APPENDIX

Assembly language codes for whole project

```
;adc address and control pins are defined
ADC_A       BIT     P2. 2
ADC_B       BIT     P2. 3
ADC_C       BIT     P2. 5
ADC_SC      BIT     P2. 0
ADC_ALE  BIT        P2. 4
ADC_EOC  BIT        P2. 1
TRIP_SIGNAL BIT P3. 7
FAULT_IND BIT P3. 6
UP BIT P3. 5
DOWN BIT P3. 3
LEFT BIT P3. 4
RIGHT BIT P3. 2
LED_RESET BIT P3. 1

;lcd control pins are defined
RS      BIT     P2. 7
EN      BIT     P2. 6

 ;program starts from this address
                                        ORG  0000H
                                        SETB PSW. 4
                                        MOV R3,#2D
                                        MOV R5,#0D
                                        SETB PSW. 3
                                        MOV R2,#2D
                                        MOV R5,#0D        ;definite time setting
                                        MOV R3,#0D        ;value for displaying relay char
```

```
                              MOV R4,#5D;
                              CLR PSW. 4
                              CLR PSW. 3
                              CLR FAULT_IND
                              CLR TRIP_SIGNAL
                              LJMP MAIN
;**********************************************************
    INITIALIZE_LCD:                   ;lcd initialization sub-routine
                              MOV   A, #38H
                              LCALL  WRITE_CMD
                              MOV   A, #0CH
                              LCALL  WRITE_CMD
                              MOV   A, #06H
                              LCALL  WRITE_CMD
                              RET
;**********************************************************
    WRITE_CMD:          ;sub-routine to write a command to lcd's instruction register
                              CLR    RS                 ;rs=0 for selecting command register
                              MOV   P0, A
                              SETB  EN
                              CLR    EN
                              LCALL  LCD_DELAY
                              RET
;**********************************************************
    WRITE_CHAR:                       ;sub-routine to write a data to lcd's data register
                              SETB  RS             ;rs=1 for selecting data register
                              MOV   P0, A
                              SETB  EN
                              CLR    EN
                              LCALL  LCD_DELAY
                              RET
```

```
;************************************************************
        WRITE_STRING:           CLR A
                                MOVC A,@A+DPTR
                                JZ EXIT
                                MOV P0, A
                                SETB RS
                                SETB EN
                                ACALL LCD_DELAY
                                CLR EN
                                INC DPTR
                                SJMP WRITE_STRING
        EXIT:                   RET
;*************************************************************
        LCD_DELAY:             ;to generate a delay between consequent lcd write operations
                                MOV   R0, #5H
        L2:                     MOV   R1, #0FFH
        L1:                     DJNZ  R1, L1
                                DJNZ  R0, L2
                                RET
;*************************************************************
        DISPLAY_CURRENT:                    ;subroutine to display 'Current:' on lcd
                                MOV   A, #80H
                                LCALL  WRITE_CMD
                                ACALL LCD_DELAY
                                MOV DPTR,#DISPLAY_MAIN_SCREEN
                                ACALL WRITE_STRING
                                ACALL LCD_DELAY
                                RET
;**********************************************************
        DISPLAY:       ;subroutine to display the digits of the measured voltage
                                MOV   A, R6         ;display the digit before decimal point
```

```
                        ADD    A, #30H
                        LCALL WRITE_CHAR
                        MOV   A, #2EH                 ;display the decimal point
                        LCALL WRITE_CHAR
                        MOV   A, R5        ;display the digit after the decimal point
                        ADD    A, #30H
                        LCALL WRITE_CHAR
                        MOV   A, R2    ;display the 2ND digit after the decimal point
                        ADD    A, #30H
                        LCALL  WRITE_CHAR
                        MOV   A, #'A'
                        ACALL  WRITE_CHAR
                        ACALL  LCD_DELAY
                        RET
;*****************************************************************
    READ_ADC:           ;sub-routine to read from adc
                        SETB  ADC_EOC
                        CLR    ADC_ALE
                        CLR    ADC_SC
                        CLR    ADC_A       ;channel 0 is selected
                        CLR    ADC_B
                        CLR    ADC_C
                        SETB  ADC_ALE    ;latch the address of the channel to adc
                        SETB  ADC_SC                  ; start the conversion
                        CLR    ADC_ALE
                        CLR    ADC_SC
    WAIT:               JNB ADC_EOC, WAIT       ;conversion complete
                        NOP
                        MOV A,P1
                        NOP
                        SETB ADC_EOC
```

```
                                RET
;***********************************************************
        FIND_MAX_AMPLITUDE:         ;subroutine to find the max amplitude
                                MOV   R7, #0FFH
        REPEAT:                 DEC R7
                                LCALL  READ_ADC
         ;find the maximum sampled current amplitude
        NEXT:                   MOV   R2, A   ;store the present current amplitude in R2
                                MOV   A, R3   ;load accumulator with previous value
                                SUBB  A, R2         ; subtract current from previous value
                ;               JC MAXIMUM_AMPLITUDE
                                AJMP  CHECK
        MAXIMUM_AMPLITUDE:
                                MOV   A, R2
                                MOV   R3, A
        CHECK:                  MOV   A, R7
                                JNZ  REPEAT
                                RET
;*********************************************************
        RMS_FIND:
        ;scale down the input adc value by 5 for DC or  by 7 for AC
                                MOV   A, R3
                                MOV   B, #7D
                                DIV   AB
                                MOV   R4, A
                                MOV R2, B

        ;convert the hex to two separate decimal digits to be displayed
        HEX_TO_DECIMAL:
                                MOV   A, R4
                                MOV   B, #10D           ;divide by 10
```

```
                        DIV     AB
                        MOV   R5, B        ;r5 contains the value after decimal
                        MOV   R6, A       ;r6 contains the value before decimal point
                        RET
;***************************************************
COMPARE_WITH_SET_VALUE:
                        MOV A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R6
                        ADD A,R5
                        MOV R4,A
                        ADD A,R4
                        ADD A,R4
                        ADD A,R4
                        ADD A,R4
                        ADD A,R4
                        ADD A,R4
                        ADD A,R2
                        MOV R4,A
                        SETB PSW. 4
                        MOV A,R3
                        ADD A,R3
                        ADD A,R3
                        ADD A,R3
ADD A,R3
ADD A,R3
ADD A,R3
ADD A,R3
ADD A,R3
ADD A,R3
ADD A,R5
SETB PSW. 3
MOV R0,A
ADD A,R0
ADD A,R0
ADD A,R0
ADD A,R0
ADD A,R0
ADD A,R0
ADD A,R5
CLR PSW. 3
SETB PSW. 3
MOV R0,A
CLR PSW. 3
CLR PSW. 4
SETB PSW. 4
SETB PSW. 3
MOV A,R0
CLR PSW. 4
CLR PSW. 3
SUBB A,R4
JC CROSSED_SETPOINT

MOV A,#3CH          ; activate second line
ACALL WRITE_CMD
```

```
                        ACALL LCD_DELAY
                        MOV A,#0C0H                   ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV DPTR,#NOT_CROSSED_SP
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
                        CLR TRIP_SIGNAL
                        RET
CROSSED_SETPOINT:
                        SETB PSW. 4
                        SETB PSW. 3
COM_CHAR1:              CJNE R2,#1D,COM_CHAR2
                        CLR PSW. 4
                        CLR PSW. 3
                        LCALL TRIP_DELAY_FOR_2SEC
                        LCALL TRIP_COMMAND
                        RET
COM_CHAR2:              CJNE R2,#2D,COM_CHAR3
                        CLR PSW. 4
                        CLR PSW. 3
                        LCALL TRIP_DELAY_FOR_2SEC
                        LCALL TRIP_DELAY_FOR_2SEC
                        LCALL TRIP_COMMAND
                        RET
COM_CHAR3:              CJNE R2,#3D,COM_CHAR4
                        CLR PSW. 4
                        CLR PSW. 3
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_COMMAND
                        RET
```

```
COM_CHAR4:              CJNE R2,#4D,COM_CHAR5
                        CLR PSW. 4
                        CLR PSW. 3
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_COMMAND
                        RET
COM_CHAR5:              CJNE R2,#5D,COM_CHAR6
                        CLR PSW. 4
                        CLR PSW. 3
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_COMMAND
                        RET
COM_CHAR6:              CJNE R2,#6D,COM_CHAR1
                        CLR PSW. 4
                        CLR PSW. 3
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_DELAY_FOR_5SEC
                        LCALL TRIP_COMMAND
                        RET

TRIP_COMMAND:
                        SETB TRIP_SIGNAL
                        MOV A, #01H
                        LCALL  WRITE_CMD
ROTATE:
                        MOV   A, #81H
                        LCALL  WRITE_CMD
                        ACALL LCD_DELAY
                        MOV DPTR,#CROSSED_SP
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
;second line display
                        MOV A,#3CH                ; activate second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV A,#0C0H               ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV DPTR,#CROSSED_SP2
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
                        ACALL DISPLAY
                        JNB  LED_RESET, OUT
                        AJMP ROTATE
OUT:                    MOV A, #01H
                        LCALL WRITE_CMD
                        ACALL  LCD_DELAY
                        RET
RELAY_SETTING:
                        MOV A, #01H
                        LCALL  WRITE_CMD
                        MOV   A, #81H
                        LCALL  WRITE_CMD
                        ACALL LCD_DELAY
                        MOV DPTR,#RSTG
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
```

```
MOV DPTR,#RSTG2
ACALL WRITE_STRING
ACALL LCD_DELAY
LCALL DISPLAY_R0
LCALL DISPLAY_2E
LCALL DISPLAY_R5
LCALL DISPLAY_R10
MOV A,#3CH; ACTIVATE SECOND LINE
ACALL WRITE_CMD
ACALL LCD_DELAY
MOV A,#0C1H                  ;jump to second line
ACALL WRITE_CMD
ACALL LCD_DELAY
MOV DPTR,#RSTS
ACALL WRITE_STRING
ACALL LCD_DELAY
SETB PSW. 3
SETB PSW. 4
MOV R3,#1D
CLR PSW. 3
CLR PSW. 4
ACALL DI_CHAR
SETB PSW. 3
SETB PSW. 4
MOV R3,#0D
CLR PSW. 3
CLR PSW. 4
LCALL DELAY_FOR_10SEC
LCALL DELAY_FOR_10SEC
JNB RIGHT,NEXT_PAGE
RET
```

```
NEXT_PAGE:          MOV A, #01H
                    LCALL  WRITE_CMD
                    MOV   A, #81H
                    LCALL  WRITE_CMD
                    ACALL  LCD_DELAY
                    MOV DPTR,#RSTGNEXTPAGE
                    ACALL WRITE_STRING
                    ACALL LCD_DELAY
NEXT_PAGEUC:        MOV A,#3CH          ; activate second line
                    ACALL WRITE_CMD
                    ACALL LCD_DELAY
                    MOV A,#0C4H         ;jump to second line
                    ACALL WRITE_CMD
                    ACALL LCD_DELAY
                    MOV A,#0EH
                    ACALL WRITE_CMD
                    ACALL LCD_DELAY
                    MOV A,#06H
                    ACALL WRITE_CMD
                    ACALL LCD_DELAY
                    MOV   A, #'I'
                    ACALL  WRITE_CHAR
                    ACALL  LCD_DELAY
                    MOV   A, #':'
                    ACALL  WRITE_CHAR
                    ACALL  LCD_DELAY
                    LCALL DISPLAY_R0
                    LCALL DISPLAY_2E
                    LCALL DISPLAY_R5
                    LCALL DISPLAY_R10
```

```
                                MOV   A, #'A'
                                ACALL  WRITE_CHAR
                                ACALL  LCD_DELAY
                                MOV A,#10H
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY
                                MOV A,#10H
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY

CHECK_LEFT_R1:JB LEFT,OP_R1                                       ;left if not pressed
CHECK_LONG_LEFT_R1:             LCALL  DELAY_FOR_10SEC              ;call for wait
                                JB LEFT,LEFT_MOVE_R5                ;left not pressed
                                MOV A, #01H
                                LCALL  WRITE_CMD
                                RET

OP_R1:
CHECK_UP_R1:                    JB UP,CHECK_DOWN_R1              ;checking increment port
                                SETB PSW. 4
                                 SETB PSW. 3
                                INC R5
                                MOV A,#0C9H                          ;jump to second line
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY
                                ACALL DISPLAY_R10
                                MOV A,#10H
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY
                                CLR PSW. 4
                                CLR PSW. 3
```

```
                        LCALL  DELAY_FOR_10SEC
CHECK_DOWN_R1:          JB DOWN ,CHECK_RIGHT_R1
                        SETB PSW. 4
                        SETB PSW. 3
                        DEC R5
                        MOV A,#0C9H   ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        ACALL DISPLAY_R10
                        MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        CLR PSW. 4
                        CLR PSW. 3

CHECK_RIGHT_R1:         LCALL  DELAY_FOR_10SEC
                        JB RIGHT,CHECK_LEFT
                        LJMP SET_CHAR

CHECK_LEFT:             AJMP CHECK_LEFT_R1
                        RET
LEFT_MOVE_R5:           MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY

CHECK_LEFT_R5:          JB LEFT,OP_R5
CHECK_LONG_LEFT_R2:     LCALL  DELAY_FOR_10SEC          ;call for wait
                        JB LEFT,LEFT_MOVE_R0     ;left not pressed
                        MOV A, #01H
                        LCALL WRITE_CMD
                        RET
```

```
OP_R5:
CHECK_UP_R5:            JB UP,CHECK_DOWN_R5         ;checking increment port
                        SETB PSW. 4
                        INC R5
                        MOV A,#0C8H                   ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        ACALL DISPLAY_R5
                        MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        CLR PSW. 4
                        LCALL  DELAY_FOR_10SEC
CHECK_DOWN_R5:          JB DOWN ,CHECK_LEFT_R5
                        SETB PSW. 4
                        DEC R5
                        MOV A,#0C8H                    ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        ACALL DISPLAY_R5
                        MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        CLR PSW. 4
                        LCALL  DELAY_FOR_10SEC
                        AJMP CHECK_LEFT_R5
                        RET
LEFT_MOVE_R0:           MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
CHECK_LEFT_R0:          JNB RIGHT,JUMP_RIGHT
                        JB LEFT, OP_R0
CHECK_LONG_LEFT_R0:     LCALL DELAY_FOR_10SEC          ;call for wait
                        JB LEFT,OP_R0                  ;left not pressed
                        MOV A, #01H
                        LCALL WRITE_CMD
                        RET

    OP_R0:
CHECK_UP_R0:            JB UP,CHECK_DOWN_R0          ;checking increment port
                        SETB PSW. 4
                        INC R3
                        MOV A,#0C6H                  ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        ACALL DISPLAY_R0
                        MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        CLR PSW. 4
                        LCALL  DELAY_FOR_10SEC
CHECK_DOWN_R0:          JB DOWN ,CHECK_LEFT_R0
                        SETB PSW. 4
                        DEC R3
                        MOV A,#0C6H   ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        ACALL DISPLAY_R0
                        MOV A,#10H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        CLR PSW. 4
                        LCALL       DELAY_FOR_10SEC
                        AJMP CHECK_LEFT_R0
                        RET
      JUMP_RIGHT:       MOV A,#14H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV A,#14H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV A,#14H
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        LJMP CHECK_LEFT_R1
      NEXT_PAGEUCC:     AJMP NEXT_PAGEUCC
                        RET
;***************************************************
      SET_CHAR:         MOV A, #01H
                        LCALL  WRITE_CMD
                        MOV   A, #81H
                        LCALL  WRITE_CMD
                        MOV DPTR,#CHAR1
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
                        MOV A,#3CH          ; activate second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
      SET_DEF:          MOV A,#0C4H   ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV DPTR,#CHAR_TYPE_DEF
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
CHECK_LEFT0:            JB LEFT,CHECK_UP0
                        LCALL DELAY_FOR_10SEC
                        JB LEFT,CHECK_UP0
                        RET
CHECK_UP0:              JB UP,CHECK_DOWN0
                        MOV A,#0C4H            ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
SET_INV:                MOV DPTR,#CHAR_TYPE_INV
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
CHECK_DOWN0:            JB DOWN,CHECK_LEFT0
                        MOV A,#0C4H          ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV DPTR,#CHAR_TYPE_DEF
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
CHECK_RIGHT0:           JB RIGHT,CHECK_UP2
                        AJMP SET_TIME_DEF_CHAR
CHECK_UP2:              JB UP,CHECK_LEFT2
                        AJMP SET_INV
CHECK_LEFT2:            JB LEFT,CHECK_RIGHT0
                         LCALL DELAY_FOR_10SEC
                        JB LEFT,CHECK_RIGHT0
                        MOV A, #01H
                        LCALL  WRITE_CMD
```

```
                                RET
    ;****************************************************
SET_TIME_DEF_CHAR:
                                MOV A, #01H
                                LCALL  WRITE_CMD
                                MOV   A, #81H
                                LCALL  WRITE_CMD
                                MOV DPTR,#CHAR_TIME
                                ACALL WRITE_STRING
                                ACALL LCD_DELAY
                                MOV A,#3CH              ; activate second line
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY
                                MOV A,#0C7H   ;JUMP TO SECOND LINE
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY
    DI_CHAR:
                                SETB PSW. 4
                                SETB PSW. 3
    COM1:                       CJNE R2,#1D,COM2
                                CLR PSW. 3
                                CLR PSW. 4
    COM11:
                                MOV A,#0C7H             ;jump to second line
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY
                                MOV DPTR,#D2
                                ACALL WRITE_STRING
                                ACALL LCD_DELAY
                                SETB PSW. 4
                                SETB PSW. 3
```

```
                        MOV R2,#1D
                        CJNE R3,#0D,DWN1
                        CLR PSW. 3
                        CLR PSW. 4
CHECK_LEFT31:          JB LEFT,CHECK_UP31
                        LCALL DELAY_FOR_10SEC
                        JB LEFT,CHECK_UP31
                        MOV A, #01H
                        LCALL   WRITE_CMD
DWN1:                   RET
CHECK_UP31:             JB UP,CHECK_LEFT31
                        LCALL  DELAY_FOR_10SEC
                        AJMP COM22
COM2:                   CJNE R2,#2D,COM3
                        CLR PSW. 3
                        CLR PSW. 4
COM22:                  MOV A,#0C7H             ;jump to second line
                        ACALL WRITE_CMD
                        ACALL LCD_DELAY
                        MOV DPTR,#D4
                        ACALL WRITE_STRING
                        ACALL LCD_DELAY
                        SETB PSW. 4
                        SETB PSW. 3
                        MOV R2,#2D
                        CJNE R3,#0D,DWN2
                        CLR PSW. 3
                        CLR PSW. 4
                        CHECK_LEFT32:JB LEFT,CHECK_UP32
                        LCALL DELAY_FOR_10SEC
                        JB LEFT,CHECK_UP32
```

```
                 MOV A, #01H
                 LCALL  WRITE_CMD
DWN2:            RET
CHECK_UP32:      JB UP,CHECK_DOWN32
                 LCALL DELAY_FOR_10SEC
                 AJMP COM33
CHECK_DOWN32:    JB DOWN,CHECK_LEFT32
                 LCALL  DELAY_FOR_10SEC
                 AJMP COM11
COM3:            CJNE R2,#3D,COM4
                 CLR PSW. 3
                 CLR PSW. 4
COM33:           MOV A,#0C7H              ;jump to second line
                 ACALL WRITE_CMD
                 ACALL LCD_DELAY
                 MOV DPTR,#D5
                 ACALL WRITE_STRING
                 ACALL LCD_DELAY
                 SETB PSW. 4
                 SETB PSW. 3
                 MOV R2,#3D
                 CJNE R3,#0D,DWN3
                 CLR PSW. 3
                 CLR PSW. 4
CHECK_LEFT33:    JB LEFT,CHECK_UP33
                 LCALL DELAY_FOR_10SEC
                 JB LEFT,CHECK_UP33
                 MOV A, #01H
                 LCALL  WRITE_CMD
DWN3:            RET
CHECK_UP33:      JB UP,CHECK_DOWN33
```

```
                LCALL  DELAY_FOR_10SEC
                AJMP COM44
CHECK_DOWN33:   JB DOWN,CHECK_LEFT33
                LCALL  DELAY_FOR_10SEC
                AJMP COM22
COM4:           CJNE R2,#4D,COM5
                CLR PSW. 3
                CLR PSW. 4
COM44:          MOV A,#0C7H              ;jump to second line
                ACALL WRITE_CMD
                ACALL LCD_DELAY
                MOV DPTR,#D10
                ACALL WRITE_STRING
                ACALL LCD_DELAY
                SETB PSW. 4
                SETB PSW. 3
                MOV R2,#4D
                CJNE R3,#0D,DWN4
                CLR PSW. 3
                CLR PSW. 4
CHECK_LEFT34:   JB LEFT,CHECK_UP34
                LCALL DELAY_FOR_10SEC
                JB LEFT,CHECK_UP34
                MOV A, #01H
                LCALL  WRITE_CMD
DWN4:           RET
CHECK_UP34:     JB UP,CHECK_DOWN34
                LCALL  DELAY_FOR_10SEC
                AJMP COM55
CHECK_DOWN34:   JB DOWN,CHECK_LEFT34
                LCALL  DELAY_FOR_10SEC
                AJMP COM33
COM5:           CJNE R2,#5D,COM6
                CLR PSW. 3
                CLR PSW. 4
COM55:          MOV A,#0C7H             ;jump to second line
                ACALL WRITE_CMD
                ACALL LCD_DELAY
                MOV DPTR,#D15
                ACALL WRITE_STRING
                ACALL LCD_DELAY
                SETB PSW. 4
                SETB PSW. 3
                MOV R2,#5D
                CJNE R3,#0D,DWN5
                CLR PSW. 3
                CLR PSW. 4
CHECK_LEFT35:   JB LEFT,CHECK_UP35
                LCALL DELAY_FOR_10SEC
                JB LEFT,CHECK_UP35
                MOV A, #01H
                LCALL  WRITE_CMD
DWN5:           RET
CHECK_UP35:     JB UP,CHECK_DOWN35
                LCALL DELAY_FOR_10SEC
                AJMP COM66
CHECK_DOWN35:   JB DOWN,CHECK_LEFT35
                LCALL DELAY_FOR_10SEC
                AJMP COM44
COM6:           CJNE R2,#6D,COM111
                CLR PSW. 3
                CLR PSW. 4
```

```
            COM66:              MOV A,#0C7H             ;jump to second line
                                ACALL WRITE_CMD
                                ACALL LCD_DELAY
                                MOV DPTR,#D20
                                LCALL WRITE_STRING
                                LCALL LCD_DELAY
                                SETB PSW. 4
                                SETB PSW. 3
                                MOV R2,#6D
                                CJNE R3,#0D,DWN6
                                CLR PSW. 3
                                CLR PSW. 4
       CHECK_LEFT36:            JB LEFT,CHECK_DOWN36
                                LCALL DELAY_FOR_10SEC
                                JB LEFT,CHECK_DOWN36
                                MOV A, #01H
                                LCALL  WRITE_CMD
        DWN6:                   RET
       CHECK_DOWN36:            JB DOWN,CHECK_LEFT36
                                LCALL  DELAY_FOR_10SEC
                                LJMP COM55
       COM111:                  LJMP COM1
                                RET
;***************************************************************
       SET_VALUE_REGISTER_DISPLAY:
                                SETB PSW. 4
                                MOV   A, R0          ;display the digit before decimal point
                                ADD   A, #30H
                                LCALL  WRITE_CHAR
                                ACALL  LCD_DELAY
                                MOV   A, #2EH
                                ACALL  WRITE_CHAR
                                ACALL  LCD_DELAY
                                MOV   A, R5           ;display the digit before decimal point
                                ADD   A, #30H
                                LCALL  WRITE_CHAR
                                ACALL  LCD_DELAY
                                SETB PSW. 3
                                MOV   A, R5           ;display the digit before decimal point
                                ADD   A, #30H
                                LCALL  WRITE_CHAR
                                ACALL  LCD_DELAY
                                CLR PSW. 4
                                CLR PSW. 3
                                RET
;*******************************************************************
     DISPLAY_R10:
                                SETB PSW. 4
                                SETB PSW. 3
                                MOV   A, R5       ;display the 2ND digit after decimal point
                                ADD   A, #30H
                                LCALL  WRITE_CHAR
                                ACALL  LCD_DELAY
                                CLR PSW. 4
                                CLR PSW. 3
                                RET
     DISPLAY_R5:                SETB PSW. 4
                                MOV   A, R5           ;display the digit after decimal point
                                ADD   A, #30H
                                LCALL  WRITE_CHAR
                                ACALL  LCD_DELAY
                                CLR PSW. 4
```

```
                      RET
      DISPLAY_R0:           SETB PSW. 4
                            MOV   A, R3          ;display the digit before decimal point
                            ADD   A, #30H
                            LCALL  WRITE_CHAR
                            ACALL  LCD_DELAY
                            CLR PSW. 4
                            RET
      DISPLAY_2E:           MOV   A, #2EH        ;display the decimal point
                            ACALL  WRITE_CHAR
                            ACALL  LCD_DELAY
                            RET
;***********************************************************************
      DELAY_FOR_10SEC:      MOV R4,#50D
      LL3:                  MOV   R0, #50H
      LL2:                  MOV   R1, #55H
      LL1:                  DJNZ  R1, LL1
                            DJNZ  R0, LL2
                            DJNZ R4,LL3
                            MOV R4,#0D
                            RET
;*************************************************************************
      TRIP_DELAY_FOR_2SEC:  MOV R4,#25D
      LL32:                 MOV   R0, #100H
      LL22:                 MOV   R1, #183H
      LL12:                 DJNZ  R1, LL12
                            SETB FAULT_IND
                            DJNZ  R0, LL22
                            CLR FAULT_IND
                            DJNZ R4,LL32
                            MOV R4,#0D
                            MOV R0,#0D
                            MOV R1,#0D
                            RET
    TRIP_DELAY_FOR_5SEC:
                            MOV R4,#100D
    LL35:                   MOV   R0, #199H
    LL25:                   MOV   R1, #184H
    LL15:                   DJNZ  R1, LL15
                            DJNZ  R0, LL25
                            SETB FAULT_IND
                            DJNZ R4,LL35
                            MOV R4,#0D
                            MOV R0,#0D
                            MOV R1,#0D
                            RET
;********************************************************
    RSTG:                   DB "CSET. :",0
    RSTG2:                  DB "I:",0
    RSTS:                   DB "TSET. :",0
    RSTS2:                  DB "T:",0
    RSTGNEXTPAGE:           DB "OVER CURRENT:",0
    CROSSED_SP:             DB "Fault in Line ",0
    CROSSED_SP2:            DB "Current I:",0
    NOT_CROSSED_SP:         DB "No Fault in Line ",0
    DISPLAY_MAIN_SCREEN:DB "Current:",0
    CHAR1:                  DB "SET RELAR CHAR. ",0
    CHAR_TYPE_INV:          DB "Inverse",0
    CHAR_TYPE_DEF:          DB "Definite",0
    CHAR_TIME:              DB "SET DEF.  TIME",0
    D2:                     DB "2 Sec",0
    D4:                     DB "4 Sec",0
```

```
        D5:                     DB "5 Sec",0
        D10:                    DB "10 Sec",0
        D15:                    DB "15 Sec",0
        D20:                    DB "20 Sec",0
;***********************************************************
         MAIN:                  MOV   R3, #0H           ;default max amplitude
                                MOV   P1, #0FFH      ;set p1 input port
                                MOV   P0, #0H           ;set p0 output port
                                LCALL INITIALIZE_LCD
                                LCALL  DISPLAY_CURRENT
                                LCALL  FIND_MAX_AMPLITUDE
                                LCALL  RMS_FIND
                                LCALL  DISPLAY
                                LCALL  COMPARE_WITH_SET_VALUE
                                JB RIGHT,NEXT_LINE        ;right key not pressed
                                LCALL RELAY_SETTING
        NEXT_LINE:              MOV A,#80H
                                LCALL  WRITE_CMD
                                AJMP   MAIN
                                END
```