```
; ***************************************************************
; *             This program writes a message  on to the LCD   * ; * ; *
* ; ***************************************************************
.INCLUDE "m8535def.inc"
;Device =ATmega8535
.LIST
.def volt=r25;
.def curr=r24;
.equ  AtBCD0      =13         ;address of tBCD0
.equ    AtBCD1    =14         ;address of tBCD1;on 19=02-2008
.equ  AtBCD2      =15         ;address of tBCD2
.def  tBCD0 =r13  ;BCD value digits 1 and 0
.def  tBCD1 =r14  ;BCD value digits 3 and 2
.def  tBCD2 =r15  ;BCD value digit 4
.def  fbinL =r16  ;binary value Low byte
.def  fbinH =r17  ;binary value High byte
.def  cnt16a      =r18  ;loop counter
.def  tmp16a      =r19  ;temporary value
.DEF Capt_Count = r21
.DEF Prev_pf = r23
.DEF Pres_pf = r22
.DEF Cap_switch = r20
.EQU RESET = 0x0000
.EQU Start_EN = 0x08
.EQU MT_SWR_EN = 0x18
.EQU MT_DWR_ACK = 0x28
.EQU MT_SRD_EN = 0x40
.EQU MT_DRD_ACK = 0x50
.EQU Re_Start = 0x10
.EQU MT_LRD_ACK = 0x58
.EQU USR = 0x0B
.EQU UDRE1 = 0x05
.EQU RXC1 = 0x07
.CSEG
.ORG 0x0000
rjmp Start
.ORG 0x001
rjmp Ext_Int_0
.ORG 0x005
rjmp Cap_Int


;************** various subroutines for LCD display****************
key_debounce:clr r16
             ldi r17,$6a
           key:inc r16
                   brne key
                   dec r17
                   brne key
                 ret
cmd:    cbi portd,1
        cbi portd,5
        out portb,r16
        sbi portd,5
        nop
          nop
        nop
        nop
        nop
          cbi portd,5
          rcall delay1
            rcall delay1
            clr r16
            out portb,r16
          ret
lcdwr:  cbi portd,1
        cbi portd,5
        sbi portd,1
        out portb,r16
        sbi portd,5
        nop
        nop
        nop
        nop
        nop
        cbi portd,5
            rcall delay1
          rcall delay1
            clr r16
            out portb,r16
            ret
init_lcd:ldi r16,$38
         rcall cmd
         rcall delay1
             ldi r16,$0c
         rcall cmd
         rcall delay1
             ldi r16,$06
         rcall cmd
             rcall delay1
         ldi r16,$01
         rcall cmd
             rcall delay1
             //ldi r16,$80
         //rcall cmd
           //rcall delay1
             ret
;*******************************************************************************
**************
;*******************************************************************************
**************

USART_Flush:

sbis UCSRA, RXC

reti

in r16, UDR

rjmp USART_Flush
```

```
;******************************************************************************
**************
;******************************************************************************
**************



Error:
ldi r16,$86
rcall cmd
        ldi r16,$45
        rcall lcdwr
            ldi r16,$72
        rcall lcdwr
            ldi r16,$72
        rcall lcdwr
            ldi r16,$6f
        rcall lcdwr
        ldi r16,$72
        rcall lcdwr

        ret


TWINT_Check:in r16,TWCR
            sbrs r16,TWINT
                  rjmp TWINT_Check
                ret

Start_wr_en: ldi r16,(1<<TWINT)|(1<<TWSTA)|(1<<TWEN)|(1<<TWEA)
             out TWCR, r16
               rcall TWINT_Check
               in r16,TWSR
                   //mov r30,r16
                   //rcall ascii
                   //rcall delay
             //mov r16,r30
                   andi r16,$f8
             cpi r16,Start_EN
               breq next
               rjmp error
      next:  ret

Stop_wr_en: ldi r16,(1<<TWSTO)|(1<<TWEN)|(1<<TWEA)|(1<<TWINT)
            out TWCR,r16
                  rcall delay1
            rcall delay1
                  ret

I2C_WR_EN: out TWDR,r16
           ldi r16,(1<<TWINT)|(1<<TWEN)|(1<<TWEA)
             out TWCR,r16
               rcall TWINT_Check
             in r16,TWSR
               //mov r30,r16
               //rcall ascii
               //rcall delay
               //mov r16,r30
               andi r16,$f8
               cpi r16,MT_SWR_EN
               breq next1
           rjmp error
next1:     ret


I2C_RD_EN: out TWDR,r16
           ldi r16,(1<<TWINT)|(1<<TWEN)|(1<<TWEA)
             out TWCR, r16
           rcall TWINT_Check
             in r16,TWSR
           //mov r30,r16
               //rcall ascii
               //rcall delay
               //mov r16,r30
               andi r16,0xf8
           cpi r16,MT_SRD_EN
             breq next2
           rjmp error
next2:     ret

Data_Write: out TWDR,r16
              ldi r16,(1<<TWINT)|(1<<TWEN)|(1<<TWEA)
            out TWCR,r16
            rcall TWINT_Check
                in r16,TWSR
                  //mov r30,r16
                  //rcall ascii
                  //rcall delay
            //mov r16,r30
                  andi r16,$f8
            cpi r16,MT_DWR_ACK
            breq next3
              rjmp error
  next3:    ret

Data_Read: ldi r16,(1<<TWINT)|(1<<TWEN)|(1<<TWEA)
           out TWCR, r16
           rcall TWINT_Check
             in r16,TWSR
               //rcall ascii
               //rcall delay
               andi r16,0xf8
           cpi r16,MT_DRD_ACK
             breq next4
           rjmp error
 next4:    ret

Data_Last_Read: ldi r16,(1<<TWINT)|(1<<TWEN)
                out TWCR, r16
                rcall TWINT_Check
                  in r16,TWSR
                        //rcall ascii
                        //rcall delay
                    andi r16,0xf8
                cpi r16,MT_LRD_ACK
                  breq next5
                rjmp error
       next5:   ret

R_Start:ldi r16,(1<<TWINT)|(1<<TWSTA)|(1<<TWEN)|(1<<TWEA)
        out TWCR,r16
          rcall TWINT_Check
          in r16,TWSR
            //rcall ascii
            //rcall delay
        andi r16,0xf8
        cpi r16,Re_Start
        breq next6
        rjmp error
  next6:ret




;;*********************************************************************************************
;*********************************************************************************************
;*********************************************************************************************
;*********************************************************************************************

kva   :           ldi r16,$1;
                  rcall cmd;
                  mov r16,tBCD2
                  andi r16,$f0;
                  swap r16;
                  ldi r17,$30;
                  add r16,r17;
                  rcall lcdwr;
                  mov r16,tBCD2
                  andi r16,$0f;
                  ldi r17,$30;
                  add r16,r17;
                  rcall lcdwr
                  rcall delay1
                  mov r16,tBCD1
                  andi r16,$f0;
                  swap r16;
                  ldi r17,$30;
                  add r16,r17;
                  rcall lcdwr;

                  ldi r16,$2e;
                  rcall lcdwr

                  mov r16,tBCD1
                  andi r16,$0f;
                  ldi r17,$30;
                  add r16,r17;
                  rcall lcdwr;

                  rcall delay1

                  mov r16,tBCD0
                  andi r16,$f0;
                  swap r16;
                  ldi r17,$30;
                  add r16,r17;
                  rcall lcdwr;
                  rcall delay1;
                  mov r16,tBCD0
                  andi r16,$0f;
                  ldi r17,$30;
                  add r16,r17;
                  rcall lcdwr;
                  ret;



new_b:

;***** Subroutine Register Variables
;***** Code

/*ldi r16,$0f;
ldi r17,$0;*/

bin2BCD16:
            ldi   cnt16a,16   ;Init loop counter
            clr   tBCD2       ;clear result (3 bytes)
            clr   tBCD1
            clr   tBCD0
            clr   r31;ZH            ;clear ZH (not needed for AT90Sxx0x)
            st z,r31;

bBCDx_1:
            lsl   fbinL       ;shift input value
            rol   fbinH       ;through all bytes
            rol   tBCD0       ;
            rol   tBCD1
            rol   tBCD2
            dec cnt16a        ;decrement loop counter
          brne bBCDx_2        ;if counter not zero
            rcall kva;
            ret               ;   return

bBCDx_2:ldi r30,AtBCD2+1      ;Z points to result MSB + 1
bBCDx_3:ld  tmp16a,-Z
      ;dec r30;
      ;mov  tmp16a,r30;-Z     ;get (Z) with pre-decrement
;----------------------------------------------------------------
;For AT90Sxx0x, substitute the above line with:
;
;     dec   ZL
;     ld    tmp16a,Z
;
```

```
;----------------------------------------------------------------
      subi  tmp16a,-$03 ;add 0x03
      sbrc  tmp16a,3    ;if bit 3 not clear
      st          Z,tmp16a    ;     store back
      ld          tmp16a,Z    ;get (Z)
      subi  tmp16a,-$30 ;add 0x30
      sbrc  tmp16a,7    ;if bit 7 not clear
      st          Z,tmp16a    ;     store back
      ldi   r20,AtBCD0
      cp          r20,r30;
      ;cpi  Z,AtBCD0    ;done all three?
      brne  bBCDx_3           ;loop again if not
      rjmp  bBCDx_1

disply:           rcall bintobcd1
                  add r11,r17
                  mov r16,r11
            rcall lcdwr
                  add r12,r17
            mov r16,r12
            rcall lcdwr
                  add r18,r17
            mov r16,r18
            rcall lcdwr
                  rcall delay1
                  ret



;*******************************************************************************
**************
;*********************************************************
;*******************************************************************************
**************
;*******************************************************************************
**************
;*******************************************************************************
**************
latch1_send1:cli
            out portb,r16
            //rcall delay1
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  sbi portd,4
            nop
                  nop
                  nop
            //    rcall delay3sec
                  cbi portd,4
            cbi portd,4
                  rcall delay1
            //sei
                  ret

latch2_send2:cli
            out portb,r16
            //rcall delay1
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  sbi portd,7
            nop
                  nop
                  nop
            //    rcall delay3sec
                  cbi portd,7
            cbi portd,7
                  rcall delay
            //sei
                  ret


pass:
            mov r16,r21
                  rcall cmd

                  //ldi r16,$2a
                  //rcall lcdwr

                  //ldi r21,$c7

                  mov r16,r19

                  andi r16,$0f

                  mov r17,r30
                  mov r18,r31


                  rcall ertyyy


            inc r30

                  inc r27
                  inc r21
                  //inc r17

            ldi r19,$30
```

```
ret


pass_read:

            mov r16,r21
                  rcall cmd
                  ldi r16,$2a
                  rcall lcdwr

                                       ;first enter
                  mov r16,r19

                  andi r16,$0f

                  mov r15,r16

mov r17,r30
mov r18,r31

//          ldi r18,$01
//          ldi r17,$30

                rcall rtyuu


  //           mov r17,r16
      //        rcall rtyuu

            cp r16,r15
                  brne jeter


                  inc r11

jeter:

            inc r30
      //      mov r16,r17
        //    ldi r18,$01
          //  ldi r17,$30
                  //rcall ertyyy

 inc r27

inc r21

           ldi r19,$30

ret


latch1_send:cli
            out portb,r16
            //rcall delay1
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  sbi portd,4
            nop
                  nop
                  nop
                  rcall delay3sec
                  cbi portd,4
            cbi portd,4
                  rcall delay
            sei
                  ret

latch2_send:cli
            out portb,r16
            //rcall delay1
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  nop
                  sbi portd,7
            nop
                  nop
                  nop
                  rcall delay3sec
                  cbi portd,7
            cbi portd,7
                  rcall delay
            sei
                  ret


Time_Stamp:

       rcall Start_wr_en

         ldi r16,$d0
         rcall I2C_WR_EN

       ldi r16,$00
       rcall Data_Write
       rcall R_Start

       ldi r16,$d1
         rcall I2C_RD_EN

       rcall Data_Read
         in r16,TWDR
       mov r6,r16             ;Seconds

         rcall Data_Read
         in r16,TWDR
       mov r5,r16             ;Minutes

         rcall Data_Read
         in r16,TWDR
       mov r4,r16             ;Hours

         rcall Data_Read
         in r16,TWDR
       mov r7,r16             ;Day

       rcall Data_Read
         in r16,TWDR
       mov r8,r16             ;Date

       rcall Data_Read
         in r16,TWDR
       mov r2,r16             ;Month

         rcall Data_Last_Read
         in r16,TWDR
       mov r3,r16             ;Year

       rcall Stop_wr_en

         ret

check_pf_limits:
;r11,r12,r30,r31
ldi r18,$01

ldi r17,$3f
rcall rtyuu
mov r30,r16       ;lead /lag

ldi r17,$41
rcall rtyuu
mov r31,r16       ;lead /lag

ldi r17,$3e
rcall rtyuu
mov r11,r16       ;lower pf value

ldi r17,$40
rcall rtyuu
mov r12,r16       ;upper pf value
```

```
cp r30,r31
brne not_equal
cp r30,r26
brne exit2   ;take regular action
;both are same then
cp Pres_pf,r11
brsh upr_test
rjmp exit2   ;take regular action

upr_test:
cp r12,Pres_pf
brsh exit1   ;all ok
rjmp exit3   ;take inverse action

not_equal:

cpi r30,$00
brne lead_first
;lag first
cpi r26,$00
brne upr_test1

rjmp one111

upr_test1:

rjmp two222

lead_first:
cpi r26,$00
brne upr_test2

rjmp two222

upr_test2:
rjmp one111

one111:
cp Pres_pf,r11
brsh exit1   ;all ok
rjmp exit2   ;take regular action            1111

two222:
cp Pres_pf,r12
brsh exit1
rjmp exit2   ;take regular action           2222


exit1:
ldi r16,$ab
ret              ; all ok ; repeat
exit2:
ldi r16,$cd
ret              ; take regular action
exit3:
ldi r16,$ef
ret              ; take inverse action
```

```
Send_Data:
//clr r30
//ldi r31,$fe
clr r30
clr r31

Repeat1:
         rcall Data_Read
         in r16,TWDR

USART_Transmit:

       sbis USR,UDRE1
       rjmp USART_Transmit
         out UDR,r16

inc r30
brne Repeat1
inc r31
brne Repeat1

rcall Stop_wr_en
rcall delay

ret






delay_seconds:mov r8,r16
tyr:          rcall delay
              dec r8
                  brne tyr
              ret

delay_minutes:mov r9,r16
ertt:         rcall delay_1min
              dec r9
                    brne ertt
                    ret


delay_1min:ldi r16,$3c
           mov r8,r16
dfggg:     rcall delay
           dec r8
               brne dfggg
               ret
```

```
delay:      ldi r16,$07
            mov r2,r16
              clr r3
d2:         ldi r16,$ba
            mov r7,r16
d1:         inc r3
            brne d1
            dec r7
            brne d1
            dec r2
            brne d2
            ret

movey:mov r16,r19
                   andi r16,$0f
                   ret

curr_min:

clr r3
clr r4
ldi r17,$09

battr1:
ldi r16,$61
out admux,r16

ldi r16,$c0
out adcsra,r16

//rcall delay1

baartt1:
ldi r16,$10
in r18,adcsra
andi r18,$10
cp r18,r16
brne baartt1

in r16,adch
add r3,r16
brcc sipa2

inc r4

sipa2:
dec r17
brne battr1

mov r25,r4
mov r24,r3

clr r16

yyaer:
```

```
inc r16
sbiw r25:r24,$09
brcc yyaer

dec r16

ret

switch_delay_inc:

ldi r18,$01
ldi r17,$1b
rcall rtyuu

cpi r16,$00
breq seconds_01
rcall delay_minutes

seconds_01:

ldi r18,$01
ldi r17,$1c
rcall rtyuu

cpi r16,$00
breq exit67
rcall delay_seconds

exit67:ret

switch_delay_dec:

ldi r18,$01
ldi r17,$1e
rcall rtyuu

cpi r16,$00
breq seconds_011
rcall delay_minutes

seconds_011:

ldi r18,$01
ldi r17,$1f
rcall rtyuu

cpi r16,$00
breq exit671
rcall delay_seconds

exit671:ret


dicval:cpi r19,$30
       breq lavelrt
       dec r19
```

```
therrr:
       mov r16,r21
         rcall cmd
       mov r16,r19
         rcall lcdwr
       mov r16,r21
         rcall cmd
       rjmp keywait


lavelrt: ldi r19,$39
         rjmp therrr




delay1:clr r16          ;255
loop2: inc r16          ;255
       brne loop2
       ret

delay3sec:ldi r19,$01
io:       rcall delay
              dec r19
              brne io
          ret

/*
ascii:mov r15,r16
      ldi r16,$07
      ldi r17,$30
      mov r18,r15
        mov r19,r15
      andi r18,$0f
        swap r19
        andi r19,$0f
        cpi r18,$0a

        brcs ii
        add r18,r16
ii:   add r18,r17
      cpi r19,$0a

        brcs iii
        add r19,r16
iii:  add r19,r17
      mov r16,r19
        rcall lcdwr
        mov r16,r18
        rcall lcdwr
        mov r16,r15
      ret

*/

ertyyy:
```

```
sbic EECR,EEWE
rjmp ertyyy
out EEARH, r18
out EEARL, r17
out EEDR,r16
sbi EECR,EEMWE
sbi EECR,EEWE

ret

delay_ww:clr r18
         clr r19
dw:      inc r18
             brne dw
             inc r19
             brne dw
             ret

delay2:ldi r18,$02
       clr r17
         clr r16
loop3: inc r17
         brne loop3
loop4: inc r16
       brne loop3
         dec r18
       brne loop4
       ret
//bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd
bintobcd
//bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd
bintobcd
//bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd  bintobcd
bintobcd
bintobcd:   mov r18,r16
                  clr r16
            ldi r17,$64
                  ldi r19,$0a
      nxt1a:inc r16
                  sub r18,r17
            brcc nxt1a
                  dec r16
            mov r11,r16
                  add r18,r17
            clr r16
 nxt21:     inc r16
                  sub r18,r19
                  brcc nxt21
            dec r16
              mov r12,r16
                add r18,r19
              ldi r17,$30
            ret

bintobcd1:  mov r18,r16
                  clr r16
            ldi r17,$32
```

```
                  ldi r19,$05
     nxt1a1:inc r16
                  sub r18,r17
            brcc nxt1a1
                  dec r16
           mov r11,r16
                  add r18,r17
            clr r16
       nxt211:inc r16
                  sub r18,r19
                  brcc nxt211
            dec r16
                mov r12,r16
                add r18,r19
                add r18,r18
                ldi r17,$30
                  ret



intcval:
       cpi r19,$39
       breq lavelrt1
         inc r19
therrr1:
       mov r16,r21
         rcall cmd
         mov r16,r19
         rcall lcdwr
       mov r16,r21
         rcall cmd
       rjmp keywait
lavelrt1:ldi r19,$30
         rjmp therrr1


/*
bintobcd2:  mov r18,r16
                  clr r16
            ldi r17,$64
                  ldi r19,$0a
     nxt1b1:inc r16
                  sub r18,r17
            brcc nxt1b1
                  dec r16
            mov r11,r16
                  add r18,r17
            clr r16
      nxtb211:inc r16
                  sub r18,r19
                  brcc nxtb211
            dec r16
                  mov r12,r16
                  add r18,r19
                  ldi r17,$30
                  add r11,r17
                  mov r16,r11
                  rcall lcdwr
                  add r12,r17
                  mov r16,r12
                  rcall lcdwr
                  add r18,r17
            mov r16,r18
                  rcall lcdwr
            ret

*/

rtyuu:
sbic EECR,EEWE
rjmp rtyuu
out EEARH, r18
out EEARL, r17
sbi EECR,EERE
in r16,EEDR
ret

//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf

EE_Ad_inc:
                ldi r18,$01
            ldi r17,$2e
            rcall rtyuu
            mov r30,r16

                  ldi r18,$01
            ldi r17,$2e
            inc r30
                  mov r16,r30
                  rcall ertyyy

                ldi r18,$01
            ldi r17,$2f
            rcall rtyuu
            mov r31,r16

            cpi r30,$ff
                  brne continue2

                  inc r31

            ldi r18,$01
            ldi r17,$2f
            //inc r31
                  mov r16,r31
                  rcall ertyyy

                cpi r31,$ff
                  brne continue2

            ldi r17,$2d
                  ldi r18,$01
                  rcall rtyuu
            cpi r16,$a2
            brne continue1

// ex_eeprom ful zaleli aahe


            ldi r18,$01
            ldi r17,$2e
        clr r16
            rcall ertyyy

            ldi r17,$2f
        clr r16
            rcall ertyyy

            ldi r17,$2d
        ldi r16,$a0
            rcall ertyyy




            ldi r17,$2a
            ldi r18,$01
        ldi r16,$01
            rcall ertyyy

          rcall Stop_wr_en
            rcall delay
            ret

continue1:
            ldi r17,$2d
                  ldi r18,$01
                  ldi r16,$a2
                  rcall ertyyy


continue2:   ret


EE_SET_AD:

                  rcall Start_wr_en

                  ldi r17,$2d
                  ldi r18,$01
                  rcall rtyuu

              rcall I2C_WR_EN

                ldi r17,$2e
                  ldi r18,$01
                  rcall rtyuu
            mov r30,r16

                ldi r17,$2f
```

```
                  ldi r18,$01
                  rcall rtyuu
            mov r31,r16

         mov r16,r31
       rcall Data_Write

         mov r16,r30
       rcall Data_Write


;Write Protocol word

                  ldi r16,$00
                  rcall Data_Write
            ldi r16,$00
                  rcall Data_Write
                  ldi r16,$00
                  rcall Data_Write
                  ldi r16,$00
                  rcall Data_Write
                  ldi r16,$00
                  rcall Data_Write
                  ldi r16,$00
                  rcall Data_Write

;Time Write

            mov r16,r8
                  rcall Data_write
            mov r16,r2
                  rcall Data_write
                  mov r16,r3
                  rcall Data_write
                  mov r16,r4
                  rcall Data_write
                  mov r16,r5
                  rcall Data_write


            ret





//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf
//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf
//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf
//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf
//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf
```

```
//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf
//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf
//Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf  Measure_pf
Measure_pf

Measure_pf:

           ldi r16,$08
           mov r6,r16
               clr r16
.              mov r4,r16
           mov r5,r16

start_int: nop
           nop
           ldi r16,$20
           out TIMSK,r16        ;
         nop
               nop
           nop
               rcall delay_ww
           nop
           nop
           nop
           clr r16
           out TIMSK,r16
             mov r16,r14
             cpi r16,$14
             brlo exita
           clr r16
               rjmp erdfyy

               //rjmp start_int

exita:
     mov r25,r14
       mov r24,r13
;*********************************************************
    clr r19
log:inc r19
      sbiw r25:r24,$14
    brcc log
      dec r19
    adiw r25:r24,$14
            clr r17
                  out EEARH,r17
            out EEARL,r19
            sbi EECR,EERE

                  in r16,EEDR

            cpi r16,$65

                  brsh start_int
            cpi r16,$30

                  brcs start_int




erdfyy:     add r4,r16
                  brcc label00b1
                inc r5
label00b1:  dec r6
                  brne label0020
                  rjmp label000a1
label0020:  rjmp start_int
;***************************************************
label000a1: movw r25:r24,r5:r4
              clr r16
log1:       inc r16
                  sbiw r25:r24,$08
                  brcc log1
                  dec r16
                  mov Pres_pf,r16

;***************************************************
label00011: rcall bintobcd
;********************************************************
label00101: ldi r16,$01
              rcall cmd
              rcall delay1

                  ldi r17,$30
                  add r11,r17
                  mov r16,r11
                  rcall lcdwr

                  ldi r16,$2e
            rcall lcdwr

                  add r12,r17
            mov r16,r12
                  rcall lcdwr

                  add r18,r17
                  mov r16,r18
                  rcall lcdwr

            ldi r17,$2a
                  ldi r18,$01
                  rcall rtyuu
                  cpi r16,$00
                  brne sjkdhfkjhj

                  rcall Time_Stamp
            rcall ee_set_ad

                mov r16,Pres_pf
```

```
                  rcall Data_Write

            rcall ee_ad_inc




sjkdhfkjhj: rcall line_vltge1


                rcall line_Curr1

                  rcall lead_lag


            ldi r17,$2b
                  ldi r18,$01
                  rcall rtyuu

             cpi r16,$01

               brne kdfjkljdg123

           ldi r16,$86
               rcall cmd
               ldi r16,$4d
               rcall lcdwr
               ldi r16,$41
               rcall lcdwr
               ldi r16,$4e
               rcall lcdwr

           rjmp ertwer


kdfjkljdg123:

                  rcall delay
                  rcall delay
                  //rcall delay
                  //rcall delay

ertwer:           ret
;********************************************************


//Original routine

lead_lag:   in r26,pina
                  sbrc r26,7
                  rjmp lead_lag
upward:           in r26,pina
            sbrs r26,7
            rjmp upward
                  andi r26,$40
                  cpi r26,$40
                  brne lag
            ldi r16,$c0
            rcall cmd
                  ldi r16,$4c
                  rcall lcdwr
            ldi r16,$65
                  rcall lcdwr
            ldi r16,$61
                  rcall lcdwr
            ldi r16,$64
                  rcall lcdwr
            clr r26         ;00000
                  inc r26
            ret
lag:              ldi r16,$c0
            rcall cmd
              ldi r16,$4c
                  rcall lcdwr
            ldi r16,$61
                  rcall lcdwr
            ldi r16,$67
                  rcall lcdwr
            clr r26        ;00000000000
                  ret


/*

//Original routine

lead_lag:   in r26,pina
                  sbrc r26,7
                  rjmp lead_lag
upward:           in r26,pina
            sbrs r26,7
            rjmp upward
                  andi r26,$40
                  cpi r26,$40
                  brne lag
            ldi r16,$c0
            rcall cmd
                  ldi r16,$4c
                  rcall lcdwr
            ldi r16,$65
                  rcall lcdwr
            ldi r16,$61
                  rcall lcdwr
            ldi r16,$64
                  rcall lcdwr
            clr r26         ;00000
                  inc r26
            ret
lag:              ldi r16,$c0
            rcall cmd
              ldi r16,$4c
                  rcall lcdwr
            ldi r16,$61
                  rcall lcdwr
            ldi r16,$67
                  rcall lcdwr
            clr r26        ;00000000000
                  ret

*/

Write_data:

            ldi r17,$2a
                  ldi r18,$01
                  rcall rtyuu
                  cpi r16,$00
                  brne sjkdhfkjhj2


mov r16,r21

rcall Data_Write

rcall ee_ad_inc

sjkdhfkjhj2:

ret




////////////////////////////////////////////////////
swapvalue:     cpi r18,$01
                     brne goto1
                   ldi r18,$80
                     ret
goto1:         cpi r18,$02
               brne goto2
                     ldi r18,$40
                     ret
goto2:         cpi r18,$03
               brne goto3
                     ldi r18,$c0
                     ret
goto3:         cpi r18,$04
               brne goto4
                     ldi r18,$20
                     ret
goto4:         cpi r18,$05
               brne goto5
                     ldi r18,$a0
                     ret
goto5:         cpi r18,$06
               brne goto6
                     ldi r18,$60
                     ret
goto6:         cpi r18,$07
               brne goto7
                     ldi r18,$e0
                     ret
goto7:         cpi r18,$08
               brne goto8
                     ldi r18,$10
                     ret
goto8:         cpi r18,$09
               brne goto9
                     ldi r18,$90
                     ret
goto9:         cpi r18,$0a
               brne goto10
                     ldi r18,$50
                     ret
goto10:        cpi r18,$0b
               brne goto11
                     ldi r18,$d0
                     ret
goto11:        cpi r18,$0c
               brne goto12
                     ldi r18,$30
                     ret
goto12:        cpi r18,$0d
               brne goto13
                     ldi r18,$b0
                     ret
goto13:        cpi r18,$0e
               brne goto14
                     ldi r18,$70
                     ret
goto14:        cpi r18,$0f
               brne goto15
                     ldi r18,$f0
                     ret
goto15:        ret
; FIFO incrememnt capapcitor

Incr_Capacitor:
ldi r18,$01
ldi r17,$2b
rcall rtyuu
cpi r16,$01
breq labelg001


ldi r17,$3c

rcall rtyuu
com r16

or r28,r16

                 ldi r18,$01
                 ldi r17,$11

                   rcall rtyuu

                   mov r29,r16
                   cpi r29,$00

                   breq labelg001

                         cpi r29,$01

                         brne lavelio11

                         rjmp hexswitchinc

       lavelio11:  rjmp Incr_LIFO


labelg001:       ldi r16,$ff
                 cpse r28,r16

                         rjmp switchthis
                 rjmp switchnext

switchthis:      mov r16,r28
                   andi r16,$80
                   cpi r16,$00
                         brne labelj002
                         sbr r28,$80
                         rjmp lablelatch1en
labelj002:       mov r16,r28
                   andi r16,$40
                   cpi r16,$00
                         brne labelj003
                         sbr r28,$40
                         rjmp lablelatch1en
       labelj003:  mov r16,r28
                   andi r16,$20
                   cpi r16,$00
                         brne labelj004
                         sbr r28,$20
                         rjmp lablelatch1en
       labelj004:  mov r16,r28
                   andi r16,$10
                   cpi r16,$00
                         brne labelj005
                         sbr r28,$10
                         rjmp lablelatch1en
     labelj005:  mov r16,r28
                   andi r16,$08
                   cpi r16,$00
                         brne labelj006
                         sbr r28,$08
                         rjmp lablelatch1en
  labelj006:     mov r16,r28
                   andi r16,$04
                   cpi r16,$00
                         brne labelj007
                         sbr r28,$04
                         rjmp lablelatch1en
     labelj007:  mov r16,r28
```

```
                   andi r16,$02
                   cpi r16,$00
                         brne labelj008
                         sbr r28,$02
                         rjmp lablelatch1en
     labelj008:  mov r16,r28
                   andi r16,$01
                   cpi r16,$00
                         brne labelexit
                         sbr r28,$01
                         rjmp lablelatch1en

labelexit:      ret


       switchnext:

ldi r17,$3d
ldi r18,$01
rcall rtyuu
com r16
andi r16,$f0

or r20,r16


       ldi r16,$f0
                   cpse r16,r20
                         rjmp labelk001
                   rjmp labelexit

        labelk001:
                   mov r16,r20
                 andi r16,$80
                   cpi r16,$00
                         brne labelj009
                         sbr r20,$80
                         rjmp lablelatch2en

       labelj009:  mov r16,r20
                 andi r16,$40
                   cpi r16,$00
                         brne labelj010
                         sbr r20,$40
                         rjmp lablelatch2en

       labelj010:  mov r16,r20
                 andi r16,$20
                   cpi r16,$00
                         brne labelj011
                         sbr r20,$20
                         rjmp lablelatch2en

       labelj011:  mov r16,r20
                 andi r16,$10
                   cpi r16,$00
                         brne labelj012
                         sbr r20,$10
                         rjmp lablelatch2en

       labelj012:  mov r16,r20
                 andi r16,$08
                   cpi r16,$00
                         brne labelj013
                         sbr r20,$08
                         rjmp lablelatch2en

     labelj013:  mov r16,r20
                 andi r16,$04
                   cpi r16,$00
                         brne labelj014
                         sbr r20,$04
                         rjmp lablelatch2en

       labelj014:  mov r16,r20
                 andi r16,$02
                   cpi r16,$00
                         brne labelj015
                         sbr r20,$02
                         rjmp lablelatch2en

       labelj015:  mov r16,r20
                 andi r16,$01
                   cpi r16,$00
                         brne labelexit
                         sbr r20,$01
                         rjmp lablelatch2en

lablelatch1en:
ldi r17,$2b
ldi r18,$01
rcall rtyuu
cpi r16,$00
brne manualmode
ldi r17,$3c
ldi r18,$01
rcall rtyuu
and r28,r16
mov r16,r28
rcall latch1_send
ret
manualmode:
mov r16,r28
rcall latch1_send1
ret

lablelatch2en:
ldi r17,$2b
ldi r18,$01
rcall rtyuu
cpi r16,$00
brne manualmode1
ldi r17,$3d
ldi r18,$01
```

```
rcall rtyuu
and r20,r16
mov r16,r20
rcall latch2_send
ret
manualmode1:
mov r16,r20
rcall latch2_send2
ret


; FIFO decrememnt capapcitor

Decr_Capacitor:
/*
ldi r17,$3c
ldi r18,$01
rcall rtyuu
and r28,r16
*/
                   ldi r18,$01

                     ldi r17,$11

                  rcall rtyuu

                    mov r29,r16

                  cpi r29,$00

                   breq labelg002

                   cpi r29,$01

                         brne lavelkio11

                         rjmp hexswitchdec

       lavelkio11: rjmp decr_LIFO

labelg002 :      clr r16
                 cpse r16,r28
                       rjmp switchthis1
                 rjmp switchnext1

switchthis1:   mov r16,r28
               andi r16,$80
               cpi r16,$80
               brne labelk002
               cbr r28,$80
               rjmp lablelatch1en
labelk002 :    mov r16,r28
               andi r16,$40
               cpi r16,$40
               brne labelk003
               cbr r28,$40
               rjmp lablelatch1en
```

```
labelk003 :    mov r16,r28
               andi r16,$20
               cpi r16,$20
               brne labelk004
               cbr r28,$20
               rjmp lablelatch1en
labelk004 :    mov r16,r28
               andi r16,$10
               cpi r16,$10
               brne labelk005
               cbr r28,$10
               rjmp lablelatch1en
labelk005 :    mov r16,r28
               andi r16,$08
               cpi r16,$08
               brne labelk006
               cbr r28,$08
               rjmp lablelatch1en
labelk006 :    mov r16,r28
               andi r16,$04
               cpi r16,$04
               brne labelk007
               cbr r28,$04
               rjmp lablelatch1en

labelk007 :    mov r16,r28
               andi r16,$02
               cpi r16,$02
               brne labelk008
               cbr r28,$02
               rjmp lablelatch1en


labelexit234:ret

labelk008 :    mov r16,r28
               andi r16,$01
               cpi r16,$01
               brne labelexit234
               cbr r28,$01
               rjmp lablelatch1en

switchnext1:
/*
ldi r17,$3d
ldi r18,$01
rcall rtyuu
and r20,r16
*/




               clr r16
               cpse r20,r16
                     rjmp lablen001
                     rjmp labelexit
```

```
lablen001:     mov r16,r20
               andi r16,$80
               cpi r16,$80
               brne lablen002
               cbr r20,$80
               rjmp lablelatch2en

lablen002:     mov r16,r20
               andi r16,$40
               cpi r16,$40
               brne lablen003
               cbr r20,$40
               rjmp lablelatch2en

lablen003:     mov r16,r20
               andi r16,$20
               cpi r16,$20
               brne lablen004
               cbr r20,$20
               rjmp lablelatch2en

lablen004:     mov r16,r20
               andi r16,$10
               cpi r16,$10
               brne lablen005
               cbr r20,$10
               rjmp lablelatch2en

lablen005:     mov r16,r20
               andi r16,$08
               cpi r16,$08
               brne lablen006
               cbr r20,$08
               rjmp lablelatch2en

lablen006:     mov r16,r20
               andi r16,$04
               cpi r16,$04
               brne lablen007
               cbr r20,$04
               rjmp lablelatch2en

lablen007:     mov r16,r20
               andi r16,$04
               cpi r16,$04
               brne lablen008
               cbr r20,$04
               rjmp lablelatch2en

lablen008:     mov r16,r20
               andi r16,$02
               cpi r16,$02
               brne lablen009
               cbr r20,$02
               rjmp lablelatch2en
```

```
lablen009:     mov r16,r20
               andi r16,$01
               cpi r16,$01
               brne labelexit1
               cbr r20,$01
               rjmp lablelatch2en

labelexit1:    ret


; hex incrememnt capapcitor
hexswitchinc:  ldi r16,$ff
               cpse r28,r16
               rjmp switthis
               rjmp switchnxt

switthis:      inc r28
                     mov r16,r28
               mov r17,r28
                     andi r16,$0f
                     andi r17,$f0
               swap r17
               mov r18,r16
                     rcall swapvalue
                     mov r16,r18
                     mov r18,r17
                     rcall swapvalue
                     mov r17,r18
                     swap r17
                     or r17,r16
                     rjmp lablelatch2hexen

switchnxt:     ldi r16,$f0
               cpse r20,r16
               rjmp switthis123
               rjmp labelexit1

switthis123:   inc r20
                     mov r16,r20
               mov r17,r20
                     andi r16,$0f
                     andi r17,$f0
               swap r17
               mov r18,r16
                     rcall swapvalue
                     mov r16,r18
                     mov r18,r17
                     rcall swapvalue
                     mov r17,r18
                     swap r17
                     or r17,r16
                     rjmp lablelatch1hexen


; hex decrememnt capapcitor

hexswitchdec:  clr r16
```

```
               cpse r20,r16
               rjmp switthis1
               rjmp switchnxt345

switthis1:     dec r20
                     mov r16,r20
               mov r17,r20
                     andi r16,$0f
                     andi r17,$f0
               swap r17
               mov r18,r16
                     rcall swapvalue
                     mov r16,r18
                     mov r18,r17
                     rcall swapvalue
                     mov r17,r18
                     swap r17
                     or r17,r16
                     rjmp lablelatch1hexen

switchnxt345:  clr r16
               cpse r28,r16
               rjmp switthis1345
               rjmp labelexit1


switthis1345:  dec r28
                     mov r16,r28
               mov r17,r28
                     andi r16,$0f
                     andi r17,$f0
               swap r17
               mov r18,r16
                     rcall swapvalue
                     mov r16,r18
                     mov r18,r17
                     rcall swapvalue
                     mov r17,r18
                     swap r17
                     or r17,r16
                     rjmp lablelatch2hexen

lablelatch1hexen:
mov r16,r17
rcall latch2_send
ret

lablelatch2hexen:
mov r16,r17
rcall latch1_send
ret

; LIFO incrememnt capapcitor
Incr_LIFO:     ldi r16,$ff
               cpse r28,r16
               rjmp switthis12
               rjmp switchnxt12
```

```
switthis12:
                 mov r16,r28
                   andi r16,$80
                   cpi r16,$00
                         brne labeljj002
                         sbr r28,$80
                         rjmp lablelatch1en

       labeljj002: mov r16,r28
                   andi r16,$40
                   cpi r16,$00
                         brne labeljj003
                         sbr r28,$40
                         rjmp lablelatch1en

       labeljj003: mov r16,r28
                   andi r16,$20
                   cpi r16,$00
                         brne labeljj004
                         sbr r28,$20
                         rjmp lablelatch1en

       labeljj004:  mov r16,r28
                   andi r16,$10
                   cpi r16,$00
                         brne labeljj005
                         sbr r28,$10
                         rjmp lablelatch1en

     labeljj005:  mov r16,r28
                   andi r16,$08
                   cpi r16,$00
                         brne labeljj006
                         sbr r28,$08
                         rjmp lablelatch1en

     labeljj006:  mov r16,r28
                   andi r16,$04
                   cpi r16,$00
                         brne labeljj007
                         sbr r28,$04
                         rjmp lablelatch1en

     labeljj007:  mov r16,r28
                   andi r16,$02
                   cpi r16,$00
                         brne labeljj008
                         sbr r28,$02
                         rjmp lablelatch1en

     labeljj008: mov r16,r28
                   andi r16,$01
                   cpi r16,$00
                         brne labeljexit
                         sbr r28,$01
                         rjmp lablelatch1en
```

```
switchnxt12:   ldi r16,$f0
               cpse r20,r16
               rjmp switthis1245
               rjmp labeljexit
switthis1245:
                 mov r16,r20
                   andi r16,$80
                   cpi r16,$00
                         brne labeljjq002
                         sbr r20,$80
                         rjmp lablelatch2en

       labeljjq002:  mov r16,r20
                   andi r16,$40
                   cpi r16,$00
                         brne labeljjq003
                         sbr r20,$40
                         rjmp lablelatch2en

       labeljjq003:  mov r16,r20
                   andi r16,$20
                   cpi r16,$00
                         brne labeljjq004
                         sbr r20,$20
                         rjmp lablelatch2en

       labeljjq004:mov r16,r20
                   andi r16,$10
                   cpi r16,$00
                         brne labeljjq005
                         sbr r20,$10
                         rjmp lablelatch2en

     labeljjq005:  mov r16,r20
                   andi r16,$08
                   cpi r16,$00
                         brne labeljjq006
                         sbr r20,$08
                         rjmp lablelatch2en

     labeljjq006:  mov r16,r20
                   andi r16,$04
                   cpi r16,$00
                         brne labeljjq007
                         sbr r20,$04
                         rjmp lablelatch2en

     labeljjq007:  mov r16,r20
                   andi r16,$02
                   cpi r16,$00
                         brne labeljjq008
                         sbr r20,$02
                         rjmp lablelatch2en

     labeljjq008: mov r16,r20
                   andi r16,$01
                   cpi r16,$00
                         brne labeljexit
                         sbr r20,$01
                         rjmp lablelatch2en


 labeljexit:     ret


; LIFO decrememnt capapcitor

Decr_LIFO:     clr r16
               cpse r20,r16
               rjmp switchnxt34
               rjmp switthis134

switchnxt34:   mov r16,r20
               andi r16,$01
               cpi r16,$01
               brne labelkkz002
               cbr r20,$01
               rjmp lablelatch2en

labelkkz002 :   mov r16,r20
               andi r16,$02
               cpi r16,$02
               brne labelkkz003
               cbr r20,$02
               rjmp lablelatch2en

labelkkz003 :    mov r16,r20
               andi r16,$04
               cpi r16,$04
               brne labelkkz004
               cbr r20,$04
               rjmp lablelatch2en

labelkkz004 :    mov r16,r20
               andi r16,$08
               cpi r16,$08
               brne labelkkz005
               cbr r20,$08
               rjmp lablelatch2en

labelkkz005 :    mov r16,r20
               andi r16,$10
               cpi r16,$10
               brne labelkkz006
               cbr r20,$10
               rjmp lablelatch2en

labelkkz006 :    mov r16,r20
               andi r16,$20
               cpi r16,$20
               brne labelkkz007
               cbr r20,$20
               rjmp lablelatch2en
```

```
labelkkz007 :    mov r16,r20
               andi r16,$40
               cpi r16,$40
               brne labelkkz008
               cbr r20,$40
               rjmp lablelatch2en

labelkkz008 :    mov r16,r20
               andi r16,$80
               cpi r16,$80
               brne labelkexit
               cbr r20,$80
               rjmp lablelatch2en


switthis134:   clr r16
               cpse r28,r16
               rjmp switthis8134
               rjmp labeljexit

switthis8134:  mov r16,r28
               andi r16,$01
               cpi r16,$01
               brne labelkk002
               cbr r28,$01
               rjmp lablelatch1en

labelkk002 :    mov r16,r28
               andi r16,$02
               cpi r16,$02
               brne labelkk003
               cbr r28,$02
               rjmp lablelatch1en

labelkk003 :    mov r16,r28
               andi r16,$04
               cpi r16,$04
               brne labelkk004
               cbr r28,$04
               rjmp lablelatch1en

labelkk004 :    mov r16,r28
               andi r16,$08
               cpi r16,$08
               brne labelkk005
               cbr r28,$08
               rjmp lablelatch1en

labelkk005 :    mov r16,r28
               andi r16,$10
               cpi r16,$10
               brne labelkk006
               cbr r28,$10
               rjmp lablelatch1en

labelkk006 :    mov r16,r28
```

```
               andi r16,$20
               cpi r16,$20
               brne labelkk007
               cbr r28,$20
               rjmp lablelatch1en

labelkk007 :    mov r16,r28
               andi r16,$40
               cpi r16,$40
               brne labelkk008
               cbr r28,$40
               rjmp lablelatch1en

labelkk008 :    mov r16,r28
               andi r16,$80
               cpi r16,$80
               brne labelkexit
               cbr r28,$80
               rjmp lablelatch1en

labelkexit:   ret


//voltage  voltage  voltage  voltage  voltage  voltage  voltage  voltage
voltage
//voltage  voltage  voltage  voltage  voltage  voltage  voltage  voltage
voltage
//voltage  voltage  voltage  voltage  voltage  voltage  voltage  voltage
voltage


line_vltge:

clr r3
clr r4
ldi r17,$09

bat1:

ldi r16,$60
out admux,r16

ldi r16,$c0
out adcsra,r16

//rcall delay1

batt1:
ldi r16,$10
in r18,adcsra
andi r18,$10
cp r18,r16
brne batt1

in r16,adch
add r3,r16
brcc skip2
```

```
inc r4
skip2:
dec r17
brne bat1

mov r25,r4
mov r24,r3

clr r16

yy:
inc r16
sbiw r25:r24,$09
brcc yy

dec r16
rcall bintobcd1

ldi r16,$01
rcall cmd
rcall delay1

ldi r17,$30

add r11,r17
mov r16,r11
rcall lcdwr

add r12,r17
mov r16,r12
rcall lcdwr

add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$56
rcall lcdwr
rcall delay

rjmp line_vltge
////////////////////////////////////////////////////////////////////////////////
//////

//current current  current  current  current  current  current  current  current
//current  current  current  current  current  current  current  current
current
//current  current  current  current  current  current  current  current
current


line_Curr:
ldi r16,$01
rcall cmd
rcall delay1

clr r3
clr r4
ldi r17,$09

batw1:
ldi r16,$61
out admux,r16

ldi r16,$c0
out adcsra,r16

//rcall delay1

batt11:
ldi r16,$10
in r18,adcsra
andi r18,$10
cp r18,r16
brne batt11

in r16,adch
add r3,r16
brcc skip21
inc r4

skip21:

dec r17
brne batw1

mov r25,r4
mov r24,r3

clr r16

yyy:
inc r16
sbiw r25:r24,$09
brcc yyy



dec r16
mov r19,r16


ert4y1:
ldi r18,$01
ldi r17,$10
rcall rtyuu

mov r17,r16
cpi r17,$00
breq werwer
rjmp werwer1
werwer:

ldi r18,$01
ldi r17,$1a
rcall rtyuu

mov r17,r16

mul r19,r17

mov r16,r0

rcall bintobcd1

ldi r17,$30

add r11,r17
mov r16,r11

cpi r16,$30

breq sfdgj

mov r16,r11
rcall lcdwr

sfdgj:

add r12,r17
mov r16,r12
rcall lcdwr

//ldi r16,$2e
//rcall lcdwr

add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$41
rcall lcdwr

rcall delay

rjmp line_Curr


werwer1:

mul r19,r17

mov r16,r0

rcall bintobcd1


ldi r16,$01
rcall cmd
rcall delay1
```

```
ldi r17,$30

add r11,r17

mov r16,r11
cpi r16,$30
breq sdfss

mov r16,r11
rcall lcdwr

sdfss:
add r12,r17
mov r16,r12
rcall lcdwr

//ldi r16,$2e
//rcall lcdwr

add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$30
rcall lcdwr

ldi r16,$41
rcall lcdwr

rcall delay


rjmp line_Curr

//voltage  voltage  voltage  voltage  voltage  voltage  voltage  voltage
voltage
//voltage  voltage  voltage  voltage  voltage  voltage  voltage  voltage
voltage
//voltage  voltage  voltage  voltage  voltage  voltage  voltage  voltage
voltage

line_vltge1:
clr r3
clr r4
ldi r17,$09

batv1:
ldi r16,$60
out admux,r16

ldi r16,$c0
out adcsra,r16

//rcall delay1


battt1:
```

```
ldi r16,$10
in r18,adcsra
andi r18,$10
cp r18,r16
brne battt1

in r16,adch
add r3,r16
brcc skip42
inc r4
skip42:
dec r17
brne batv1

mov r25,r4
mov r24,r3

clr r16

yy1:
inc r16
sbiw r25:r24,$09
brcc yy1

dec r16

mov r21,r16
rcall Write_data

mov r16,r21

clr r21




rcall bintobcd1

ldi r16,$c5
rcall cmd

ldi r17,$30

add r11,r17
mov r16,r11
rcall lcdwr

add r12,r17
mov r16,r12
rcall lcdwr

add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$56
rcall lcdwr
```

```
ret


//current current  current  current  current  current  current  current  current
//current  current  current  current  current  current  current  current
current
//current  current  current  current  current  current  current  current
current
//current current  current  current  current  current  current  current  current
//current  current  current  current  current  current  current  current
current
//current  current  current  current  current  current  current  current
current

line_Curr1:

clr r3
clr r4
ldi r17,$09

batr1:
ldi r16,$61
out admux,r16

ldi r16,$c0
out adcsra,r16

//rcall delay1

baatt1:
ldi r16,$10
in r18,adcsra
andi r18,$10
cp r18,r16
brne baatt1

in r16,adch
add r3,r16
brcc skipa2

inc r4

skipa2:
dec r17
brne batr1

mov r25,r4
mov r24,r3

clr r16

yya:
inc r16
sbiw r25:r24,$09
brcc yya
```

```
dec r16

mov r21,r16

rcall Write_data


rcall Stop_wr_en


mov r16,r21

clr r21

mov r19,r16


erty1:
ldi r18,$01
ldi r17,$10
rcall rtyuu

mov r17,r16

cpi r17,$00

breq ertert1
rjmp ertert

ertert1:

ldi r18,$01
ldi r17,$1a
rcall rtyuu

mov r17,r16

mul r19,r17

mov r16,r0

rcall bintobcd1

ldi r16,$ca
rcall cmd


ldi r17,$30

add r11,r17

mov r16,r11

cpi r16,$30
breq skjss

mov r16,r11
```

```
rcall lcdwr

skjss:
add r12,r17
mov r16,r12
rcall lcdwr

//ldi r16,$2e
//rcall lcdwr

add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$41
rcall lcdwr

ret


ertert:
mul r19,r17

mov r16,r0

rcall bintobcd1

ldi r16,$ca
rcall cmd


ldi r17,$30

add r11,r17

mov r16,r11

cpi r16,$30
breq lsjkdd

mov r16,r11
rcall lcdwr

lsjkdd:
add r12,r17
mov r16,r12
rcall lcdwr

//ldi r16,$2e
//rcall lcdwr

add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$30
rcall lcdwr
```

```
ldi r16,$30
rcall lcdwr

ldi r16,$41
rcall lcdwr

ret
//Auto_relay_switch    Auto_relay_switch    Auto_relay_switch
Auto_relay_switch
//Auto_relay_switch     Auto_relay_switch    Auto_relay_switch
Auto_relay_switch
//Auto_relay_switch    Auto_relay_switch    Auto_relay_switch
Auto_relay_switch
//Auto_relay_switch    Auto_relay_switch    Auto_relay_switch
Auto_relay_switch


Auto_relay_switch:

Cck:


           nop
           nop

               ldi r16,$20
           out TIMSK,r16

               nop
               nop
           nop
               rcall delay_ww
           nop
           nop
           nop

               clr r16
           out TIMSK,r16

               ldi r16,$01
               rcall cmd
               rcall delay1


             mov r16,r14
             cpi r16,$14
             brcs exit123

        // Reverse CT Connection



               nop
           nop

               ldi r16,$20
           out TIMSK,r16
           nop
               nop
           nop
               rcall delay_ww
           nop
           nop
           nop

               clr r16
           out TIMSK,r16




               mov r16,r14
             cpi r16,$14
             brcc exit22

                          rjmp exit1

              // Reverse CT Connection

exit22:    ldi r16,$01
             rcall cmd
             rcall delay1
             ldi r16,$52
             rcall lcdwr
             ldi r16,$65
             rcall lcdwr
             ldi r16,$76
             rcall lcdwr
             ldi r16,$65
             rcall lcdwr
             ldi r16,$72
             rcall lcdwr
           ldi r16,$73
             rcall lcdwr
             ldi r16,$65
             rcall lcdwr
             ldi r16,$20
             rcall lcdwr
             ldi r16,$43
             rcall lcdwr
             ldi r16,$54
             rcall lcdwr
             ldi r16,$20
             rcall lcdwr
             ldi r16,$43
             rcall lcdwr
             ldi r16,$6f
             rcall lcdwr
             ldi r16,$6e
             rcall lcdwr
             ldi r16,$6e
             rcall lcdwr
               rcall delay
           rjmp Cck


exit123:

; set 1 for LEAD and 0 for LAG

up1:rcall Measure_pf


rcall check_pf_limits
cpi r16,$ab
brne nexxt

rjmp up1

nexxt:
cpi r16,$ef
brne nexxt1
cpi r26,$00
brne lkertkkr
inc r26
rjmp label0093
lkertkkr:
dec r26
rjmp label0093

nexxt1:
cpi r16,$cd
brne up1

;*******************************
label0093:
cpi r26,$00
breq label0030

cpi r26,$01
breq label0031

rjmp up1




label0031:
;************Main Decrement Loop**********************
cpi r28,$00
brne decrer
cpi r20,$00
brne decrer
rjmp up1

decrer:
rcall Decr_Capacitor
            rcall switch_delay_dec
```

```
   mov Prev_pf,Pres_pf


              cpi Pres_pf,$5a

          brcs label0042

              rcall Measure_pf
             //   rcall delay
            //    rcall delay
          cp Pres_pf,Prev_pf

              brcc up1



label0042:rcall Measure_pf
           //   rcall delay
            //    rcall delay
          mov r16,Pres_pf
          mov r17,Prev_pf
              cp Pres_pf,Prev_pf


              brcs label0016
          sub r16,r17

              cpi r16,$01
          brcc up1

              rcall Decr_Capacitor

                  rcall switch_delay_inc
              rjmp label0042

label0016:

          sub r17,r16

              cpi r17,$01



              brcc up1                      ;Major change in  switching method





              rcall Decr_Capacitor
            rcall switch_delay_dec
```

```
                rjmp  label0042


/*

;*********Increment Subloop***************************

label0036:

rcall Incr_Capacitor
rcall delay
rcall delay
rcall delay
up3:rcall Measure_pf

ldi r16,$5a
rcall lcdwr        ;Z

mov r16,Pres_pf

mov r17,Prev_pf

cp r16,r17

brcs label0012

sub r16,r17

cpi r16,$03

brcc up1

rjmp up3

label0012:sub r17,r16

cpi r17,$03

brcc up1
rjmp up3

*/
;************Main Increment Loop**********************
label0030:
cpi r28,$ff
brne Increr
cpi r20,$f0
brne Increr
rjmp up1

Increr:
rcall curr_min
nop
nop
```

```
cpi r16,$05
brsh dfgertghj
rjmp up1

dfgertghj:  rcall Incr_Capacitor

            rcall switch_delay_inc


              mov Prev_pf,Pres_pf


              cpi Pres_pf,$5a

          brcs label0041
              rcall Measure_pf
            // rcall delay
                  //rcall delay
          cp Pres_pf,Prev_pf

              brcc label0049
              rjmp label0035

label0041:rcall Measure_pf
            //rcall delay
                  //rcall delay
          mov r16,Pres_pf
          mov r17,Prev_pf
              cp Pres_pf,Prev_pf

              brcs label0014
          sub r16,r17

              cpi r16,$03
              brcc label0048

              rcall Incr_Capacitor

            rcall switch_delay_inc
              rjmp label0041

label0014:sub r17,r16

              cpi r17,$03


              brcc label0035
              rcall Incr_Capacitor

                  rcall switch_delay_inc
              rjmp label0041


label0049:rjmp up1


label0035:
```

```
;*********Decrement Subloop***************************

rcall Decr_Capacitor

      rcall switch_delay_dec

up2:rcall Measure_pf
      //rcall delay
            //    rcall delay


mov r16,Pres_pf

mov r17,Prev_pf

cp r16,r17

brcs label0011

sub r16,r17

cpi r16,$03

brcc label0048

rjmp up2

label0011:

sub r17,r16

cpi r17,$03

brcc label0048
rjmp up2


label0048:rjmp up1

;********************************************************

label0030a:sei
           clr Cap_switch
             out porta,Cap_switch
label0031a:rcall Measure_pf
//    rcall delay
//                rcall delay
           rjmp label0031a

/*
menu1:ldi r16,$4c
      rcall lcdwr     ;L
      ldi r16,$69
      rcall lcdwr     ;i
        ldi r16,$6e
      rcall lcdwr     ;n
        ldi r16,$65
      rcall lcdwr     ;e
        ldi r16,$20
      rcall lcdwr
        ldi r16,$56
      rcall lcdwr     ;V
        ldi r16,$6f
      rcall lcdwr     ;o
        ldi r16,$6c
      rcall lcdwr     ;l
        ldi r16,$74
      rcall lcdwr     ;t
        ldi r16,$61
      rcall lcdwr     ;a
        ldi r16,$67
      rcall lcdwr     ;g
        ldi r16,$65
      rcall lcdwr     ;e
*/

        /*

        ldi r16,$73
      rcall lcdwr     ;s
        ldi r16,$28
      rcall lcdwr
        ldi r16,$52
      rcall lcdwr
        ldi r16,$4d
      rcall lcdwr
        ldi r16,$53
      rcall lcdwr
        ldi r16,$29
      rcall lcdwr
        */
        ret

/*
menu2:ldi r16,$4c
      rcall lcdwr     ;L
      ldi r16,$69
      rcall lcdwr     ;i
        ldi r16,$6e
      rcall lcdwr     ;n
        ldi r16,$65
      rcall lcdwr     ;e
        ldi r16,$20
      rcall lcdwr
      ldi r16,$43
      rcall lcdwr      ;C
        ldi r16,$75
      rcall lcdwr      ;u
        ldi r16,$72
      rcall lcdwr      ;r
        ldi r16,$72
      rcall lcdwr      ;r
        ldi r16,$65
      rcall lcdwr      ;e
```

```
        ldi r16,$6e
      rcall lcdwr      ;n
        ldi r16,$74
      rcall lcdwr      ;t
*/

        /*
        ldi r16,$73
        rcall lcdwr
        ldi r16,$28
      rcall lcdwr
        ldi r16,$52
      rcall lcdwr
        ldi r16,$4d
      rcall lcdwr
        ldi r16,$53
      rcall lcdwr
        ldi r16,$29
      rcall lcdwr
        */

        ret

menu3:ldi r16,$54      ;T
      rcall lcdwr
        ldi r16,$6f
      rcall lcdwr      ;o
        ldi r16,$74
      rcall lcdwr      ;t
        ldi r16,$61
      rcall lcdwr      ;a
        ldi r16,$6c
      rcall lcdwr      ;l
        ldi r16,$20
      rcall lcdwr
        ldi r16,$6b
      rcall lcdwr      ;k
        ldi r16,$57
      rcall lcdwr      ;W
        ldi r16,$2c
      rcall lcdwr
        ldi r16,$6b
      rcall lcdwr      ;k
        ldi r16,$56
      rcall lcdwr      ;V
        ldi r16,$41
      rcall lcdwr      ;A

       /*

        ldi r16,$2c
      rcall lcdwr
        ldi r16,$6b
      rcall lcdwr      ;k
        ldi r16,$56
      rcall lcdwr      ;V
        ldi r16,$41
      rcall lcdwr      ;A
        */

        ret

menu4:
      ldi r16,$53
        rcall lcdwr
        ldi r16,$77
        rcall lcdwr
        ldi r16,$69
        rcall lcdwr
        ldi r16,$74
        rcall lcdwr
        ldi r16,$63
        rcall lcdwr
        ldi r16,$68
        rcall lcdwr
        ldi r16,$69
        rcall lcdwr
        ldi r16,$6e
        rcall lcdwr
        ldi r16,$67
        rcall lcdwr
        ldi r16,$20
        rcall lcdwr
        ldi r16,$44
        rcall lcdwr
        ldi r16,$65
        rcall lcdwr
        ldi r16,$6c
        rcall lcdwr
        ldi r16,$61
        rcall lcdwr
        ldi r16,$79
        rcall lcdwr
        ret


/*
      ldi r16,$43      ;C
      rcall lcdwr
        ldi r16,$2d
      rcall lcdwr
        ldi r16,$6b
      rcall lcdwr      ;k
        ldi r16,$56
      rcall lcdwr      ;V
        ldi r16,$41
      rcall lcdwr      ;A
        ldi r16,$72
      rcall lcdwr      ;r
        ret
*/
```

```
/*
menu5:
      ldi r16,$4d
      rcall lcdwr       ;M
        ldi r16,$61
      rcall lcdwr       ;a
        ldi r16,$69
      rcall lcdwr       ;i
        ldi r16,$6e
      rcall lcdwr       ;n
        ldi r16,$73
      rcall lcdwr       ;s
        ldi r16,$20
      rcall lcdwr
        ldi r16,$46
      rcall lcdwr       ;F
        ldi r16,$72
      rcall lcdwr       ;r
        ldi r16,$65
      rcall lcdwr       ;e
        ldi r16,$71
      rcall lcdwr       ;q
        ldi r16,$75
      rcall lcdwr       ;u
        ldi r16,$65
      rcall lcdwr       ;e
        ldi r16,$6e
      rcall lcdwr       ;n
        ldi r16,$63
      rcall lcdwr       ;c
        ldi r16,$79
      rcall lcdwr       ;y
        ret

*/


menu6:ldi r16,$43
      rcall lcdwr      ;k
        ldi r16,$54
      rcall lcdwr      ;W
      ldi r16,$20
      rcall lcdwr      ;h
        ldi r16,$52
      rcall lcdwr
        ldi r16,$61
      rcall lcdwr      ;k
        ldi r16,$74
      rcall lcdwr      ;V
        ldi r16,$69
      rcall lcdwr      ;A
        ldi r16,$6f
      rcall lcdwr      ;h
        ret

/*
```

```
menu7:
      ldi r16,$4c
      rcall lcdwr       ;L
      ldi r16,$6f
      rcall lcdwr       ;o
        ldi r16,$61
      rcall lcdwr       ;a
      ldi r16,$64
      rcall lcdwr       ;d
        ldi r16,$20
      rcall lcdwr
        ldi r16,$53
      rcall lcdwr       ;S
        ldi r16,$69
      rcall lcdwr       ;i
        ldi r16,$64
      rcall lcdwr       ;d
        ldi r16,$65
      rcall lcdwr       ;e
        ldi r16,$20
      rcall lcdwr
        ldi r16,$50
      rcall lcdwr       ;P
        ldi r16,$46
      rcall lcdwr       ;F
        ret

*/


/*


menu8:
      ldi r16,$4c
      rcall lcdwr       ;L
      ldi r16,$6f
      rcall lcdwr       ;o
        ldi r16,$61
      rcall lcdwr       ;a
      ldi r16,$64
      rcall lcdwr       ;d
        ldi r16,$20
      rcall lcdwr
        ldi r16,$53
      rcall lcdwr       ;S
        ldi r16,$69
      rcall lcdwr       ;i
        ldi r16,$64
      rcall lcdwr       ;d
        ldi r16,$65
      rcall lcdwr       ;e
        ldi r16,$20
      rcall lcdwr
        ldi r16,$6b
      rcall lcdwr      ;k
```

```
        ldi r16,$56
      rcall lcdwr      ;V
        ldi r16,$41
      rcall lcdwr      ;A
        ldi r16,$72
      rcall lcdwr      ;r
        ldi r16,$20
        rcall lcdwr
      ldi r16,$6b
      rcall lcdwr      ;k
        ldi r16,$56
      rcall lcdwr      ;V
        ldi r16,$41
      rcall lcdwr      ;A
        ret

*/


menu9: ldi r16,$4d
         rcall lcdwr; m
       ldi r16,$61
         rcall lcdwr;   a
         ldi r16,$6e
         rcall lcdwr;   n
         ldi r16,$75
         rcall lcdwr;   u
         ldi r16,$61
         rcall lcdwr;   a
         ldi r16,$6c
         rcall lcdwr;   l
         ldi r16,$20
         rcall lcdwr;
         ldi r16,$53
         rcall lcdwr;   s
         ldi r16,$77
         rcall lcdwr;   w
         ldi r16,$69
         rcall lcdwr;   i
         ldi r16,$74
         rcall lcdwr;   t
         ldi r16,$63
         rcall lcdwr;   c
         ldi r16,$68
         rcall lcdwr;   h
         ret


/*

      ldi r16,$43
      rcall lcdwr       ;C
      ldi r16,$61
      rcall lcdwr       ;a
        ldi r16,$70
      rcall lcdwr       ;p
      ldi r16,$61
      rcall lcdwr       ;a
        ldi r16,$63
      rcall lcdwr       ;c
        ldi r16,$69
      rcall lcdwr       ;i
        ldi r16,$74
      rcall lcdwr       ;t
        ldi r16,$6f
      rcall lcdwr       ;o
        ldi r16,$72
      rcall lcdwr       ;r
        ldi r16,$20
      rcall lcdwr
        ldi r16,$43
      rcall lcdwr      ;C
        ldi r16,$68
      rcall lcdwr      ;h
        ldi r16,$61
      rcall lcdwr      ;a
        ldi r16,$6e
      rcall lcdwr      ;n
        ldi r16,$6e
        rcall lcdwr       ;n
      ldi r16,$65
      rcall lcdwr      ;e
        ldi r16,$6c
      rcall lcdwr      ;l
        ldi r16,$73
      rcall lcdwr      ;s
        ret

*/


/*

menu10:
      ldi r16,$43
       rcall lcdwr       ;C
       ldi r16,$61
       rcall lcdwr       ;a
         ldi r16,$70
       rcall lcdwr       ;p
       ldi r16,$61
       rcall lcdwr       ;a
         ldi r16,$63
       rcall lcdwr       ;c
         ldi r16,$69
       rcall lcdwr       ;i
         ldi r16,$74
       rcall lcdwr       ;t
         ldi r16,$6f
       rcall lcdwr       ;o
         ldi r16,$72
       rcall lcdwr       ;r
         ldi r16,$20
       rcall lcdwr
```

```
          ldi r16,$46
        rcall lcdwr      ;F
          ldi r16,$61
        rcall lcdwr      ;a
          ldi r16,$69
        rcall lcdwr      ;i
          ldi r16,$6c
        rcall lcdwr      ;l
          ret

*/



menu11:
        ldi r16,$50
        rcall lcdwr       ;P
        ldi r16,$46
        rcall lcdwr       ;F
          ldi r16,$20
        rcall lcdwr
        ldi r16,$4c
        rcall lcdwr       ;L
          ldi r16,$69
        rcall lcdwr       ;i
          ldi r16,$6d
        rcall lcdwr       ;m
          ldi r16,$69
        rcall lcdwr       ;i
          ldi r16,$74
        rcall lcdwr       ;t
          ldi r16,$73
        rcall lcdwr       ;s
          ret



menu12:
        ldi r16,$53    ;switch method
          rcall lcdwr
          ldi r16,$77
          rcall lcdwr
          ldi r16,$69
          rcall lcdwr
          ldi r16,$74
          rcall lcdwr
          ldi r16,$63
          rcall lcdwr
          ldi r16,$68
          rcall lcdwr
        ldi r16,$20
          rcall lcdwr
          ldi r16,$4d
          rcall lcdwr
          ldi r16,$65
          rcall lcdwr
          ldi r16,$74
```

```
         rcall lcdwr
         ldi r16,$68
         rcall lcdwr
         ldi r16,$6f
         rcall lcdwr
         ldi r16,$64
         rcall lcdwr
         ret



/*



*/



menu13:
ldi r16,$53
       rcall lcdwr       ;S
       ldi r16,$65
       rcall lcdwr       ;e
         ldi r16,$72
       rcall lcdwr       ;r
       ldi r16,$69
       rcall lcdwr       ;i
         ldi r16,$61
       rcall lcdwr       ;a
         ldi r16,$6c
       rcall lcdwr       ;l
         ldi r16,$20
       rcall lcdwr
         ldi r16,$43
       rcall lcdwr       ;C
         ldi r16,$6f
       rcall lcdwr       ;o
         ldi r16,$6d
       rcall lcdwr       ;m
         ldi r16,$6d
       rcall lcdwr       ;m
         ret





/*
      ldi r16,$43
      rcall lcdwr
      ldi r16,$68
      rcall lcdwr
        ldi r16,$61
      rcall lcdwr
        ldi r16,$6e
      rcall lcdwr
        ldi r16,$67
      rcall lcdwr
        ldi r16,$65
      rcall lcdwr
        ldi r16,$20
      rcall lcdwr
        ldi r16,$4b
           rcall lcdwr
               ldi r16,$45
           rcall lcdwr
               ldi r16,$59
           rcall lcdwr
        ret

*/









/*
menu13:
       ldi r16,$43
       rcall lcdwr       ;S
       ldi r16,$68
       rcall lcdwr       ;e
         ldi r16,$61
       rcall lcdwr       ;t
       ldi r16,$6e
       rcall lcdwr
         ldi r16,$67
       rcall lcdwr       ;D
         ldi r16,$65
       rcall lcdwr       ;a
         ldi r16,$20
       rcall lcdwr       ;t
         ldi r16,$50
       rcall lcdwr       ;e
         ldi r16,$61
       rcall lcdwr
         ldi r16,$73
       rcall lcdwr
         ldi r16,$73
       rcall lcdwr
         ldi r16,$77
       rcall lcdwr
         ldi r16,$6f
       rcall lcdwr
         ldi r16,$72
       rcall lcdwr
         ldi r16,$64
       rcall lcdwr
         ret
```

```
*/




menu14:
ldi r16,$53
rcall lcdwr
ldi r16,$74
rcall lcdwr
ldi r16,$61
rcall lcdwr
ldi r16,$67
rcall lcdwr
ldi r16,$65
rcall lcdwr
ldi r16,$20
rcall lcdwr
ldi r16,$53
rcall lcdwr
ldi r16,$65
rcall lcdwr
ldi r16,$6c
rcall lcdwr
ldi r16,$65
rcall lcdwr
ldi r16,$63
rcall lcdwr
ldi r16,$74
rcall lcdwr
ret


menu15:ldi r16,$45
       rcall lcdwr       ;E
       ldi r16,$78
       rcall lcdwr       ;x
         ldi r16,$69
       rcall lcdwr       ;i
       ldi r16,$74
       rcall lcdwr       ;t
         ret

Screen1:ldi r16,$01
          rcall cmd
          rcall delay1
          //rcall delay1
            ldi r16,$7e
          rcall lcdwr
        rcall menu3
          ldi r16,$c1
        rcall cmd
          rcall menu4
         // ldi r16,$95
         // rcall cmd
         // rcall menu3
         // ldi r16,$d5
         // rcall cmd
         // rcall menu4
          ldi r27,$01
        ret

Screen2:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu4
        ldi r16,$c1
        rcall cmd
          rcall menu6
          //ldi r16,$95
          //rcall cmd
        //rcall menu4
        //ldi r16,$d5
          //rcall cmd
        //rcall menu6
          ldi r27,$02
          ret

Screen3:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu6
          ldi r16,$c1
        rcall cmd
        rcall menu9
          //ldi r16,$95
          //rcall cmd
        //rcall menu6
          //ldi r16,$d5
          //rcall cmd
        //rcall menu9
          ldi r27,$03
          ret

Screen4:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu9
        ldi r16,$c1
        rcall cmd
          rcall menu11
          //ldi r16,$95
          //rcall cmd
        //rcall menu9
          //ldi r16,$d5
          //rcall cmd
        //rcall menu11
          ldi r27,$04
        ret

Screen5:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu11
        ldi r16,$c1
        rcall cmd
          rcall menu12
          //ldi r16,$95
          //rcall cmd
        //rcall menu11
          //ldi r16,$d5
          //rcall cmd
        //rcall menu12
          ldi r27,$05
          ret

Screen6:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu12
        ldi r16,$c1
        rcall cmd
          rcall menu13
          //ldi r16,$95
          //rcall cmd
        //rcall menu12
          //ldi r16,$d5
          //rcall cmd
        //rcall menu13
          ldi r27,$06
          ret

Screen7:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu13
        ldi r16,$c1
        rcall cmd
          rcall menu14
          //ldi r16,$95
          //rcall cmd
        //rcall menu13
          //ldi r16,$d5
          //rcall cmd
        //rcall menu14
        ldi r27,$07
          ret
```

```
Screen8:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu14
        ldi r16,$c1
        rcall cmd
          rcall menu15
          //ldi r16,$95
          //rcall cmd
        //rcall menu14
          //ldi r16,$d5
          //rcall cmd
        //rcall menu15
          ldi r27,$08
          ret

Screen9:ldi r16,$01
          rcall cmd
          rcall delay1
          ldi r16,$7e
          rcall lcdwr
        rcall menu15
        ldi r16,$c1
        rcall cmd
          rcall menu3
          //ldi r16,$95
          //rcall cmd
        //rcall menu15
          //ldi r16,$d5
          //rcall cmd
        //rcall menu1
          ldi r27,$09
          ret

/*
Screen10:ldi r16,$01
           rcall cmd
           rcall delay1
           ldi r16,$7e
           rcall lcdwr
         rcall menu14
         ldi r16,$c1
         rcall cmd
           rcall menu15
           //ldi r16,$95
           //rcall cmd
         //rcall menu1
           //ldi r16,$d5
           //rcall cmd
         //rcall menu2
           ldi r27,$10
           ret
```

```
Screen11:ldi r16,$01
           rcall cmd
           rcall delay1
           ldi r16,$7e
           rcall lcdwr
         rcall menu15
         ldi r16,$c1
         rcall cmd
           rcall menu1
           //ldi r16,$95
           //rcall cmd
         //rcall menu2
           //ldi r16,$d5
           //rcall cmd
         //rcall menu3
           ldi r27,$11
           ret

*/

/*
Screen12:ldi r16,$01
           rcall cmd
           rcall delay1
           ldi r16,$7e
           rcall lcdwr
         rcall menu12
         ldi r16,$c1
         rcall cmd
           rcall menu13
           ldi r16,$95
           rcall cmd
         rcall menu14
           ldi r16,$d5
           rcall cmd
         rcall menu15
           ldi r27,$12
           ret

Screen13:ldi r16,$01
           rcall cmd
           rcall delay1
           ldi r16,$7e
           rcall lcdwr
         rcall menu13
         ldi r16,$c1
         rcall cmd
           rcall menu14
           ldi r16,$95
           rcall cmd
         rcall menu15
           ldi r16,$d5
           rcall cmd
         rcall menu1
           ldi r27,$13
           ret
```

```
Screen14:ldi r16,$01
           rcall cmd
           rcall delay1
           ldi r16,$7e
           rcall lcdwr
         rcall menu14
         ldi r16,$c1
         rcall cmd
           rcall menu15
           ldi r16,$95
           rcall cmd
         rcall menu1
           ldi r16,$d5
           rcall cmd
         rcall menu2
           ldi r27,$14
           ret

Screen15:ldi r16,$01
           rcall cmd
           rcall delay1
           ldi r16,$7e
           rcall lcdwr
         rcall menu15
         ldi r16,$c1
         rcall cmd
           rcall menu1
           ldi r16,$95
           rcall cmd
         rcall menu2
           ldi r16,$d5
           rcall cmd
         rcall menu3
           ldi r27,$15
           ret


*/

one_zero:
           cpi r19,$30
           brne kjsdhfkjsh
           mov r16,r21
               rcall cmd
               ldi r19,$31
               mov r16,r19
               rcall lcdwr
               mov r16,r21
               rcall cmd
           rjmp keywait

kjsdhfkjsh:mov r16,r21
               rcall cmd
           ldi r19,$30
               mov r16,r19
               rcall lcdwr
               mov r16,r21
               rcall cmd
               rjmp keywait




hjgsdfyh:
             cpi r19,$c0
             brne ksjd456fhe

              mov r16,r19
              rcall cmd

              ldi r16,$20
              rcall lcdwr

              ldi r19,$c8
              mov r16,r19
              rcall cmd

          ldi r16,$7e
              rcall lcdwr

              rjmp keywait



ksjd456fhe: mov r16,r19
             rcall cmd

              ldi r16,$20
              rcall lcdwr

              ldi r19,$c0
              mov r16,r19
              rcall cmd

          ldi r16,$7e
              rcall lcdwr

              rjmp keywait




;******************************************************************

;***************** End of the subroutine section ******************
;
;************************ Main program ****************************
;
; Main program routine starts here
;
Start:cli

      ldi r16,$40
      out GICR,r16
```

```
        ldi r16,$03
        out MCUCR,r16

      ldi   r16,low(RAMEND)         ; Load low byte address of end of RAM into
register R16
        out SPL,r16                 ; Initialize stack pointer to end of
internal RAM
        ldi       R16,high(RAMEND)  ; Load high byte address of end of RAM into
register R16
      out   SPH,r16               ; Initialize high byte of stack pointer to end
of internal RAM

     ldi r16,(1<<PUD)
       out SFIOR,r16
     nop
       nop



       ldi r16,$ff
     out ddrb,r16



     ldi r16,(1<<DDD1)|(1<<DDD0)|(1<<DDD4)|(1<<DDD5)|(1<<DDD7)
     out ddrd,r16


    clr r16
     out portd,r16

     clr r16

     rcall latch1_send1

     clr r16

       rcall latch2_send2

        clr r26
        clr r27
        clr r28
        clr r29

      rcall init_lcd



ldi r16,$0c
out UBRRL,r16

ldi r16,(1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1)
out UCSRC,r16

//ldi r16,(1<<RXEN)|(1<<TXEN)
//out UCSRB,r16
```

```
ldi r16,(1<<U2X)
out UCSRA,r16

         clr r16
         out TCNT1H,r16
         out TCNT1L,r16


         clr Capt_Count
       clr Prev_pf
         clr Pres_pf
         clr Cap_switch

       ldi r16,$40
       out TCCR1B,r16

       ldi r16,$64
         out TWBR,r16

         ldi r16,$00
         out TWSR,r16

       sei


//ldi r16,$42
//rcall lcdwr
// rcall delay



//rr: rjmp rr


;********************************************************************************

       rjmp Auto_relay_switch


;****************************************************************

Cap_Int:cli
        cpi Capt_Count,$00
        breq rising
            rjmp falling

rising:ldi r16,$01
         out TCCR1B,r16

       ldi Capt_Count,$01
       reti


falling:in r16,ICR1L
       in r17,ICR1H

         mov r13,r16
         mov r14,r17

         ldi r16,$40
         out TCCR1B,r16
       clr r16
         out TCNT1H,r16
       out TCNT1L,r16
         clr Capt_Count
       reti

;*******************************************************************************

;************************ KEYPAD INTERRUPT ROUTINE ****************************

Ext_Int_0:cli

              clr r16
          out TIMSK,r16

          rcall key_debounce



               in r16,pinc
           andi r16,$f0


               ldi r17,$70
               cpse r16,r17
               rjmp key_two

;key_one key_one key_one key_one key_one key_one key_one key_one key_one
;key_one key_one key_one key_one key_one key_one key_one key_one key_one
;key_one key_one key_one key_one key_one key_one key_one key_one key_one

//Decrement

         cpi r27,$00
         brne check1
//Screen1
       clr r16
       out TIMSK,r16

       rjmp paestrghttt

         rjmp keywait

check1:cpi r27,$01
       brne check2
//Screen2
       rcall Screen2

         rjmp keywait

check2:cpi r27,$02
       brne check3
//Screen3
```

```
         rcall Screen3

         rjmp keywait

check3:cpi r27,$03
       brne check4
//Screen4
         rcall Screen4

        rjmp keywait

check4:cpi r27,$04
       brne check5
//Screen5
         rcall Screen5

rjmp keywait

check5:cpi r27,$05
       brne check6
//Screen6
         rcall Screen6

       rjmp keywait

check6:cpi r27,$06
       brne check7
//Screen7
         rcall Screen7

rjmp keywait

check7:cpi r27,$07
       brne check8
//Screen8
         rcall Screen8

rjmp keywait

check8:cpi r27,$09
       brne check234161
//Screen9
         rcall Screen1

         rjmp keywait

check234161:cpi r27,$08

            brne check23416123
//Screen9
         rcall Screen9

         rjmp keywait


check23416123:
/*
```

```
cpi r27,$10
        brne check161
//Screen11
          rcall Screen11

rjmp keywait


check161:cpi r27,$11
        brne check121
//Screen12
          rcall Screen1

        rjmp keywait

*/


/*

rjmp keywait

check121:cpi r27,$12
        brne check131
//Screen13
//        rcall Screen13

rjmp keywait

check131:cpi r27,$13
        brne check141
//Screen14
//        rcall Screen14

          rjmp keywait

check141:cpi r27,$14
        brne check151
//Screen15
//        rcall Screen15

rjmp keywait



check151:cpi r27,$15
         brne check161
//Screen1
          rcall Screen1

rjmp keywait

*/


check121:
         cpi r27,$16
```

```
             brne check181

           sei

             rcall Measure_pf
             cli

           rcall Decr_capacitor

eywait:      sei
         rcall Measure_pf
           ldi r16,$86
             rcall cmd
             ldi r16,$4d
             rcall lcdwr
             ldi r16,$41
             rcall lcdwr
             ldi r16,$4e
             rcall lcdwr
             //rcall delay
       rjmp eywait

check181:cpi r27,$c0
         brne check191
         ldi r16,$c0
             rcall cmd
             ldi r16,$20
         rcall lcdwr
             ldi r16,$cc
             rcall cmd
             ldi r16,$7e
             rcall lcdwr
             ldi r27,$cc
             rjmp keywait

check191:cpi r27,$cc
         brne check1191
         ldi r16,$cc
             rcall cmd
             ldi r16,$20
         rcall lcdwr
             ldi r16,$c6
             rcall cmd
             ldi r16,$7e
             rcall lcdwr
             ldi r27,$c6
             rjmp keywait

check1191:cpi r27,$c6
         brne check11911
         ldi r16,$c6
             rcall cmd
             ldi r16,$20
         rcall lcdwr
             ldi r16,$c0
             rcall cmd
             ldi r16,$7e
             rcall lcdwr
             ldi r27,$c0
             rjmp keywait

check11911:
mov r16,r27
andi r16,$f0
cpi r16,$30

brne check11920

 rjmp dicval

check11920:mov r16,r27
andi r16,$f0
cpi r16,$60
brne check11932
rjmp dicval

check11932:cpi r27,$81

             brne check11936

rjmp hjgsdfyh

check11936:

mov r16,r27

andi r16,$f0

cpi r16,$50

brne latch22

clr r16

rcall latch2_send2

rjmp measure12

latch22:

cpi r16,$70

brne check11937

clr r16

rcall latch1_send1

rjmp measure12
```

```
check11937:
mov r16,r27
andi r16,$f0
cpi r16,$20
brne check1ff43

  rjmp dicval



check1ff43:
mov r16,r27
andi r16,$f0
cpi r16,$40

brne check1ff44
rjmp one_zero


check1ff44:
mov r16,r27
andi r16,$f0
cpi r16,$90
brne check1ff45
rjmp one_zero

check1ff45:
cpi r27,$a1
brne check1sf45
rjmp one_zero

check1sf45:
mov r16,r27
andi r16,$f0
cpi r16,$a0
brne check1ff46
rjmp dicval

check1ff46:
cpi r27,$b1
brne check1sf46
rjmp one_zero



check1sf46:

reti

//key_two key_two key_two key_two key_two key_two key_two key_two key_two
key_two
//key_two key_two key_two key_two key_two key_two key_two key_two key_two
key_two
//key_two key_two key_two key_two key_two key_two key_two key_two key_two
key_two

key_two:  ldi r17,$e0
```

```
          cpse r16,r17
              rjmp key_three

;key_two

//Decrement

         cpi r27,$00
         brne check11
//Screen1
       rjmp paestrghttt

       rjmp keywait

check11:cpi r27,$01
        brne check91
//Screen8
          rcall Screen9

        rjmp keywait

check91:cpi r27,$09
        brne checkA1
//Screen7
          rcall Screen8

rjmp keywait

checkA1:cpi r27,$07
        brne checkB1
//Screen6
          rcall Screen6

         rjmp keywait

checkB1:cpi r27,$06
        brne checkC1
//Screen5
          rcall Screen5

rjmp keywait

checkC1:cpi r27,$05
        brne checkD1
//Screen4
          rcall Screen4

rjmp keywait

checkD1: cpi r27,$04
         brne checkE1
//Screen3
           rcall Screen3

rjmp keywait

checkE1: cpi r27,$03
```

```
         brne ch234eckF1
//Screen2
           rcall Screen2

rjmp keywait


ch234eckF1:cpi r27,$08
         brne ch1
//Screen2
           rcall Screen7

rjmp keywait


ch1:
/*
cpi r27,$10
         brne ch2
//Screen2
           rcall Screen9

rjmp keywait

ch2:cpi r27,$11
         brne checkF1
//Screen2
           rcall Screen10

rjmp keywait

*/

checkF1: cpi r27,$02
         brne checkG1
//Screen1
           rcall Screen1

           rjmp keywait



checkG1: cpi r27,$16
         brne checkI1

          sei
             rcall Measure_pf
             rcall Incr_capacitor

eywait1: sei
         rcall Measure_pf
             ldi r16,$86
             rcall cmd
             ldi r16,$4d
             rcall lcdwr
             ldi r16,$41
             rcall lcdwr
             ldi r16,$4e
             rcall lcdwr
             //rcall delay
             rjmp eywait1


checkI1: cpi r27,$c0
         brne checkJ1
         ldi r16,$c0
             rcall cmd
             ldi r16,$20
         rcall lcdwr
             ldi r16,$c6
             rcall cmd
             ldi r16,$7e
             rcall lcdwr
             ldi r27,$c6
             rjmp keywait


checkJ1: cpi r27,$c6
         brne checkK1
         ldi r16,$c6
             rcall cmd
             ldi r16,$20
         rcall lcdwr
             ldi r16,$cc
             rcall cmd
             ldi r16,$7e
             rcall lcdwr
             ldi r27,$cc
             rjmp keywait

checkK1: cpi r27,$cc
         brne checkL1
         ldi r16,$cc
             rcall cmd
             ldi r16,$20
         rcall lcdwr
             ldi r16,$c0
             rcall cmd
             ldi r16,$7e
             rcall lcdwr
             ldi r27,$c0
             rjmp keywait

checkL1:
mov r16,r27
andi r16,$f0
cpi r16,$30
         brne checkM1

      rjmp intcval


checkM1:
```

```
mov r16,r27
andi r16,$f0
cpi r16,$60

         brne checkV1


rjmp intcval


checkV1: cpi r27,$81
         brne checkii1


         rjmp hjgsdfyh


checkii1:
cpi r27,$aa
brne checkkk2

reti


// Manual switch for JAIVIC

checkkk2:cpi r27,$70

brne checkkk3

clr r16
rcall latch2_send2

ldi r16,$80
rcall latch1_send1

rjmp measure12

checkkk3:cpi r27,$71

brne checkkk4

ldi r16,$40
rcall latch1_send1

rjmp measure12


checkkk4:cpi r27,$72

brne checkkk5

ldi r16,$20
rcall latch1_send1

rjmp measure12
```

```
checkkk5:cpi r27,$73

brne checkkk6

ldi r16,$10
rcall latch1_send1

rjmp measure12

checkkk6:cpi r27,$74

brne checkkk7

ldi r16,$08
rcall latch1_send1

rjmp measure12


checkkk7:cpi r27,$75

brne checkkk8

ldi r16,$04
rcall latch1_send1

rjmp measure12



checkkk8:cpi r27,$76

brne checkkk9

ldi r16,$02
rcall latch1_send1

rjmp measure12



checkkk9:cpi r27,$77

brne checkkk10

ldi r16,$01
rcall latch1_send1

rjmp measure12



checkkk10:cpi r27,$50

brne checkkk11

clr r16
```

```
rcall latch1_send1

ldi r16,$80
rcall latch2_send2

rjmp measure12


checkkk11:cpi r27,$51

brne checkkk12

ldi r16,$40
rcall latch2_send2

rjmp measure12


checkkk12:cpi r27,$52

brne checkkk13

ldi r16,$20
rcall latch2_send2

rjmp measure12


checkkk13:cpi r27,$53

brne checkkk14

ldi r16,$10
rcall latch2_send2

rjmp measure12


checkkk14:

mov r16,r27
andi r16,$f0
cpi r16,$20

brne check12123

rjmp intcval

check12123:
mov r16,r27
andi r16,$f0

cpi r16,$40
brne check12124
rjmp one_zero

check12124:
```

```
mov r16,r27
andi r16,$f0
cpi r16,$90
brne check12125
rjmp one_zero

check12125:
cpi r27,$a1
brne check1213425
rjmp one_zero

check1213425:
mov r16,r27
andi r16,$f0
cpi r16,$a0
brne check12126
rjmp intcval

check12126:
cpi r27,$b1

brne check12124
rjmp one_zero

reti

;key_three ;key_three ;key_three ;key_three ;key_three ;key_three ;key_three
;key_three ;key_three ;key_three ;key_three ;key_three ;key_three ;key_three
;key_three ;key_three ;key_three ;key_three ;key_three ;key_three ;key_three


key_three:ldi r17,$d0

               cpse r16,r17

              rjmp key_four


;key_three
;***********************************************
//Enter
;en_screen1
/*
           cpi r27,$01
           brne en_screen2

;en_screen1:
           clr r16
           out TIMSK,r16
           sei
               rjmp line_vltge

en_screen2:cpi r27,$02
           brne en_screen3

;en_screen2:
           clr r16
           out TIMSK,r16
           sei
               rjmp line_curr
*/

en_screen3:cpi r27,$01
           breq en_sscreen3
           rjmp en_screen4

en_sscreen3:

//kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA
kVA kVA kVA kVA
//kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW
kW kW kW kW
//kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA
kVA kVA kVA kVA
//kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW
kW kW kW kW

batcv1:           ldi r16,$60             ;voltage
                  out admux,r16
                  ldi r16,$c0
                  out adcsra,r16;
                  rcall delay1
                  ldi r17,$a;
bttt1:            ldi r16,$10
                  in r18,adcsra
                  andi r18,$10
                  cp r18,r16
                  brne bttt1
                  in r16,adch
                  mov volt,r16                        ;voltage in
                  rcall disply
                  rcall delay;
                  rcall delay;                        ;current
                  ldi r16,$01;
                  rcall cmd;
                  ldi r16,$49;
                  rcall lcdwr;
                  ldi r16,$20;
                  rcall lcdwr;
batww1:
                  ldi r16,$61
                  out admux,r16
                  ldi r16,$c0
                  out adcsra,r16
                  rcall delay1
battw11:    ldi r16,$10
                  in r18,adcsra
                  andi r18,$10
                  cp r18,r16
                  brne batww1;
                  in r16,adch
                  mov curr,r16      ;current in
                  rcall disply
                  rcall delay
                  rcall delay
                  rcall delay

////////////////////////////////////////////////////////////////
//////////pf
/////// KVA KVA KVA KVA       KVA KVA KVA KVA   KVA KVA KVA KVA    KVA KVA KVA
KVA
                  mov r10,r25;
                  mul r24,r25;
                  mov r16,r0;
                  mov r17,r1;
                  mov r24,r16;
                  mov r25,r17;
                  mov r3,r16;
                  mov r4,r17;
                  rcall new_b;

                  ldi r16,$4b;
                  rcall lcdwr;
                  ldi r16,$56;
                  rcall lcdwr;
                  ldi r16,$41;
                  rcall lcdwr;

                  rcall delay;
                  rcall delay

                  clr r18;
                  clr r19;
la1:        inc r18;
                  cpi r18,$FF
                  brcs la2
                  inc r19
//////////////////////////////////////////////////////////////////////
la2:        sbiw r25:r24,$a;divide the the above result by 10
                  cpi r25,$01;
                  brsh la1;
                  cpi r24,$a;
                  brsh la1;
                  //dec r19
                  mov r5,r18;
                  mov r6,r19; store the result in r5&r6
                  mov r24,r18;
                  mov r25,r19;
                  ///////////////////////////
                  clr r18;
                  clr r19;
//////////////////////////////////////////////////////////////////////////

la3:        inc r18;
                  cpi r18,$FF;
                  brcs la4
                  inc r19
la4:        sbiw r25:r24,$2;divde the above result by 2
                  cpi r25,$01;
                  brsh la3;
                  cpi r24,$2;
                  brsh la3;
                  mov r7,r18;no after decimal point of power factor
                  mov r8,r19;lno after decimal point of power factor
///////////////////////////////////////////////////////////////////\\\\\\\\\\\\
////////////////MAIN SUBSTRACTION///////////////////////////////
///////////////// SET PF VALUE ////////////////////////////////
            ///////////////// conversion of pf ////////////////////////////////

d__:        clr r16;
                  mov r16,Pres_pf;
                  mov r15,r16;
                  rcall bintobcd;
                  ldi r17,$10
                  //ldi r17,$10;
                  mul r12,r17;
                  mov r16,r0;
                  add r16,r18;
                  mov r19,r16;
                  ldi r17,$f0;
                  and r17,r16;
                  mov r13,r17;
                  swap r13;
                  mov r16,r13;
                  cpi r16,$01;
                  brcc r_1
end_:       clr r16;
                  rcall lcdwr;
                  clr r16;
                  rcall lcdwr;
                  rjmp gg;

r_1:        ldi r17,$0f;
                  mov r16,r19;
                  and r17,r16;
                  mov r14,r17;
                  clr r17;
                  rjmp r__
///////  multiplication of pf /////
r__:        mov r17,r13;
dd_:        ldi r18,$a;load the 10
                  sub r18,r17;sub pf from 10 to get requird value to minus;

                  clr r16
la5:        inc r16;set loop counter
                  sub r3,r5;
                  sbc r4,r6;
                  cp r16,r18;
                  brcs la5
                  mov r16,r14;second no after decimal point
                  cpi r16,$4;
                  brcs la7;
                  add r3,r7;
                  adc r4,r8;
la7:
                  mov fbinl,r3;
                  mov fbinh,r4
                  rcall new_b;
                  ldi r16,$4b;
                  rcall lcdwr;
                  ldi r16,$57;
                  rcall lcdwr;
end:
                  rcall delay;
                  rcall delay;

gg:               ldi r16,$01
                  rcall cmd
                  rcall delay1
gx:               rjmp en_sscreen3




//kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW
kW kW kW kW
//kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA
kVA kVA kVA kVA
//kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA
kVA kVA kVA kVA
//kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA
kVA kVA kVA kVA
//kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW kW
kW kW kW kW

//kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA kVA
kVA kVA kVA kVA


////////////////////////////////////////////////////////////////////////////////
///////
////////////////////////////////////////////////////////////////////////////////
///////

en_screen4:cpi r27,$02
           brne en_screen27

;en_screen4:
          ldi r16,$01
              rcall cmd
              rcall delay1

              ldi r16,$53
              rcall lcdwr
          ldi r16,$57
              rcall lcdwr
              ldi r16,$20
              rcall lcdwr
              ldi r16,$4f
              rcall lcdwr
              ldi r16,$4e
              rcall lcdwr
              ldi r16,$c0
              rcall cmd
              ldi r18,$01
              ldi r17,$1b
          rcall rtyuu
          rcall bintobcd
              add r12,r17
              mov r16,r12
              rcall lcdwr
              add r18,r17
              mov r16,r18
              rcall lcdwr
              ldi r16,$3a
              rcall lcdwr
              ldi r18,$01
              ldi r17,$1c
          rcall rtyuu
          rcall bintobcd
              add r12,r17
              mov r16,r12
              rcall lcdwr
              add r18,r17
              mov r16,r18
              rcall lcdwr


              ldi r27,$61

              ldi r19,$30
              ldi r21,$c0

              ldi r16,$c0
              rcall cmd

              ldi r16,$0d
              rcall cmd

              rjmp keywait

;*******************************************
en_screen27:cpi r27,$61
            brne en_screen28

           rcall movey

                  ldi r18,$01
                  ldi r17,$16

              rcall ertyyy

            ldi r21,$c1
                  mov r16,r21
                  rcall cmd

            ldi r19,$30
                  ldi r27,$62
                  rjmp keywait


en_screen28:cpi r27,$62
            brne en_screen29

           rcall movey

                  ldi r18,$01
                  ldi r17,$17

              rcall ertyyy

            ldi r21,$c3
                  mov r16,r21
                  rcall cmd

            ldi r19,$30
                  ldi r27,$63

                rjmp keywait


en_screen29:cpi r27,$63
            brne en_screen30

         rcall movey

                  ldi r18,$01
                  ldi r17,$18

              rcall ertyyy

            ldi r21,$c4
                  mov r16,r21
                  rcall cmd

            ldi r19,$30
                  ldi r27,$64


                  rjmp keywait


en_screen30:cpi r27,$64
            breq en_sscreen31
              rjmp en_screen31

en_sscreen31:

      rcall movey

                  ldi r18,$01
                  ldi r17,$19

              rcall ertyyy
```

```
                ldi r18,$01
            ldi r17,$1d
                  rcall rtyuu
                  cpi r16,$00

               breq sw_on
           rjmp sw_off

sw_on:      ldi r18,$01
            ldi r17,$1d
                  ldi r16,$01
                  rcall ertyyy

                  ldi r18,$01
                  ldi r17,$16
                  rcall rtyuu

                  ldi r17,$0a
                  mul r16,r17
                  mov r21,r0

            ldi r18,$01
                  ldi r17,$17

                  rcall rtyuu

                  add r21,r16

                  ldi r18,$01
                  ldi r17,$1b
            mov r16,r21

                rcall ertyyy


                  ldi r18,$01
                  ldi r17,$18

                  rcall rtyuu

                  ldi r17,$0a
                  mul r16,r17
                  mov r21,r0

            ldi r18,$01
                  ldi r17,$19

                  rcall rtyuu
                  add r21,r16

                  ldi r18,$01
                  ldi r17,$1c
            mov r16,r21

                rcall ertyyy
```

```
    ldi r16,$01
    rcall cmd
    rcall delay1

    ldi r16,$53
    rcall lcdwr
ldi r16,$57
    rcall lcdwr
    ldi r16,$20
    rcall lcdwr
    ldi r16,$4f
    rcall lcdwr
    ldi r16,$46
    rcall lcdwr
    ldi r16,$46
    rcall lcdwr
    ldi r16,$c0
    rcall cmd

    ldi r18,$01
    ldi r17,$1e
rcall rtyuu
rcall bintobcd
    add r12,r17
    mov r16,r12
    rcall lcdwr
    add r18,r17
    mov r16,r18
    rcall lcdwr
    ldi r16,$3a
    rcall lcdwr
    ldi r18,$01
    ldi r17,$1f
rcall rtyuu
rcall bintobcd
    add r12,r17
    mov r16,r12
    rcall lcdwr
    add r18,r17
    mov r16,r18
    rcall lcdwr



    ldi r27,$61

    ldi r19,$30
    ldi r21,$c0

    ldi r16,$c0
    rcall cmd

    ldi r16,$0d
    rcall cmd

    rjmp keywait
```

```
sw_off:

            ldi r18,$01
            ldi r17,$1d
                  ldi r16,$00
                  rcall ertyyy

                  ldi r18,$01
                  ldi r17,$16
                  rcall rtyuu

                  ldi r17,$0a
                  mul r16,r17
                  mov r21,r0

            ldi r18,$01
                  ldi r17,$17

                  rcall rtyuu

                  add r21,r16

                  ldi r18,$01
                  ldi r17,$1e
            mov r16,r21

                rcall ertyyy


                  ldi r18,$01
                  ldi r17,$18

                  rcall rtyuu

                  ldi r17,$0a
                  mul r16,r17
                  mov r21,r0

            ldi r18,$01
                  ldi r17,$19

                  rcall rtyuu
                  add r21,r16

                  ldi r18,$01
                  ldi r17,$1f
            mov r16,r21

                rcall ertyyy


clr r27
clr r21
ldi r16,$0c
rcall cmd
```

```
sei

rjmp Auto_relay_switch


//CT Ratio
en_screen31:cpi r27,$03
            breq en_sscreen5
            rjmp en_screen21

en_sscreen5:
;CT ratio

           ldi r16,$01
               rcall cmd
               rcall delay1

               ldi r16,$45
               rcall lcdwr
           ldi r16,$6e
               rcall lcdwr
               ldi r16,$74
               rcall lcdwr
               ldi r16,$65
               rcall lcdwr
               ldi r16,$72
               rcall lcdwr
               ldi r16,$20
               rcall lcdwr
               ldi r16,$43
               rcall lcdwr
               ldi r16,$54
               rcall lcdwr
               ldi r16,$20
               rcall lcdwr

        ldi r16,$52
      rcall lcdwr
        ldi r16,$61
      rcall lcdwr
        ldi r16,$74
      rcall lcdwr
        ldi r16,$69
      rcall lcdwr
        ldi r16,$6f
      rcall lcdwr

      ldi r16,$c0
        rcall cmd

        ldi r18,$01
     ldi r17,$10
rcall rtyuu

      ldi r17,$30
      add r16,r17
```

```
        //ldi r16,$30
        rcall lcdwr


 ldi r18,$01
     ldi r17,$2c
rcall rtyuu

      ldi r17,$30
      add r16,r17


      //  ldi r16,$30
        rcall lcdwr



       ldi r18,$01
ldi r17,$1a
rcall rtyuu

       ldi r17,$30

       add r16,r17

        rcall lcdwr

        ldi r16,$30
        rcall lcdwr

        ldi r16,$30
        rcall lcdwr


      ldi r16,$3a
        rcall lcdwr

        ldi r16,$30
        rcall lcdwr

      ldi r16,$35
        rcall lcdwr

      ldi r16,$c0
        rcall cmd

      ldi r16,$0d
        rcall cmd


        ldi r19,$30
        ldi r21,$c0

        ldi r27,$31

        rjmp keywait
```

```
en_screen21:  cpi r27,$31
              brne en_screen22

      rcall movey

                   ldi r17,$10
                   ldi r18,$01

                   rcall ertyyy

                   ldi r21,$c1
                   ldi r19,$30
             mov r16,r21
                   rcall cmd

             ldi r27,$32

                   rjmp keywait

en_screen22: cpi r27,$32
             brne enn_screen23


         rcall movey

                   ldi r17,$2c
                   ldi r18,$01
                   rcall ertyyy

                   ldi r21,$c2
                   ldi r19,$30
             mov r16,r21
                   rcall cmd
             ldi r27,$33

                   rjmp keywait

enn_screen23:cpi r27,$33
             brne enn_screen24

            rcall movey


             ldi r17,$1a
                   ldi r18,$01
                   rcall ertyyy

                   ldi r21,$c3
                   ldi r19,$30
             mov r16,r21
                   rcall cmd

             ldi r27,$34

                   rjmp keywait
```

```
enn_screen24:cpi r27,$34
             brne enn_screen25

             ldi r21,$c4
                   ldi r19,$30
             mov r16,r21
                   rcall cmd
             ldi r27,$35

                   rjmp keywait

enn_screen25:cpi r27,$35
             brne enn_screen26
             ldi r21,$c6
                   ldi r19,$30
             mov r16,r21
                   rcall cmd
             ldi r27,$36
             rjmp keywait

enn_screen26:cpi r27,$36
             brne enn_screen27
             ldi r21,$c7
                   ldi r19,$30
             mov r16,r21
                   rcall cmd
             ldi r27,$37
             rjmp keywait

enn_screen27:cpi r27,$37
             brne en_screen6

                   clr r27

                   ldi r16,$0c
                   rcall cmd
                   rcall delay1

                   clr r21
                   clr r19

             sei

                   rjmp Auto_relay_switch


en_screen6:cpi r27,$04
           breq en_sscreen7
           rjmp en_screen7345

en_sscreen7:

         ldi r16,$01
               rcall cmd
         rcall delay1
```

```
ldi r18,$01
ldi r17,$2b
ldi r16,$01
rcall ertyyy

           ldi r16,$4d
               rcall lcdwr
               ldi r16,$61
               rcall lcdwr
               ldi r16,$6e
               rcall lcdwr
               ldi r16,$75
               rcall lcdwr
               ldi r16,$61
               rcall lcdwr
               ldi r16,$6c
               rcall lcdwr
               ldi r16,$20
               rcall lcdwr
               ldi r16,$4d
               rcall lcdwr
               ldi r16,$6f
               rcall lcdwr
               ldi r16,$64
               rcall lcdwr
               ldi r16,$65
               rcall lcdwr


               ldi r16,$c0
               rcall cmd

               ldi r16,$7e
               rcall lcdwr

           ldi r16,$4d
               rcall lcdwr
               ldi r16,$6f
               rcall lcdwr
               ldi r16,$64
               rcall lcdwr
               ldi r16,$65
               rcall lcdwr
           ldi r16,$30
               rcall lcdwr


           ldi r16,$c9
               rcall cmd

               ldi r16,$4d
               rcall lcdwr
               ldi r16,$6f
               rcall lcdwr
               ldi r16,$64
               rcall lcdwr
               ldi r16,$65
               rcall lcdwr
           ldi r16,$31
               rcall lcdwr

     ldi r27,$81

       clr r28
       clr r20

       ldi r19,$c0

       rjmp keywait



mode0:         ldi r27,$70
           ldi r16,$0c
               rcall cmd

measure12: sei
           rcall Measure_pf
               //rcall delay
               //rcall delay
               rjmp measure12



mode1:     ldi r27,$16
           ldi r16,$0c
               rcall cmd


measure:   sei
           rcall Measure_pf

               //rcall delay
               //rcall delay

               rjmp measure


en_screen7345 :
cpi r27,$77
breq en_screen73445
mov r16,r27
andi r16,$f0
cpi r16,$70
brne en_screen7353
inc r27

rjmp measure12

en_screen73445:
ldi r27,$50
rjmp measure12
```

```
en_screen7353 :
cpi r27,$53
breq en_screen73453

mov r16,r27
andi r16,$f0
cpi r16,$50
brne en_screen39
inc r27
rjmp measure12

en_screen73453:
ldi r27,$70
rjmp measure12

en_screen39:cpi r27,$81
            brne en_screen41
              cpi r19,$c0
                brne iuwerhd
            rjmp mode0


iuwerhd:    rjmp mode1





en_screen41:cpi r27,$16
            brne en_screen7

               ldi r27,$06
           rjmp en_screen6




en_screen7:cpi r27,$05
             breq en_sscreen78
           rjmp dead_1

en_sscreen78:

ldi r16,$01
rcall cmd
rcall delay1

ldi r17,$3e
ldi r18,$01
rcall rtyuu
mov r30,r16
cpi r16,$64
brne lsdgkks

lsdgkks:
ldi r16,$30
rcall lcdwr
```

```
ldi r16,$2e
rcall lcdwr

mov r16,r30
rcall bintobcd
add r12,r17
mov r16,r12
rcall lcdwr
add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$20
rcall lcdwr
ldi r16,$20
rcall lcdwr

ldi r17,$3f
ldi r18,$01
rcall rtyuu
ldi r17,$30
add r16,r17
rcall lcdwr

ldi r16,$c0
rcall cmd


ldi r16,$4c
rcall lcdwr
ldi r16,$4f
rcall lcdwr
ldi r16,$57
rcall lcdwr
ldi r16,$45
rcall lcdwr
ldi r16,$52
rcall lcdwr




ldi r16,$80
rcall cmd

ldi r16,$0d
rcall cmd

ldi r27,$a1
ldi r19,$30
ldi r21,$80
clr r30

rjmp keywait

dead_1:      cpi r27,$a1
               brne dead_2
```

```
             rcall movey

                   cpi r16,$01
                   brne goitt
               ser r31


goitt:
                   ldi r17,$16
                   ldi r18,$01

                   rcall ertyyy

             cpi r31,$ff
                   brne lskdfj

                   ldi r21,$86
                   ldi r19,$30
             mov r16,r21
                   rcall cmd

             ldi r27,$b1

                   rjmp keywait

lskdfj:      ldi r21,$82
                   ldi r19,$30
             mov r16,r21
                   rcall cmd

             ldi r27,$a2

                   rjmp keywait

dead_2:      cpi r27,$a2
               brne dead_3

             rcall movey

                   ldi r17,$17
                   ldi r18,$01

                   rcall ertyyy

                   ldi r21,$83
                   ldi r19,$30
             mov r16,r21
                   rcall cmd

             ldi r27,$a3

                   rjmp keywait

dead_3:      cpi r27,$a3
               brne dead_4
             rcall movey

                   ldi r17,$18
                   ldi r18,$01

                   rcall ertyyy

                   ldi r21,$86
                   ldi r19,$30
             mov r16,r21
                   rcall cmd

             ldi r27,$b1

                   rjmp keywait



dead_4:      cpi r27,$b1
               breq kjafueimr
                    rjmp en_screenn7

kjafueimr:   rcall movey

                   cpi r30,$01
                   brne kjasdfkjkerw
             rjmp lower_pt

kjasdfkjkerw:

                   ldi r17,$3f
                   ldi r18,$01

                   rcall ertyyy


             ldi r17,$16
                   rcall rtyuu

                   cpi r16,$01
                   brne goire

                   ldi r16,$64
                   ldi r17,$3e

                   rcall ertyyy

goire:             ldi r17,$17
             rcall rtyuu
                   ldi r17,$0a
                   mul r16,r17
                   ldi r17,$18
             rcall rtyuu

                   add r16,r0
                   ldi r17,$3e
                   rcall ertyyy
```

```
ldi r16,$01
rcall cmd
rcall delay1

ldi r17,$40
ldi r18,$01
rcall rtyuu
mov r30,r16
cpi r16,$64
brne lsdgkks1

lsdgkks1:
ldi r16,$30
rcall lcdwr
ldi r16,$2e
rcall lcdwr

mov r16,r30
rcall bintobcd
add r12,r17
mov r16,r12
rcall lcdwr
add r18,r17
mov r16,r18
rcall lcdwr

ldi r16,$20
rcall lcdwr
ldi r16,$20
rcall lcdwr

ldi r17,$41
ldi r18,$01
rcall rtyuu
ldi r17,$30
add r16,r17
rcall lcdwr

ldi r16,$c0
rcall cmd


ldi r16,$55
rcall lcdwr
ldi r16,$50
rcall lcdwr
ldi r16,$50
rcall lcdwr
ldi r16,$45
rcall lcdwr
ldi r16,$52
rcall lcdwr

ldi r16,$80
rcall cmd
```

```
ldi r16,$0d
rcall cmd

ldi r27,$a1
ldi r19,$30
ldi r21,$80
ldi r30,$01
rjmp keywait

lower_pt:
                   ldi r17,$3f
                   ldi r18,$01

                   rcall ertyyy
             ldi r17,$41
                   rcall rtyuu

                   cpi r16,$01
                   brne goire1

                   ldi r16,$64
                   ldi r17,$40

                   rcall ertyyy

goire1:            ldi r17,$17
             rcall rtyuu
                   ldi r17,$0a
                   mul r16,r17
                   ldi r17,$18
             rcall rtyuu

                   add r16,r0
                   ldi r17,$40
                   rcall ertyyy

ldi r16,$0c
rcall cmd



clr r27
clr r21
clr r19
clr r31
clr r30
 sei
rjmp Auto_relay_switch



en_screenn7:cpi r27,$06
           breq en_sscreen8
           rjmp en_screen8

;en_screen7:
```

```
en_sscreen8:
          ldi r16,$01
              rcall cmd
              rcall delay1
              ldi r16,$53
              rcall lcdwr
              ldi r16,$77
              rcall lcdwr
              ldi r16,$69
              rcall lcdwr
          ldi r16,$74
              rcall lcdwr
              ldi r16,$63
              rcall lcdwr
              ldi r16,$68
              rcall lcdwr
              ldi r16,$20
              rcall lcdwr
              ldi r16,$4d
              rcall lcdwr
              ldi r16,$65
              rcall lcdwr
              ldi r16,$74
              rcall lcdwr
          ldi r16,$68
              rcall lcdwr
              ldi r16,$6f
              rcall lcdwr
              ldi r16,$64
              rcall lcdwr
              ldi r16,$c0
              rcall cmd
              ldi r16,$7e
              rcall lcdwr
              ldi r16,$46
              rcall lcdwr
              ldi r16,$49
              rcall lcdwr         ;
              ldi r16,$46
              rcall lcdwr      ;3
              ldi r16,$4f
              rcall lcdwr       ;4
              ldi r16,$20
              rcall lcdwr        ;   5
              ldi r16,$20
              rcall lcdwr         ;   6   ;
              ldi r16,$4c
              rcall lcdwr   ;7
              ldi r16,$49
              rcall lcdwr    ;      8
              ldi r16,$46
              rcall lcdwr     ;       9
              ldi r16,$4f
              rcall lcdwr
              ldi r16,$20
              rcall lcdwr
              ldi r16,$20
              rcall lcdwr
              ldi r16,$48
              rcall lcdwr
              ldi r16,$45
              rcall lcdwr
              ldi r16,$58
              rcall lcdwr
              ldi r27,$c0
              clr r28
              rjmp keywait


en_screen8 :cpi r27,$c0
            brne en_screen19
              clr r29
              clr r27
                          ldi r18,$01
                  ldi r17,$11
               mov r16,r29
      rcall ertyyy
                  sei
                  rjmp Auto_relay_switch

en_screen19:cpi r27,$c6
            brne en_screen20
               ldi r29,$02
               ldi r18,$01
                  ldi r17,$11
mov r16,r29
                  rcall ertyyy


               clr r27
                  sei
                  rjmp Auto_relay_switch


en_screen20:cpi r27,$cc
            brne en_screen213
                ldi r29,$01
                        ldi r18,$01
                  ldi r17,$11
            mov r16,r29
                 rcall ertyyy

            clr r27
                   sei
                  rjmp Auto_relay_switch



en_screen213:cpi r27,$07
           breq en_sscreen9
           rjmp en_screen9
en_sscreen9:
;en_screen8:
```

```
                    ldi r16,$01
                    rcall cmd
                    rcall delay1

                    ldi r16,$53
                    rcall lcdwr
              ldi r16,$65
                    rcall lcdwr
                    ldi r16,$61
                    rcall lcdwr
                    ldi r16,$72
                    rcall lcdwr
                    ldi r16,$63
                    rcall lcdwr
                    ldi r16,$68
                    rcall lcdwr
                    ldi r16,$69
                    rcall lcdwr
                    ldi r16,$6e
                    rcall lcdwr
                    ldi r16,$67
                    rcall lcdwr
                    ldi r16,$2e
                    rcall lcdwr
                    ldi r16,$2e
                    rcall lcdwr
                    ldi r16,$2e
                    rcall lcdwr
              ldi r16,$2e
                    rcall lcdwr

rtyio:
cli
rcall Stop_wr_en

cbi ddrd,0
cbi ddrd,1

ldi r16,(1<<RXEN)|(1<<TXEN)
out UCSRB,r16

USART_receive:

sbis USR,RXC1
rjmp USART_Receive
in r16,UDR
cpi r16,$66
breq lkdfglk
rjmp USART_receive

lkdfglk :

rcall Start_wr_en
ldi r16,$a0
rcall I2C_WR_EN
clr r16
rcall Data_Write
```

```
clr r16
rcall Data_Write

rcall R_Start
ldi r16,$a1
rcall I2C_RD_EN

rcall send_data

rcall Start_wr_en
ldi r16,$a2
rcall I2C_WR_EN

clr r16
rcall Data_Write

clr r16
rcall Data_Write

rcall R_Start
ldi r16,$a3
rcall I2C_RD_EN

rcall send_data

clr r16
out UCSRB,r16

ldi r16,(1<<DDD1)|(1<<DDD0)|(1<<DDD4)|(1<<DDD5)|(1<<DDD7)
    out ddrd,r16

clr r27
clr r21
ldi r18,$01
ldi r16,$00
ldi r17,$2e
rcall ertyyy
ldi r17,$2f
rcall ertyyy
ldi r17,$2a
rcall ertyyy
ldi r16,$a0
ldi r17,$2d
rcall ertyyy


rcall init_lcd

sei

rjmp Auto_relay_switch



//Stage Selection
en_screen9: cpi r27,$08
            brne en_screen121
    ldi r16,$01
      rcall cmd
      rcall delay1

ldi r16,$31
rcall lcdwr
ldi r16,$32
rcall lcdwr
ldi r16,$33
rcall lcdwr
ldi r16,$34
rcall lcdwr
ldi r16,$35
rcall lcdwr
ldi r16,$36
rcall lcdwr
ldi r16,$37
rcall lcdwr
ldi r16,$38
rcall lcdwr
ldi r16,$20
rcall lcdwr
ldi r16,$31
rcall lcdwr
ldi r16,$32
rcall lcdwr
ldi r16,$33
rcall lcdwr
ldi r16,$34
rcall lcdwr

ldi r16,$c0
rcall cmd


ldi r19,$08
ldi r17,$30
ldi r18,$01
jumpai:
rcall rtyuu
ldi r30,$30
add r16,r30
rcall lcdwr
inc r17
dec r19
brne jumpai

ldi r16,$20
rcall lcdwr

ldi r19,$04
//ldi r17,$30
//ldi r18,$01
jumpai1:
rcall rtyuu
ldi r30,$30
```

```
add r16,r30
rcall lcdwr
inc r17
dec r19
brne jumpai1

ldi r16,$c0
rcall cmd
ldi r16,$0d
rcall cmd

ldi r27,$41
ldi r19,$30


ldi r21,$c0

ldi r30,$30
ldi r31,$01

rjmp keywait


en_screen121:
mov r16,r27
andi r16,$f0
cpi r16,$40
brne en_screen122
cpi r27,$48
brsh en_screen122
rcall pass
mov r16,r21
rcall cmd
rjmp keywait

en_screen122:
cpi r27,$48
brne en_screen123
rcall pass
inc r21
mov r16,r21
rcall cmd

clr r19

ldi r18,$01
ldi r17,$30
rcall rtyuu
cpi r16,$01
brne next121
sbr r19,$80
next121:
inc r17
rcall rtyuu
cpi r16,$01
brne next122
sbr r19,$40
```

```
next122:
inc r17
rcall rtyuu
cpi r16,$01
brne next123
sbr r19,$20
next123:
inc r17
rcall rtyuu
cpi r16,$01
brne next124
sbr r19,$10
next124:
inc r17
rcall rtyuu
cpi r16,$01
brne next125
sbr r19,$08
next125:
inc r17
rcall rtyuu
cpi r16,$01
brne next126
sbr r19,$04
next126:
inc r17
rcall rtyuu
cpi r16,$01
brne next127
sbr r19,$02
next127:
inc r17
rcall rtyuu
cpi r16,$01
brne next128
sbr r19,$01

next128:

mov r16,r19
ldi r17,$3c
ldi r18,$01
rcall ertyyy
ldi r27,$91
ldi r17,$38
clr r28
rjmp keywait




en_screen123:
mov r16,r27
andi r16,$f0
cpi r16,$90
brne en_screen124
cpi r27,$94
```

```
brsh en_screen124



rcall pass
mov r16,r21
rcall cmd
rjmp keywait

en_screen124:
cpi r27,$94
brne en_screen137
rcall pass

clr r19
ldi r18,$01
ldi r17,$38
rcall rtyuu
cpi r16,$01
brne next131
sbr r19,$80
next131:
inc r17
rcall rtyuu
cpi r16,$01
brne next132
sbr r19,$40
next132:
inc r17
rcall rtyuu
cpi r16,$01
brne next133
sbr r19,$20
next133:
inc r17
rcall rtyuu
cpi r16,$01
brne next134
sbr r19,$10
next134:

ldi r17,$3d
mov r16,r19
rcall ertyyy

clr r20

clr r27
clr r21
    ldi r16,$0c
              rcall cmd
              rcall delay1

            sei
            rjmp Auto_relay_switch
rjmp keywait



/*

;en_screen15:

ldi r16,$01
rcall cmd
rcall delay1

           ldi r16,$45
               rcall lcdwr
           ldi r16,$6e
               rcall lcdwr
               ldi r16,$74
               rcall lcdwr
               ldi r16,$65
               rcall lcdwr
               ldi r16,$72
           rcall lcdwr
               ldi r16,$20
               rcall lcdwr
           ldi r16,$4e
           rcall lcdwr
               ldi r16,$65
           rcall lcdwr
               ldi r16,$77
           rcall lcdwr
               ldi r16,$20
               rcall lcdwr
               ldi r16,$4b
           rcall lcdwr
               ldi r16,$45
           rcall lcdwr
               ldi r16,$59
           rcall lcdwr

ldi r16,$c6
```

```
rcall cmd
ldi r16,$58
rcall lcdwr
ldi r16,$58
rcall lcdwr
ldi r16,$58
rcall lcdwr
ldi r16,$58
rcall lcdwr
ldi r16,$c6
rcall cmd
ldi r16,$0d
rcall cmd

           ldi r19,$30
               ldi r27,$22
           ldi r21,$c6
               ldi r30,$12
               ldi r31,$01
//
           rjmp keywait



en_screen16:
mov r16,r27
cpi r16,$25
brsh en_screen131
rcall pass

            rjmp keywait

en_screen131: cpi r27,$25
               brne en_screen132

    rcall pass

      clr r21

                  clr r27
                  ldi r19,$30


                  ldi r17,$30
                  ldi r18,$01
                  ldi r16,$12
                  rcall ertyyy

          ldi r16,$0c
              rcall cmd
              rcall delay1

            sei
            rjmp Auto_relay_switch

en_screen132:
```

```
cpi r27,$29
            brsh en_screen133

         rcall pass_read



           rjmp keywait


en_screen133:cpi r27,$29
            breq en_screen136
            rjmp en_screen137

en_screen136:
rcall pass_read

        //   clr r27
           //  clr r19
              // clr r21

               ldi r16,$0c
               rcall cmd
               rcall delay1

           mov r16,r11
               cpi r16,$04
               brne noprte

               //clr r27
               rjmp gohert

noprte:  ldi r16,$01
         rcall cmd
             rcall delay1
             ldi r16,$57
         rcall lcdwr
         ldi r16,$72
         rcall lcdwr
             ldi r16,$6f
         rcall lcdwr
             ldi r16,$6e
         rcall lcdwr
             ldi r16,$67
         rcall lcdwr
             ldi r16,$20
         rcall lcdwr
             ldi r16,$50
         rcall lcdwr
             ldi r16,$61
         rcall lcdwr
             ldi r16,$73
         rcall lcdwr
             ldi r16,$73
         rcall lcdwr
             ldi r16,$77
         rcall lcdwr
```

```
             ldi r16,$6f
         rcall lcdwr
             ldi r16,$72
         rcall lcdwr
             ldi r16,$64
         rcall lcdwr
             cli
         rcall delay
             rcall delay
             rcall delay
         clr r21
             clr r27
             clr r19
             clr r11
         sei
             rjmp Auto_relay_switch

*/


en_screen137:cpi r27,$09
            brne en_screen138

;en_screen15:
clr r27
            sei
            rjmp Auto_relay_switch


en_screen138:

reti

;key_four ;key_four ;key_four ;key_four ;key_four ;key_four ;key_four ;key_four
;key_four ;key_four ;key_four ;key_four ;key_four ;key_four ;key_four ;key_four
;key_four ;key_four ;key_four ;key_four ;key_four ;key_four ;key_four ;key_four



key_four: ldi r17,$b0
          cpse r16,r17
              rjmp null

;key_four


;*********************************************
//Menu
      cpi r27,$00
        brne labels0080



paestrghttt:
/*
```

```
ldi r16,$01
 rcall cmd
 rcall delay1

   ldi r16,$45
   rcall lcdwr
 ldi r16,$6e
   rcall lcdwr
   ldi r16,$74
   rcall lcdwr
   ldi r16,$65
   rcall lcdwr
   ldi r16,$72
 rcall lcdwr

 ldi r16,$20
   rcall lcdwr

   ldi r16,$50
 rcall lcdwr
   ldi r16,$61
 rcall lcdwr
   ldi r16,$73
 rcall lcdwr
   ldi r16,$73
 rcall lcdwr
   ldi r16,$77
 rcall lcdwr
   ldi r16,$6f
 rcall lcdwr
   ldi r16,$72
 rcall lcdwr
   ldi r16,$64
 rcall lcdwr

 ldi r16,$c6
   rcall cmd

   ldi r16,$58
   rcall lcdwr
 ldi r16,$58
   rcall lcdwr
   ldi r16,$58
   rcall lcdwr
   ldi r16,$58
   rcall lcdwr

   ldi r16,$c6
   rcall cmd
   ldi r16,$0d
   rcall cmd

   ldi r27,$26

   ldi r19,$30

   ldi r21,$c6
```

```
               ldi r30,$12

               ldi r31,$01


               clr r11

               rjmp keywait


      */



        gohert:





        rcall screen1



keywait:sei

            rcall delay_1min

            clr r27
                  clr r21

                  ldi r16,$0c
                  rcall cmd

//rcall delay1

            sei
      rjmp Auto_relay_switch



labels0080:


ldi r17,$2b
ldi r18,$01
clr r16
rcall ertyyy


            clr r27
            ldi r16,$0c
                  rcall cmd
                  rcall delay1

                  sei
                  rjmp Auto_relay_switch


;**************************************************


null:     reti

; in this program for 16 X 4 LCD voltage and current is displayed with power
factor on the
; front screen
;***************************************************************************
```