Hrefb = r29

.equ Data = portD

.equ Data_bit = 0x28

.equ D_clear = 0x01

.equ D_on_off = 0x0C

.equ Shift_move_cursor = 0x06

.equ DD_RAM_Line0 = 0x00

.equ DD_RAM_Line1 = 0x40

.equ RS = 1

.equ RW = 2

.equ EN = 3

.equ exc_hum = 4

.equ exc_t = 5

.equ perigo = 6

.equ hrefha = 0x02 ;Byte mais significativo do valor de referencia maxima para
humidade

.equ lrefha = 0x9A ;Byte menos significativo do valor de referencia maxima para
humidade

.equ hrefhb = 0x02 ;Byte mais significativo do valor de referencia minima para
humidade

```

.equ lrefhb = 0x33 ;Byte menos significativo do valor de referencia minima para
humidade
.equ hrefta = 0x03 ;Byte mais significativo do valor de referencia maxima para
temperatura
.equ lrefta = 0xA0 ;Byte menos significativo do valor de referencia maxima para
temperatura
.equ hreftb = 0x00 ;Byte mais significativo do valor de referencia minima para
temperatura
.equ lreftb = 0x0A ;Byte menos significativo do valor de referencia minima para
temperatura
.equ Adc0 = 0x00 ;Multiplexador ADC para entrada analogica0
.equ Adc2 = 0x02 ;Multiplexador ADC para entrada analogica2

```

```

rjmp main

```

```

main:

```

```

;Inicializacao do Strack
ldi temp,high(RAMEND)
out SPH,temp
ldi temp,low(RAMEND)
out SPL,temp

```

```

;Direccao dos pinos das portas

```

```

ldi temp,0xFF
out DDRD,temp
ldi temp,0xFE //DDRB=0b11111110
out DDRB,temp
ldi temp,0x00
out DDRA,temp
out DDRC,temp
ldi temp,0xFF
out PortC,temp
ldi temp,0x01
out PortB,temp
clr r11

```

```

/****Inicializacao do display****/

```

```

cbi portB,RS
cbi portB,RW
sbi portB,EN
rcall delay_100ms

```

```

;Definicao do numero de bits a usar para vizualizar no LCD (4bits ou 8bits)

```

```

sbi portB,EN
ldi output_result,Data_bit
swap output_result
out Data,output_result
rcall delay_10ms
cbi portB,EN

```

```
rcall delay_10ms
sbi portB,EN
```

;Function set comand: data bists, N,F

```
ldi temp,Data_bit
mov output_result,temp
rcall Send_data
```

;Specify activation of display (D), cursor(C), and blinking of character at cursor position(B)

```
ldi temp,D_on_off
mov output_result,temp
rcall Send_data
rcall delay_100ms
```

;Shift display or move cursor

```
ldi temp,Shift_move_cursor //entry mode comand
mov output_result,temp
rcall Send_data
```

;Set display clear

```
rcall clear_display
ldi temp,D_clear
mov output_result,temp
rcall Send_data
```

/*****Fim da inicializacao do display*****/

;leitura do valor para a conversao

;Set ADC

```
sbi ADCSRA,7
cbi ADCSRA,6
cbi ADCSRA,5
cbi ADCSRA,4
cbi ADCSRA,3
cbi ADCSRA,2
cbi ADCSRA,1
cbi ADCSRA,0
```

Switch:

```
rcall keystatus
ldi temp,Adc0 ;Escolha do ordem dos registos com resultado da conversao
out ADMUX,temp
sbi ADCSRA,6
ldi Lrefa,lrefha ;Byte menos significativo da referencia de humidade alta
ldi Hrefa,hrefha ;Byte mais significativo da referencia de humidade alta
ldi Lrefb,lrefhb ;Byte menos significativo da referencia de humidade baixa
ldi Hrefb,hrefhb ;Byte mais significativo da referencia de humidade baixa
rcall convercao ;Convercao analogica para digital do sinal do sensor de
```

humidade

```
rcall comparacao_hum  
rcall delay_10s
```

```
rcall keystatus  
ldi temp,Adc2 ;Set conversion Result register input A2  
out ADMUX,temp  
sbi ADCSRA,6  
ldi Lrefa,lrefa ;Byte menos significativo da referencia da temperatura alta  
ldi Hrefa,hrefa ;Byte mais significativo da referencia da temperatura alta  
ldi Lrefb,lrefb ;Byte menos significativo da referencia da temperatura baixa  
ldi Hrefb,hrefb ;Byte mais significativo da referencia da temperatura baixa  
rcall convercao ;Convercao analogica para digital do sinal do sensor de
```

temperatura

```
rcall comparacao_temp ;comparacao com os valores de referencia  
rcall delay_10s  
rjmp Switch
```

keystatus:

```
sbic PortB,0  
ret  
off:  
rjmp off
```

convercao:

;Visualizacao do conteudo do ADCL

test:

```
sbis ADCSRA,4 ;testa se a conversao terminou  
rjmp test  
in backupL,ADCL  
rcall delay_10ms  
mov r1,backupL  
in backupH,ADCH  
mov r2,backupH  
rcall delay_10ms  
ret
```

comparacao_hum:

```
cp Lrefa,backupL ;Comparacao entre valor lido e valor de referencia alto  
cpc Hrefa,backupH  
brcs exc_agua ;Caso esteja acima da media chama subrotina do excesso  
cp Lrefb,backupL ;Comparacao entre valor lido e valor de referencia baixa  
cpc Hrefb,backupH  
brcc falt_agua ;Caso esteja abaixo da media chama subrotina da falta  
rcall result_L2  
rcall rotina  
clt  
bld r11,4  
rcall Set_bit5  
ret
```

comparacao_temp:

```
    cp Lrefa,backupL    ;Comparacao entre valor lido e valor de referencia alto
    cpc Hrefa,backupH
    brcs T_alta         ;Caso esteja acima da media chama subrotina do excesso
    cp Lrefb,backupL    ;Comparacao entre valor lido e valor de referencia baixa
    cpc Hrefb,backupH
    brcc T_baixa        ;Caso esteja abaixo da media chama subrotina da falta
    rcall result_L2
    rcall rotina
    clt
    bld r11,5
    rcall Set_bit4
    ret
```

Set_bitvm:

```
    sbi PortB,6
    cbi PortB,4
    ret
```

Set_bitvd:

```
    cbi PortB,4
    cbi PortB,6
    ret
```

Set_bit5:

```
    sbrs r11,5
    rcall Set_bitvd
    rcall on
    ret
```

Set_bit4:

```
    sbrs r11,4
    rcall Set_bitvd
    rcall on
    ret
```

on:

```
    sbi PortB,4
    cbi PortB,6
    ret
```

exc_agua:

```
    rcall result_L2e
    rcall rotina
    set
    bld r11,4
    bst r11,5
    brtc exc1
    rcall Set_bitvm
```

```
ret
exc1:
rcall on
ret
```

```
falt_agua:
    rcall result_L2f
    rcall rotina
    set
    bld r11,4
    bst r11,5
    brtc falt1
    rcall Set_bitvm
    ret
falt1:
    rcall on
    ret
```

```
T_alta:
    rcall result_L2et
    rcall rotina
    set
    bld r11,5
    bst r11,4
    brtc alt1
    rcall Set_bitvm
    ret
alt1:
    rcall on
    ret
```

```
T_baixa:
    rcall result_L2ft
    rcall rotina
    set
    bld r11,5
    bst r11,4
    brtc baix1
    rcall Set_bitvm
    ret
baix1:
    rcall on
    ret
```

```
rotine:
    rcall conv_rotine
    rcall DD_RAM_L2
    rcall result_inic_L1
    rcall result
    ret
```

```
result_inic_L1:
;Visualizacao da inicializacao
    ldi temp,'V'
    rcall letra
    ldi temp,'i'
    rcall letra
    ldi temp,'n'
    rcall letra
    ldi temp,':'
    rcall letra
    ldi temp,' '
    rcall letra
    ret
```

;Subrotina para escrever "NORMAL"

```
result_L2:
    rcall DD_RAM_L1
    rcall clear_display
    ldi temp,'N'
    rcall letra
    ldi temp,'O'
    rcall letra
    ldi temp,'R'
    rcall letra
    ldi temp,'M'
    rcall letra
    ldi temp,'A'
    rcall letra
    ldi temp,'L'
    rcall letra
    ret
```

;Subrotina para escrever "EXCESSO DE AGUA"

```
result_L2e:
    rcall DD_RAM_L1
    rcall clear_display
    ldi temp,'E'
    rcall letra
    ldi temp,'X'
    rcall letra
    ldi temp,'C'
    rcall letra
    ldi temp,'E'
    rcall letra
    ldi temp,'S'
    rcall letra
    ldi temp,'S'
    rcall letra
```

```

ldi temp,'O'
rcall letra
ldi temp,' '
rcall letra
ldi temp,'D'
rcall letra
ldi temp,'E'
rcall letra
ldi temp,' '
rcall letra
ldi temp,'A'
rcall letra
ldi temp,'G'
rcall letra
ldi temp,'U'
rcall letra
ldi temp,'A'
rcall letra
ret

```

;Subrotina para escrever "FALTA DE AGUA"
result_L2f:

```

rcall DD_RAM_L1
rcall clear_display
ldi temp,'F'
rcall letra
ldi temp,'A'
rcall letra
ldi temp,'L'
rcall letra
ldi temp,'T'
rcall letra
ldi temp,'A'
rcall letra
ldi temp,' '
rcall letra
ldi temp,'D'
rcall letra
ldi temp,'E'
rcall letra
ldi temp,' '
rcall letra
ldi temp,'A'
rcall letra
ldi temp,'G'
rcall letra
ldi temp,'U'
rcall letra
ldi temp,'A'
rcall letra

```


ret

result_L2et:

```
rcall DD_RAM_L1
rcall clear_display
ldi temp,'T'
rcall letra
ldi temp,'E'
rcall letra
ldi temp,'M'
rcall letra
ldi temp,'P'
rcall letra
ldi temp,'E'
rcall letra
ldi temp,'R'
rcall letra
ldi temp,'A'
rcall letra
ldi temp,'T'
rcall letra
ldi temp,'U'
rcall letra
ldi temp,'R'
rcall letra
ldi temp,'A'
rcall letra
ldi temp,' '
rcall letra
ldi temp,'A'
rcall letra
ldi temp,'L'
rcall letra
ldi temp,'T'
rcall letra
ldi temp,'A'
rcall letra
ret
```

result_L2ft:

```
rcall DD_RAM_L1
rcall clear_display
ldi temp,'T'
rcall letra
ldi temp,'E'
rcall letra
ldi temp,'M'
rcall letra
ldi temp,'P'
rcall letra
```

```

ldi temp,'E'
rcall letra
ldi temp,'R'
rcall letra
ldi temp,'A'
rcall letra
ldi temp,'T'
rcall letra
ldi temp,'U'
rcall letra
ldi temp,'R'
rcall letra
ldi temp,'A'
rcall letra
ldi temp,' '
rcall letra
ldi temp,'B'
rcall letra
ldi temp,'A'
rcall letra
ldi temp,'T'
rcall letra
ldi temp,'X'
rcall letra
ldi temp,'A'
rcall letra
ret

```

;Subrotina de conversao

result:

```

;Visualizacao da parte inteira
    mov output_result,r5
    sbi portB,RS
    rcall Send_data
;Visualizacao do ponto decimal
    mov output_result,r6
    sbi portB,RS
    rcall Send_data
;Visualizacao da parte decimal
    mov output_result,r7
    sbi portB,RS
    rcall Send_data
    mov output_result,r8
    sbi portB,RS
    rcall Send_data
    mov output_result,r9
    sbi portB,RS
    rcall Send_data
    mov output_result,r10

```

```
sbi portB,RS
rcall Send_data
ret ;fim da convercao e vizualizacao dos resultados
```

conv_rotine:

```
rcall check ;Verifica se o valor nao excede o maximo
brcs error_mesage ;Vizaliza que houve erro
rcall multiplication ;Multiplicacao por 320308 (5000*65535/1023)
rcall division ;Divisao do resultado por 65535
rcall conv_ascii ;Conversao para caracteres ASCII
rjmp set_point ;Colocacao da virgula para vizualizacao das casas decimais
```

check:

```
ldi temp,0x03
cp temp,r2
ret
```

error_mesage:

```
ldi temp,'E'
mov r5,temp
mov r7,temp
mov r8,temp
mov r9,temp
```

set_point:

```
ldi temp','
mov r6,temp
clr temp
mov r10,temp
ret
```

multiplication:

```
clr r6
ldi temp,0x04
mov r5,temp
ldi temp,0xE3
mov r4,temp
ldi temp,34
mov r3,temp
clr r10
clr r9
clr r8
clr r7
```

mult1:

```
mov temp,r1
or temp,r2
brne mult2
ret
```

mult2:

```
lsl r2 ;divide conteudo do byte mais significativo por 2
ror r1 ;divide conteudo do byte menos significativo por 2
brcc mult3
add r7,r3
adc r8,r4
adc r9,r5
adc r10,r6
```

mult3:

```
lsl r3 ;multiplica conteudo por 2
rol r4
rol r5
rol r6
rjmp mult1
```

division:

```
mov r2,r10
mov r1,r9
ret
```

conv_ascii:

```
ldi temp,LOW(1000)
mov r3,temp
ldi temp,HIGH(1000)
mov r4,temp
rcall sub_suc
mov r5,temp
clr r4
ldi temp,100
mov r3,temp
rcall sub_suc
mov r7,temp
ldi temp,10
mov r3,temp
rcall sub_suc
mov r8,temp
ldi temp,'0'
add temp,r1
mov r9,temp
ret
```

sub_suc:

```
ldi temp,'0' ;inicializacao do contador
sub1:
    cp r1,r3
    cpc r2,r4
    brcc sub2
    ret
sub2:
```

```

    sub r1,r3
    sbc r2,r4
    inc temp
    rjmp sub1

```

;DD RAM Line1 address set

DD_RAM_L1:

```

    ldi temp,DD_RAM_Line0
    ori temp,0x80
    mov output_result,temp
    cbi portB,RS
    rcall Send_data
    ret

```

;DD RAM Line2 address set

DD_RAM_L2:

```

    ldi temp,DD_RAM_Line1
    ori temp,0x80
    mov output_result,temp
    cbi portB,RS
    rcall Send_data
    ret

```

letra:

```

    mov output_result,temp
    sbi portB,RS
    rcall Send_data
    ret

```

clear_display:

```

    ldi temp,D_clear
    mov output_result,temp
    rcall Send_data
    ret

```

;Sending data or command

Send_data: ;Output_resulr = input argument

```

    sbi portB,EN
    swap output_result
    out Data,output_result ;send High nibble
    rcall delay_10ms
    cbi portB,EN
    rcall delay_10ms
    sbi portB,EN
    swap output_result
    out Data,output_result ;Send Low nibble
    rcall delay_10ms
    cbi portB,EN
    rcall delay_10ms

```

```
sbi portB,EN
rcall delay_100ms
ret
```

```
delay_10ms:
    ldi r20,55
    loop1:
        ldi r21,0xFF
        loop2:
            dec r21
            brne loop2 ;Se nao tiver a condicao FFFF volta a repetir a operacao
        dec r20
        brne loop1 ;Se nao for zero, apaga os registos contadores e inicia nova
```

```
contagem
ret
```

```
;Delay de 100ms
```

```
delay_100ms:
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    rcall delay_10ms
    ret
```

```
delay_10s:
    ldi r23,0xC8
    loop3:
        clr r24
        clr r25
        loop4:
            adiw r24,1
            brne loop4 ;Se nao tiver a condicao FFFF volta a repetir a operacao
        dec r23
        brne loop3 ;Se nao for zero, apaga os registos contadores e inicia nova
contagem
ret
```