

PROJECT: BUILDING CONTROL SYSTEM
TEAM: AGUILAR, SEAN KARL TYRESE G.
ALOTAYA, ISAAC JADON R.
CAÑOS, BEA BELLE THERESE B.

```
PROCED1 SEGMENT 'CODE'
ISR1 PROC FAR
ASSUME CS:PROCED1, DS:DATA
ORG 00000H
    PUSHF
    PUSH AX
    PUSH DX
    MOV AL, 00H
    OUT PORTJ, AL
    MOV FAN1_FLAG, 0
    MOV FAN2_FLAG, 0
    MOV ROOM1_WARNING_FLAG, 0
    POP DX
    POP AX
    POPF
    IRET
ISR1 ENDP
PROCED1 ENDS
```

```
PROCED2 SEGMENT 'CODE'
ISR2 PROC FAR
ASSUME CS:PROCED2, DS:DATA
ORG 00100H
    PUSHF
    PUSH AX
    PUSH DX
    MOV AL, 00H
    OUT PORTL, AL
    MOV FAN3_FLAG, 0
    MOV FAN4_FLAG, 0
    MOV ROOM2_WARNING_FLAG, 0
    POP DX
    POP AX
    POPF
    IRET
ISR2 ENDP
PROCED2 ENDS
```

```
PROCED3 SEGMENT 'CODE'
ISR3 PROC FAR
ASSUME CS:PROCED3, DS:DATA
```

```
ORG 00200H
    PUSHF
    PUSH AX
    PUSH DX

    MOV AL, 00H
    OUT PORTN, AL
    MOV FAN5_FLAG, 0
    MOV FAN6_FLAG, 0
    MOV ROOM3_WARNING_FLAG, 0

    POP DX
    POP AX
    POPF
    IRET
ISR3 ENDP
PROCED3 ENDS
```

```
PROCED4 SEGMENT 'CODE'
ISR4 PROC FAR
ASSUME CS:PROCED4, DS:DATA
ORG 00300H
    PUSHF
    PUSH AX
    PUSH DX

    CMP ROOM1_SCHED_ON_FLAG, 1
    JE ROOM1_ON_FANS
    CMP ROOM1_SCHED_OFF_FLAG, 1
    JE ROOM1_OFF_FANS
    JMP EXIT_ISR4
```

```
ROOM1_ON_FANS:
MOV AL, 03H
OUT PORTJ, AL
MOV AL, 01H
OUT PORTK, AL
MOV FAN1_FLAG, 1
MOV FAN2_FLAG, 1
JMP EXIT_ISR4
```

```
ROOM1_OFF_FANS:
MOV AL, 00H
OUT PORTJ, AL
MOV AL, 00H
OUT PORTK, AL
MOV FAN1_FLAG, 0
MOV FAN2_FLAG, 0
JMP EXIT_ISR4

EXIT_ISR4:
POP DX
POP AX
POPF
IRET
ISR4 ENDP
PROCED4 ENDS
```

```
PROCED5 SEGMENT 'CODE'
ISR5 PROC FAR
ASSUME CS:PROCED5, DS:DATA
ORG 00400H
    PUSHF
    PUSH AX
    PUSH DX

    CMP ROOM2_SCHED_ON_FLAG, 1
    JE ROOM2_ON_FANS
    CMP ROOM2_SCHED_OFF_FLAG, 1
    JE ROOM2_OFF_FANS
    JMP EXIT_ISR5
```

```
ROOM2_ON_FANS:
MOV AL, 03H
OUT PORTL, AL
MOV AL, 01H
OUT PORTM, AL
MOV FAN3_FLAG, 1
MOV FAN4_FLAG, 1
JMP EXIT_ISR5
```

```
ROOM2_OFF_FANS:
```

```

MOV AL, 00H
OUT PORTL, AL
MOV AL, 00H
OUT PORTM, AL
MOV FAN3_FLAG, 0
MOV FAN4_FLAG, 0
JMP EXIT_ISR5

EXIT_ISR5:
POP DX
POP AX
POPF
IRET
ISR5 ENDP
PROCED5 ENDS

PROCED6 SEGMENT 'CODE'
ISR6 PROC FAR
ASSUME CS:PROCED6, DS:DATA
ORG 00500H
    PUSHF
    PUSH AX
    PUSH DX

    CMP ROOM3_SCHED_ON_FLAG, 1
    JE ROOM3_ON_FANS
    CMP ROOM3_SCHED_OFF_FLAG, 1
    JE ROOM3_OFF_FANS
    JMP EXIT_ISR6

    ROOM3_ON_FANS:
    MOV AL, 03H
    OUT PORTN, AL
    MOV AL, 01H
    OUT PORTO, AL
    MOV FAN5_FLAG, 1
    MOV FAN6_FLAG, 1
    JMP EXIT_ISR6

    ROOM3_OFF_FANS:
    MOV AL, 00H
    OUT PORTN, AL
    MOV AL, 00H
    OUT PORTO, AL

```

```

MOV FAN5_FLAG, 0
MOV FAN6_FLAG, 0
JMP EXIT_ISR6

EXIT_ISR6:
POP DX
POP AX
POPF
IRET
ISR6 ENDP
PROCED6 ENDS

DATA SEGMENT
ORG 03000H
    ; LCD & Keypad PPI
    PORTA EQU 0D0H
    PORTB EQU 0D2H
    PORTC EQU 0D4H
    COM_REG1 EQU 0D6H

    ; ADC & Interrupt PPI
    PORTD EQU 0D8H
    PORTE EQU 0DAH
    PORTF EQU 0DCH
    COM_REG2 EQU 0DEH

    ; Timer & Buzzer PPI
    PORTG EQU 0E0H
    PORTH EQU 0E2H
    PORTI EQU 0E4H
    COM_REG3 EQU 0E6H

    ; Fans PPI
    PORTJ EQU 0F0H
    PORTK EQU 0F2H
    PORTL EQU 0F4H
    COM_REG4 EQU 0F6H
    PORTM EQU 0F8H
    PORTN EQU 0FAH
    PORTO EQU 0FCH
    COM_REG5 EQU 0FEH

    ; 8259 PIC
    PIC1 EQU 0C8H

```

```

PIC2 EQU 0CAH
ICW1 EQU 013H
ICW2 EQU 080H
ICW4 EQU 003H
OCW1 EQU 0C0H      ;1100 0000 = C0

; 8253 Timer
PORT_T EQU 0C0H
COM_REGT EQU 0C6H

; Strings to be displayed on LCD
MENU1_STR DB "Room 1  [1]", "$"
MENU2_STR DB "Room 2  [2]", "$"
MENU3_STR DB "Room 3  [3]", "$"
MENU4_STR DB "Schedule [4]", "$"
ROOM1_STR DB "Room 1  ", "$"
ROOM2_STR DB "Room 2  ", "$"
ROOM3_STR DB "Room 3  ", "$"
ROOM1_FAN1_STR DB "Fan 1: ", "$"
ROOM1_FAN2_STR DB "Fan 2: ", "$"
ROOM2_FAN1_STR DB "Fan 1: ", "$"
ROOM2_FAN2_STR DB "Fan 2: ", "$"
ROOM3_FAN1_STR DB "Fan 1: ", "$"
ROOM3_FAN2_STR DB "Fan 2: ", "$"
TURN_ON_STR DB "Turn ON: [1]", "$"
TURN_OFF_STR DB "Turn OFF: [2]", "$"
SCHED_STR DB "Schedule Room", "$"
FAN_ON_STR DB "[ON] ", "$"
FAN_OFF_STR DB "[OFF]", "$"
TEMP_STR DB "Temperature: ", "$"
WARNING_STR DB "Temperature Warning!", "$"
SCHED_TIME_STR DB "Time: ", "$"
CLEAR_BOTTOM DB "      ", "$"
CLEAR_TOP DB "      ", "$"

; Data Variables
HR_ONES_DIGIT DB 30H
HR_TENS_DIGIT DB 31H
MIN_ONES_DIGIT DB 30H
MIN_TENS_DIGIT DB 30H
AT_ROOM_FLAG DB 0
AT_ROOM1_FLAG DB 0
AT_ROOM2_FLAG DB 0
AT_ROOM3_FLAG DB 0

```

```

AT_SCHED_FLAG DB 0
AT_SCHED_ROOM1_FLAG DB 0
AT_SCHED_ROOM2_FLAG DB 0
AT_SCHED_ROOM3_FLAG DB 0
ADC_CURR DB 0
FAN1_FLAG DB 0
FAN2_FLAG DB 0
FAN3_FLAG DB 0
FAN4_FLAG DB 0
FAN5_FLAG DB 0
FAN6_FLAG DB 0
ROOM1_FANS_STATE DB 00H
ROOM2_FANS_STATE DB 00H
ROOM3_FANS_STATE DB 00H
ROOM1_WARNING_FLAG DB 0
ROOM2_WARNING_FLAG DB 0
ROOM3_WARNING_FLAG DB 0
HELPER DB 9AH
ROOM1_SCHED_HR_TENS DB ?
ROOM1_SCHED_HR_ONES DB ?
ROOM1_SCHED_MIN_TENS DB ?
ROOM1_SCHED_MIN_ONES DB ?
ROOM2_SCHED_HR_TENS DB ?
ROOM2_SCHED_HR_ONES DB ?
ROOM2_SCHED_MIN_TENS DB ?
ROOM2_SCHED_MIN_ONES DB ?
ROOM3_SCHED_HR_TENS DB ?
ROOM3_SCHED_HR_ONES DB ?
ROOM3_SCHED_MIN_TENS DB ?
ROOM3_SCHED_MIN_ONES DB ?
ROOM1_SCHED_ON_FLAG DB 0
ROOM1_SCHED_OFF_FLAG DB 0
ROOM2_SCHED_ON_FLAG DB 0
ROOM2_SCHED_OFF_FLAG DB 0
ROOM3_SCHED_ON_FLAG DB 0
ROOM3_SCHED_OFF_FLAG DB 0
T16 DB "16", "$"
T17 DB "17", "$"
T18 DB "18", "$"
T19 DB "19", "$"
T20 DB "20", "$"
T21 DB "21", "$"
T22 DB "22", "$"
T23 DB "23", "$"

```

```

T24 DB "24", "$"
T25 DB "25", "$"
T26 DB "26", "$"
T27 DB "27", "$"
T28 DB "28", "$"
T29 DB "29", "$"
T30 DB "30", "$"
DATA ENDS

STK SEGMENT STACK
    BOS DW 64d DUP (?)
    TOS LABEL WORD
STK ENDS

CODE SEGMENT PUBLIC 'CODE'
    ASSUME CS:CODE, DS:DATA, SS:STK
    ORG 08000H

START:
    MOV AX, DATA
    MOV DS, AX                ; set the Data Segment
address
    MOV AX, STK
    MOV SS, AX                ; set the Stack Segment
address
    LEA SP, TOS                ; set SP as Top of Stack
    CLI

    ; configure 8255 PPIs
    MOV DX, COM_REG1
    MOV AL, 10001001B
    OUT DX, AL
    MOV DX, COM_REG3
    MOV AL, 10000010B
    OUT DX, AL
    MOV AL, 10000000B
    MOV DX, COM_REG2
    OUT DX, AL
    MOV DX, COM_REG4
    OUT DX, AL
    MOV DX, COM_REG5
    OUT DX, AL

```

```

; configure 8283 Timer
    MOV AL, 00111000B
    OUT COM_REGT, AL

; configure 8259 PIC
    MOV AL, ICW1
    OUT PIC1, AL
    MOV AL, ICW2
    OUT PIC2, AL
    MOV AL, ICW4
    OUT PIC2, AL
    MOV AL, OCW1
    OUT PIC2, AL
    STI

; Storing interrupt vector to interrupt vector
table in memory
    MOV AX, OFFSET ISR1
    MOV [ES:200H], AX
    MOV AX, SEG ISR1
    MOV [ES:202H], AX
    MOV AX, OFFSET ISR2
    MOV [ES:204H], AX
    MOV AX, SEG ISR2
    MOV [ES:206H], AX
    MOV AX, OFFSET ISR3
    MOV [ES:208H], AX
    MOV AX, SEG ISR3
    MOV [ES:20AH], AX
    MOV AX, OFFSET ISR4
    MOV [ES:20CH], AX
    MOV AX, SEG ISR4
    MOV [ES:20EH], AX
    MOV AX, OFFSET ISR5
    MOV [ES:210H], AX
    MOV AX, SEG ISR5
    MOV [ES:212H], AX
    MOV AX, OFFSET ISR6
    MOV [ES:214H], AX
    MOV AX, SEG ISR6
    MOV [ES:216H], AX

```

HERE:

```

CALL INIT_LCD
CALL SHOW_MENU
CALL MENU_CHECK_DAVBL
JMP HERE

; MODULE: display menu
SHOW_MENU:
    MOV AL, 080H      ; set cursor position
    LEA SI, MENU1_STR ; move strng to display
    CALL DISPLAY_STR  ; instruct LCD and display
string from SI

    MOV AL, 0C0H
    LEA SI, MENU2_STR
    CALL DISPLAY_STR

    MOV AL, 094H
    LEA SI, MENU3_STR
    CALL DISPLAY_STR

    MOV AL, 0D4H
    LEA SI, MENU4_STR
    CALL DISPLAY_STR
RET

; MODULE: Check DAVBL for menu
MENU_CHECK_DAVBL:
    CALL CLOCK_TIME
    CALL CHECK_SCHEDULE
    MOV DX, PORTC
    IN AL, DX; read PORTC
    TEST AL, 10H ; check if DAVBL is high
    JZ MENU_CHECK_DAVBL ;if low then check again
    IN AL, DX ; read 4-bit keypad data
    AND AL, 0FH ;mask upper nibble
    CMP AL, 00H ;check if key pressed is 1 (00H)
    JE ROOM1 ;go to room 1 menu
    CMP AL, 01H ;check if key pressed is 2 (01H)
    JE ROOM2 ; go to room 2 menu
    CMP AL, 02H ;check if key pressed is 3 (02H)
    JE ROOM3 ; go to room 3 menu
    CMP AL, 04H ;check if key pressed is 4 (02H)
    JE SCHEDULE ; go to room all menu

```

```

CALL DELAY_1MS
JMP MENU_CHECK_DAVBL

; MODULE: Check DAVBL for rooms
ROOM_CHECK_DAVBL:
    CALL CLOCK_TIME
    CALL CHECK_SCHEDULE
    CMP ROOM1_WARNING_FLAG, 1
    JE ROOM1_WARNING
    CMP ROOM2_WARNING_FLAG, 1
    JE ROOM2_WARNING
    CMP ROOM3_WARNING_FLAG, 1
    JE ROOM3_WARNING
    CMP AT_ROOM_FLAG, 1
    JNE CONT_ROOM_CHECK_DAVBL
    CALL READ_ADC
    CALL ADC_DATA_CONVERTER
    MOV AL, 0E1H
    CALL DISPLAY_STR

CONT_ROOM_CHECK_DAVBL:
    MOV DX, PORTC
    IN AL, DX; read PORTC
    TEST AL, 10H ; check if DAVBL is high
    JZ ROOM_CHECK_DAVBL ;if low then check
again
    IN AL, DX ; read 4-bit keypad data
    AND AL, 0FH ; mask upper nibble
    CMP AL, 08H ; check if key pressed is 7
(08H)
    JE FIRST_FAN ; go to FIRST_FAN module
    CMP AL, 0AH ; check if key pressed is 9
(0AH)
    JE SECOND_FAN ; go to SECOND_FAN module
    CMP AL, 0EH ; check if key pressed is #
(0EH)
    JE BACK ; go back to menu

    CALL DELAY_1MS
    JMP ROOM_CHECK_DAVBL

; MODULE: Check davbl for scheduling
SCHEDULE_CHECK_DAVBL:
    CALL CLOCK_TIME

```

```

MOV DX, PORTC
IN AL, DX; read PORTC
TEST AL, 10H ; check if DAVBL is high
JZ SCHEDULE_CHECK_DAVBL ; if low then check
again
    IN AL, DX ; read 4-bit keypad data
    AND AL, 0FH ; mask upper nibble
    CMP AL, 00H ; check if key pressed is 1 (00H)
    JE SCHED_ROOM1
    CMP AL, 01H ; check if key pressed is 2 (01H)
    JE SCHED_ROOM2
    CMP AL, 02H ; check if key pressed is 3 (02H)
    JE SCHED_ROOM3
    CMP AL, 0EH ; check if key pressed is # (0EH)
    JE BACK ; go back to menu
    JMP SCHEDULE_CHECK_DAVBL

CHECK_DAVBL:
    CMP HELPER, 9CH
    JE DISP_COLON
    CMP HELPER, 9FH
    JE CONIRM_SCHEDULE
    CALL CLOCK_TIME
    MOV DX, PORTC
    IN AL, DX
    AND AL, 0FH
    CMP AL, 0CH ; check if key pressed
is *
    JE KEY_BACKSPACE
    CMP AL, 0EH ; check if key pressed
is #
    JE BACK
    CMP AL, 0DH ; check if key pressed
is 0
    JE KEY_D0
    CMP AL, 00H ; check if key pressed
is 1
    JE KEY_D1
    CMP AL, 01H ; check if key pressed
is 2
    JE KEY_D2
    CMP AL, 02H ; check if key pressed
is 3
    JE KEY_D3

```

```

is 4    CMP AL, 04H          ; check if key pressed
        JE KEY_D4
is 5    CMP AL, 05H          ; check if key pressed
        JE KEY_D5
is 6    CMP AL, 06H          ; check if key pressed
        JE KEY_D6
is 7    CMP AL, 08H          ; check if key pressed
        JE KEY_D7
is 8    CMP AL, 09H          ; check if key pressed
        JE KEY_D8
is 9    CMP AL, 0AH          ; check if key pressed
        JE KEY_D9
        JMP CHECK_DAVBL

        FINAL_CHECK_DAVBL:
        CALL CLOCK_TIME
        MOV DX, PORTC
        IN AL, DX
        AND AL, 0FH
        CMP AL, 0EH          ; check if key pressed
is #    JE BACK
        CMP AL, 00H          ; check if key pressed
is 1    JE TURN_ON
        CMP AL, 01H          ; check if key pressed
is 2    JE TURN_OFF
        JMP FINAL_CHECK_DAVBL

; helpers
        KEY_BACKSPACE:
        CMP HELPER, 9DH
        JE DOUBLE_BACKSPACE
        DEC HELPER
        JMP CONT3
        DOUBLE_BACKSPACE:
        DEC HELPER

```

```

        DEC HELPER
        CONT3:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, ''
        CALL DATA_CTRL
        CALL DELAY_500MS
        JMP CHECK_DAVBL
KEY_D0:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '0'
        JMP CONT2
KEY_D1:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '1'
        JMP CONT2
KEY_D2:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '2'
        JMP CONT2
KEY_D3:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '3'
        JMP CONT2
KEY_D4:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '4'
        JMP CONT2
KEY_D5:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '5'
        JMP CONT2
KEY_D6:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '6'
        JMP CONT2
KEY_D7:

```

```

        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '7'
        JMP CONT2
KEY_D8:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '8'
        JMP CONT2
KEY_D9:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, '9'
        JMP CONT2
DISP_COLON:
        MOV AL, HELPER
        CALL INST_CTRL
        MOV AL, 3AH
        JMP CONT2
CONT2:
        CALL DATA_CTRL
        CMP HELPER, 9AH
        JE STORE_SCHED_HR_TENS
        CMP HELPER, 9BH
        JE STORE_SCHED_HR_ONES
        CMP HELPER, 9DH
        JE STORE_SCHED_MIN_TENS
        CMP HELPER, 9EH
        JE STORE_SCHED_MIN_ONES

        CONT4:
        INC HELPER
        CALL DELAY_500MS
        JMP CHECK_DAVBL

STORE_SCHED_HR_TENS:
        CMP AT_SCHED_ROOM1_FLAG, 1
        JE STORE_ROOM11
        CMP AT_SCHED_ROOM2_FLAG, 1
        JE STORE_ROOM21
        CMP AT_SCHED_ROOM3_FLAG, 1
        JE STORE_ROOM31

STORE_ROOM11:

```

```

        MOV ROOM1_SCHED_HR_TENS, AL
JMP CONT4

STORE_ROOM21:
    MOV ROOM2_SCHED_HR_TENS, AL
JMP CONT4

STORE_ROOM31:
    MOV ROOM3_SCHED_HR_TENS, AL
JMP CONT4

STORE_SCHED_HR_ONES:
    CMP AT_SCHED_ROOM1_FLAG, 1
    JE STORE_ROOM12
    CMP AT_SCHED_ROOM2_FLAG, 1
    JE STORE_ROOM22
    CMP AT_SCHED_ROOM3_FLAG, 1
    JE STORE_ROOM32

STORE_ROOM12:
    MOV ROOM1_SCHED_HR_ONES, AL
JMP CONT4

STORE_ROOM22:
    MOV ROOM2_SCHED_HR_ONES, AL
JMP CONT4

STORE_ROOM32:
    MOV ROOM3_SCHED_HR_ONES, AL
JMP CONT4

STORE_SCHED_MIN_TENS:
    CMP AT_SCHED_ROOM1_FLAG, 1
    JE STORE_ROOM13
    CMP AT_SCHED_ROOM2_FLAG, 1
    JE STORE_ROOM23
    CMP AT_SCHED_ROOM3_FLAG, 1
    JE STORE_ROOM33

STORE_ROOM13:
    MOV ROOM1_SCHED_MIN_TENS, AL
JMP CONT4
STORE_ROOM23:
    MOV ROOM2_SCHED_MIN_TENS, AL

```

```

JMP CONT4

STORE_ROOM33:
    MOV ROOM3_SCHED_MIN_TENS, AL
JMP CONT4

STORE_SCHED_MIN_ONES:
    CMP AT_SCHED_ROOM1_FLAG, 1
    JE STORE_ROOM14
    CMP AT_SCHED_ROOM2_FLAG, 1
    JE STORE_ROOM24
    CMP AT_SCHED_ROOM3_FLAG, 1
    JE STORE_ROOM34

STORE_ROOM14:
    MOV ROOM1_SCHED_MIN_ONES, AL
JMP CONT4

STORE_ROOM24:
    MOV ROOM2_SCHED_MIN_ONES, AL
JMP CONT4

STORE_ROOM34:
    MOV ROOM3_SCHED_MIN_ONES, AL
JMP CONT4

; MODULE: Displays a 24-hour clock
CLOCK_TIME:
    CALL DISPLAY_HOUR_TENS
    CALL DISPLAY_HOUR_ONES
    CALL DISPLAY_COLON
    CALL DISPLAY_MIN_TENS
    CALL DISPLAY_MIN_ONES
    CALL UPDATE_TIME
RET
; MODULE: Update the time
UPDATE_TIME:
; 32 33 35 39
    CMP MIN_ONES_DIGIT, 3AH
    JE RESET_MIN_ONES
    CMP MIN_TENS_DIGIT, 36H
    JE RESET_MIN_TENS
    CMP HR_ONES_DIGIT, 3AH
    JE RESET_HR_ONES

```

```

    CMP HR_TENS_DIGIT, 33H
    JE RESET_HR_TENS
    CMP HR_ONES_DIGIT, 34H
    JE CHECK_IF_PM

CONT_UPDATE_TIME:
    INC MIN_ONES_DIGIT
    CALL DELAY_500MS
    RET

RESET_MIN_ONES:
    MOV MIN_ONES_DIGIT, 30H
    CALL DISPLAY_MIN_ONES
    INC MIN_TENS_DIGIT
    RET

RESET_MIN_TENS:
    MOV MIN_TENS_DIGIT, 30H
    CALL DISPLAY_MIN_TENS
    INC HR_ONES_DIGIT
    RET

RESET_HR_ONES:
    MOV HR_ONES_DIGIT, 30H
    CALL DISPLAY_HOUR_ONES
    INC HR_TENS_DIGIT
    RET

RESET_HR_TENS:
    MOV HR_TENS_DIGIT, 30H
    CALL DISPLAY_HOUR_TENS
    RET

CHECK_IF_PM:
    CMP HR_TENS_DIGIT, 32H
    JE RESET_TO_AM
    JNE CONT_UPDATE_TIME
    RET

RESET_TO_AM:
    MOV HR_ONES_DIGIT, 30H
    CALL DISPLAY_HOUR_ONES
    MOV HR_TENS_DIGIT, 30H
    CALL DISPLAY_HOUR_TENS

```

```

    RET
JMP UPDATE_TIME

; MODULE: Checks if time sync with schedule time
CHECK_SCHEDULE:
    MOV AL, HR_TENS_DIGIT
    CMP AL, ROOM1_SCHED_HR_TENS
    JE CHECK_HR_ONES
    CMP AL, ROOM2_SCHED_HR_TENS
    JE CHECK_HR_ONES
    CMP AL, ROOM3_SCHED_HR_TENS
    JE CHECK_HR_ONES
    RET

CHECK_HR_ONES:
    MOV AL, HR_ONES_DIGIT
    CMP AL, ROOM1_SCHED_HR_ONES
    JE CHECK_MIN_TENS
    CMP AL, ROOM2_SCHED_HR_ONES
    JE CHECK_MIN_TENS
    CMP AL, ROOM3_SCHED_HR_ONES
    JE CHECK_MIN_TENS
    RET

CHECK_MIN_TENS:
    MOV AL, MIN_TENS_DIGIT
    CMP AL, ROOM1_SCHED_MIN_TENS
    JE CHECK_MIN_ONES
    CMP AL, ROOM2_SCHED_MIN_TENS
    JE CHECK_MIN_ONES
    CMP AL, ROOM3_SCHED_MIN_TENS
    JE CHECK_MIN_ONES
    RET

CHECK_MIN_ONES:
    MOV AL, MIN_ONES_DIGIT
    CMP AL, ROOM1_SCHED_MIN_ONES
    JE SCHEDULE_ROOM1
    CMP AL, ROOM2_SCHED_MIN_ONES
    JE SCHEDULE_ROOM2
    CMP AL, ROOM3_SCHED_MIN_ONES
    JE SCHEDULE_ROOM3
    RET

```

```

SCHEDULE_ROOM1:
    MOV AL, 08H
    OUT PORTF, AL
    MOV AL, 00H
    OUT PORTF, AL
    RET

SCHEDULE_ROOM2:
    MOV AL, 10H
    OUT PORTF, AL
    MOV AL, 00H
    OUT PORTF, AL
    RET

SCHEDULE_ROOM3:
    MOV AL, 20H
    OUT PORTF, AL
    MOV AL, 00H
    OUT PORTF, AL
    RET

; MODULE: Menu, Rooms, Fans, Emergency, Scheduling
controls
ROOM1:
    MOV AT_ROOM_FLAG, 1
    MOV AT_ROOM1_FLAG, 1
    MOV AT_ROOM2_FLAG, 0
    MOV AT_ROOM3_FLAG, 0
    MOV AT_SCHED_FLAG, 0
    MOV DX, PORTE
    MOV AL, 01H
    OUT DX, AL
    CALL INIT_LCD
    MOV AL, 080H ; displays "Room 1"
    LEA SI, ROOM1_STR
    CALL DISPLAY_STR
    MOV AL, 0C0H ; displays "Fan 1: "
    LEA SI, ROOM1_FAN1_STR
    CALL DISPLAY_STR
    CMP FAN1_FLAG, 1
    JE FAN1_ON
    MOV AL, 0C7H ; displays "[OFF]"
    LEA SI, FAN_OFF_STR

```

```

CALL DISPLAY_STR

CONT_FAN2:
    MOV AL, 094H ; displays "Fan 2: "
    LEA SI, ROOM1_FAN2_STR
    CALL DISPLAY_STR
    CMP FAN2_FLAG, 1
    JE FAN2_ON
    MOV AL, 09BH ; displays "[OFF]"
    LEA SI, FAN_OFF_STR
    CALL DISPLAY_STR

CONT_ROOM1:
    JMP CONT

FAN1_ON:
    MOV AL, 0C7H ; displays "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_FAN2

FAN2_ON:
    MOV AL, 09BH ; displays "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_ROOM1

ROOM2:
    MOV AT_ROOM_FLAG, 1
    MOV AT_ROOM1_FLAG, 0
    MOV AT_ROOM2_FLAG, 1
    MOV AT_ROOM3_FLAG, 0
    MOV AT_SCHED_FLAG, 0
    MOV DX, PORTE
    MOV AL, 02H
    OUT DX, AL
    CALL INIT_LCD
    MOV AL, 080H ; displays "Room 2"
    LEA SI, ROOM2_STR
    CALL DISPLAY_STR
    MOV AL, 0C0H ; displays "Fan 1: "
    LEA SI, ROOM2_FAN1_STR
    CALL DISPLAY_STR
    CMP FAN3_FLAG, 1

```

```

JE FAN3_ON
MOV AL, 0C7H           ; displays "[OFF]"
LEA SI, FAN_OFF_STR
CALL DISPLAY_STR

CONT_FAN4:
MOV AL, 094H           ; displays "Fan 2: "
LEA SI, ROOM2_FAN2_STR
CALL DISPLAY_STR
CMP FAN4_FLAG, 1
JE FAN4_ON
MOV AL, 09BH           ; displays "[OFF]"
LEA SI, FAN_OFF_STR
CALL DISPLAY_STR

CONT_ROOM2:
JMP CONT

FAN3_ON:
MOV AL, 0C7H           ; displays "[ON] "
LEA SI, FAN_ON_STR
CALL DISPLAY_STR
JMP CONT_FAN4

FAN4_ON:
MOV AL, 09BH           ; displays "[ON] "
LEA SI, FAN_ON_STR
CALL DISPLAY_STR
JMP CONT_ROOM2

ROOM3:
MOV AT_ROOM_FLAG, 1
MOV AT_ROOM1_FLAG, 0
MOV AT_ROOM2_FLAG, 0
MOV AT_ROOM3_FLAG, 1
MOV AT_SCHED_FLAG, 0
MOV DX, PORTE
MOV AL, 03H
OUT DX, AL
CALL INIT_LCD
MOV AL, 080H           ; displays "Room 3"
LEA SI, ROOM3_STR
CALL DISPLAY_STR
MOV AL, 0C0H           ; displays "Fan 1: "

```

```

LEA SI, ROOM3_FAN1_STR
CALL DISPLAY_STR
CMP FAN5_FLAG, 1
JE FAN5_ON
MOV AL, 0C7H           ; displays "[OFF]"
LEA SI, FAN_OFF_STR
CALL DISPLAY_STR

CONT_FAN6:
MOV AL, 094H           ; displays "Fan 2: "
LEA SI, ROOM3_FAN2_STR
CALL DISPLAY_STR
CMP FAN6_FLAG, 1
JE FAN6_ON
MOV AL, 09BH           ; displays "[OFF]"
LEA SI, FAN_OFF_STR
CALL DISPLAY_STR

CONT_ROOM3:
JMP CONT

FAN5_ON:
MOV AL, 0C7H           ; displays "[ON] "
LEA SI, FAN_ON_STR
CALL DISPLAY_STR
JMP CONT_FAN6

FAN6_ON:
MOV AL, 09BH           ; displays "[ON] "
LEA SI, FAN_ON_STR
CALL DISPLAY_STR
JMP CONT_ROOM3

SCHEDULE:
MOV AT_ROOM_FLAG, 0
MOV AT_SCHED_FLAG, 1
MOV AT_ROOM1_FLAG, 0
MOV AT_ROOM2_FLAG, 0
MOV AT_ROOM3_FLAG, 0
CALL INIT_LCD
MOV AL, 080H
LEA SI, SCHED_STR
CALL DISPLAY_STR
MOV AL, 0C0H

```

```

LEA SI, MENU1_STR
CALL DISPLAY_STR
MOV AL, 094H
LEA SI, MENU2_STR
CALL DISPLAY_STR
MOV AL, 0D4H
LEA SI, MENU3_STR
CALL DISPLAY_STR
CALL DELAY_1MS
JMP SCHEDULE_CHECK_DAVBL

BACK:
MOV AT_ROOM_FLAG, 0
MOV AT_ROOM1_FLAG, 0
MOV AT_ROOM2_FLAG, 0
MOV AT_ROOM3_FLAG, 0
MOV AT_SCHED_FLAG, 0
MOV AL, 00H
OUT PORTE, AL
OUT PORTG, AL
JMP HERE

SCHED_ROOM1:
MOV AT_SCHED_ROOM1_FLAG, 1
MOV AL, 0C0H
CALL INST_CTRL
LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 094H
CALL INST_CTRL
LEA SI, SCHED_TIME_STR
CALL DISPLAY_STR
MOV AL, 09AH
CALL INST_CTRL
LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 9CH
CALL INST_CTRL
MOV AL, 3AH
CALL DATA_CTRL

```



```

CALL DELAY_1MS
JMP CHECK_DAVBL

SCHED_ROOM2:
MOV AT_SCHED_ROOM2_FLAG, 1
MOV AL, 0C0H
CALL INST_CTRL
LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 094H
CALL INST_CTRL
LEA SI, SCHED_TIME_STR
CALL DISPLAY_STR
MOV AL, 09AH
CALL INST_CTRL
LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 0D4H
CALL INST_CTRL
LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 9CH
CALL INST_CTRL
MOV AL, 3AH
CALL DATA_CTRL
CALL DELAY_1MS
JMP CHECK_DAVBL

SCHED_ROOM3:
MOV AT_SCHED_ROOM3_FLAG, 1
MOV AL, 0C0H
CALL INST_CTRL
LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 094H
CALL INST_CTRL
LEA SI, SCHED_TIME_STR
CALL DISPLAY_STR
MOV AL, 09AH
CALL INST_CTRL
LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 0D4H
CALL INST_CTRL

```

```

LEA SI, CLEAR_TOP
CALL DISPLAY_STR
MOV AL, 9CH
CALL INST_CTRL
MOV AL, 3AH
CALL DATA_CTRL
CALL DELAY_1MS
JMP CHECK_DAVBL

CONIRM_SCHEDULE:
MOV AL, 0C0H
CALL INST_CTRL
LEA SI, TURN_ON_STR
CALL DISPLAY_STR
MOV AL, 094H
CALL INST_CTRL
LEA SI, TURN_OFF_STR
CALL DISPLAY_STR
JMP FINAL_CHECK_DAVBL

TURN_ON:
CMP AT_SCHED_ROOM1_FLAG, 1
JE CONFIRM_ROOM1_ON_SCHED
CMP AT_SCHED_ROOM2_FLAG, 1
JE CONFIRM_ROOM2_ON_SCHED
CMP AT_SCHED_ROOM3_FLAG, 1
JE CONFIRM_ROOM3_ON_SCHED
JMP BACK

CONFIRM_ROOM1_ON_SCHED:
MOV ROOM1_SCHED_ON_FLAG, 1
MOV ROOM1_SCHED_OFF_FLAG, 0
MOV HELPER, 9AH
JMP BACK

CONFIRM_ROOM2_ON_SCHED:
MOV ROOM2_SCHED_ON_FLAG, 1
MOV ROOM2_SCHED_OFF_FLAG, 0
MOV HELPER, 9AH
JMP BACK

CONFIRM_ROOM3_ON_SCHED:
MOV ROOM3_SCHED_ON_FLAG, 1
MOV ROOM3_SCHED_OFF_FLAG, 0

```

```

MOV HELPER, 9AH
JMP BACK

TURN_OFF:
CMP AT_SCHED_ROOM1_FLAG, 1
JE CONFIRM_ROOM1_OFF_SCHED
CMP AT_SCHED_ROOM2_FLAG, 1
JE CONFIRM_ROOM2_OFF_SCHED
CMP AT_SCHED_ROOM3_FLAG, 1
JE CONFIRM_ROOM3_OFF_SCHED
JMP BACK

CONFIRM_ROOM1_OFF_SCHED:
MOV ROOM1_SCHED_ON_FLAG, 0
MOV ROOM1_SCHED_OFF_FLAG, 1
MOV HELPER, 9AH
JMP BACK

CONFIRM_ROOM2_OFF_SCHED:
MOV ROOM2_SCHED_ON_FLAG, 0
MOV ROOM2_SCHED_OFF_FLAG, 1
MOV HELPER, 9AH
JMP BACK

CONFIRM_ROOM3_OFF_SCHED:
MOV ROOM3_SCHED_ON_FLAG, 0
MOV ROOM3_SCHED_OFF_FLAG, 1
MOV HELPER, 9AH
JMP BACK

FIRST_FAN:
CMP AT_ROOM1_FLAG, 1
JE AT_ROOM1_1
CMP AT_ROOM2_FLAG, 1
JE AT_ROOM2_1
CMP AT_ROOM3_FLAG, 1
JE AT_ROOM3_1
CMP AT_SCHED_FLAG, 1
JE AT_ROOMS
JMP CONT

AT_ROOM1_1:
CMP FAN1_FLAG, 1
JE RESET_FAN1_FLAG

```

```

    CMP FAN2_FLAG, 1
    JE FANS_ROOM1_1
    MOV FAN1_FLAG, 1
    MOV AL, 01H
    OUT PORTJ, AL
    MOV AL, 0C7H                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM1_1

FANS_ROOM1_1:
    MOV FAN1_FLAG, 1
    MOV FAN2_FLAG, 1
    MOV AL, 03H
    OUT PORTJ, AL
    MOV AL, 0C7H                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM1_1

RESET_FAN1_FLAG:
    MOV FAN1_FLAG, 0
    MOV AL, 00H
    OUT PORTJ, AL
    MOV AL, 0C7H                ; Update LCD to
display "[OFF]"
    LEA SI, FAN_OFF_STR
    CALL DISPLAY_STR
    CMP FAN2_FLAG, 1
    JE OFF_FAN1_ONLY
    JMP CONT_AT_ROOM1_1

OFF_FAN1_ONLY:
    MOV AL, 02H
    OUT PORTJ, AL

CONT_AT_ROOM1_1:
    JMP CONT

```

```

AT_ROOM2_1:
    CMP FAN3_FLAG, 1
    JE RESET_FAN3_FLAG
    CMP FAN4_FLAG, 1
    JE FANS_ROOM2_1
    MOV FAN3_FLAG, 1
    MOV AL, 01H
    OUT PORTL, AL
    MOV AL, 0C7H                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM2_1

FANS_ROOM2_1:
    MOV FAN3_FLAG, 1
    MOV FAN4_FLAG, 1
    MOV AL, 03H
    OUT PORTL, AL
    MOV AL, 0C7H                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM2_1

RESET_FAN3_FLAG:
    MOV FAN3_FLAG, 0
    MOV AL, 00H
    OUT PORTL, AL
    MOV AL, 0C7H                ; Update LCD to
display "[OFF]"
    LEA SI, FAN_OFF_STR
    CALL DISPLAY_STR
    CMP FAN4_FLAG, 1
    JE OFF_FAN3_ONLY
    JMP CONT_AT_ROOM2_1

OFF_FAN3_ONLY:
    MOV AL, 02H

```

```

    OUT PORTL, AL

CONT_AT_ROOM2_1:
    JMP CONT

AT_ROOM3_1:
    CMP FAN5_FLAG, 1
    JE RESET_FAN5_FLAG
    CMP FAN6_FLAG, 1
    JE FANS_ROOM3_1
    MOV FAN5_FLAG, 1
    MOV AL, 01H
    OUT PORTN, AL
    MOV AL, 0C7H                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM3_1

FANS_ROOM3_1:
    MOV FAN5_FLAG, 1
    MOV FAN6_FLAG, 1
    MOV AL, 03H
    OUT PORTN, AL
    MOV AL, 0C7H                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; Update LCD to
display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM3_1

RESET_FAN5_FLAG:
    MOV FAN5_FLAG, 0
    MOV AL, 00H
    OUT PORTN, AL
    MOV AL, 0C7H                ; Update
LCD to display "[OFF]"
    LEA SI, FAN_OFF_STR
    CALL DISPLAY_STR
    CMP FAN6_FLAG, 1
    JE OFF_FAN5_ONLY

```

```

        JMP CONT_AT_ROOM3_1

OFF_FAN5_ONLY:
    MOV AL, 02H
    OUT PORTN, AL

CONT_AT_ROOM3_1:
    JMP CONT

AT_ROOMS:
    MOV DX, PORTJ
    MOV AL, 03H
    OUT DX, AL
    MOV AL, 0C7H                ; Update LCD
to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; Update LCD
to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV DX, PORTL
    MOV AL, 03H
    OUT DX, AL
    MOV AL, 0C7H                ; Update LCD
to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV DX, PORTN
    MOV AL, 03H
    OUT DX, AL
    MOV AL, 0C7H                ; Update LCD
to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; Update LCD
to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT

```

```

SECOND_FAN:
    CMP AT_ROOM1_FLAG, 1
    JE AT_ROOM1_2
    CMP AT_ROOM2_FLAG, 1
    JE AT_ROOM2_2
    CMP AT_ROOM3_FLAG, 1
    JE AT_ROOM3_2

AT_ROOM1_2:
    CMP FAN2_FLAG, 1
    JE RESET_FAN2_FLAG
    CMP FAN1_FLAG, 1
    JE FANS_ROOM1_2
    MOV FAN2_FLAG, 1
    MOV AL, 02H
    OUT PORTJ, AL
    MOV AL, 09BH                ; Update LCD
to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM1_2

FANS_ROOM1_2:
    MOV FAN1_FLAG, 1
    MOV FAN2_FLAG, 1
    MOV AL, 03H
    OUT PORTJ, AL
    MOV AL, 0C7H                ; Update
LCD to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; Update
LCD to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM1_2

RESET_FAN2_FLAG:
    MOV FAN2_FLAG, 0
    MOV AL, 00H
    OUT PORTJ, AL
    MOV AL, 09BH                ; Update
LCD to display "[OFF]"

```

```

    LEA SI, FAN_OFF_STR
    CALL DISPLAY_STR
    CMP FAN1_FLAG, 1
    JE OFF_FAN2_ONLY
    JMP CONT_AT_ROOM1_2

OFF_FAN2_ONLY:
    MOV AL, 01H
    OUT PORTJ, AL

CONT_AT_ROOM1_2:
    JMP CONT

AT_ROOM2_2:
    CMP FAN4_FLAG, 1
    JE RESET_FAN4_FLAG
    CMP FAN3_FLAG, 1
    JE FANS_ROOM2_2
    MOV FAN4_FLAG, 1
    MOV DX, PORTL
    MOV AL, 02H
    OUT DX, AL
    MOV AL, 09BH                ; Update LCD
to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM2_2

FANS_ROOM2_2:
    MOV FAN3_FLAG, 1
    MOV FAN4_FLAG, 1
    MOV AL, 03H
    OUT PORTL, AL
    MOV AL, 0C7H                ; Update
LCD to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; Update
LCD to display "[ON] "
    LEA SI, FAN_ON_STR
    CALL DISPLAY_STR
    JMP CONT_AT_ROOM2_2

RESET_FAN4_FLAG:

```

```

        MOV FAN4_FLAG, 0
        MOV DX, PORTL
        MOV AL, 00H
        OUT DX, AL
        MOV AL, 09BH
        LCD to display "[OFF]"
        LEA SI, FAN_OFF_STR
        CALL DISPLAY_STR
        CMP FAN3_FLAG, 1
        JE OFF_FAN4_ONLY
        JMP CONT_AT_ROOM2_2

        OFF_FAN4_ONLY:
        MOV AL, 01H
        OUT PORTL, AL

        CONT_AT_ROOM2_2:
        JMP CONT

        AT_ROOM3_2:
        CMP FAN6_FLAG, 1
        JE RESET_FAN6_FLAG
        CMP FAN5_FLAG, 1
        JE FANS_ROOM3_2
        MOV FAN6_FLAG, 1
        MOV AL, 02H
        OUT PORTN, AL
        MOV AL, 09BH
        to display "[ON] "
        LEA SI, FAN_ON_STR
        CALL DISPLAY_STR
        JMP CONT_AT_ROOM3_2

        FANS_ROOM3_2:
        MOV FAN5_FLAG, 1
        MOV FAN6_FLAG, 1
        MOV AL, 03H
        OUT PORTN, AL
        MOV AL, 0C7H
        LCD to display "[ON] "
        LEA SI, FAN_ON_STR
        CALL DISPLAY_STR
        MOV AL, 09BH
        LCD to display "[ON] "

```

```

        LEA SI, FAN_ON_STR
        CALL DISPLAY_STR
        JMP CONT_AT_ROOM3_2

        RESET_FAN6_FLAG:
        MOV FAN6_FLAG, 0
        MOV AL, 00H
        OUT PORTN, AL
        MOV AL, 09BH
        LCD to display "[OFF]"
        LEA SI, FAN_OFF_STR
        CALL DISPLAY_STR
        CMP FAN5_FLAG, 1
        JE OFF_FAN6_ONLY
        JMP CONT_AT_ROOM3_2

        OFF_FAN6_ONLY:
        MOV AL, 01H
        OUT PORTN, AL

        CONT_AT_ROOM3_2:
        JMP CONT

        ROOM1_WARNING:
        MOV AL, 01H
        OUT PORTF, AL
        MOV AL, 00H
        OUT PORTF, AL
        JMP CONT

        ROOM2_WARNING:
        MOV AL, 02H
        OUT PORTF, AL
        MOV AL, 00H
        OUT PORTF, AL
        JMP CONT

        ROOM3_WARNING:
        MOV AL, 04H
        OUT PORTF, AL
        MOV AL, 00H
        OUT PORTF, AL
        JMP CONT

        CONT:
        CALL DELAY_1MS

```

```

        JMP ROOM_CHECK_DAVBL

; MODULE: fetch data from temperature sensor using ADC
; ADDC ADDB ADDA
; 0 0 0 = TEMPSEN_1
; 0 0 1 = TEMPSEN_2
; 0 1 0 = TEMPSEN_3
        READ_ADC:
        MOV DX, PORTD ; select ADC out port
        IN AL, DX ; store digital data to AL
        MOV ADC_CURR, AL
        RET

; MODULE: Endless loop
        ENDLESS:
        JMP ENDLESS

        INST_CTRL:
        PUSH AX ; preserve value of AL
        MOV DX, PORTA ; set port of LCD data bus
        (PORTA)
        OUT DX, AL ; write data in AL to PORTA
        MOV DX, PORTB ; set port of LCD control lines
        (PORTB)
        MOV AL, 02H ; E=1, RS=0 (access instruction
        reg)
        OUT DX, AL ; write data in AL to PORTB
        CALL DELAY_1MS ; delay for 1 ms
        MOV DX, PORTB ; set port of LCD control lines
        (PORTB)
        MOV AL, 00H ; E=0, RS=0
        OUT DX, AL ; write data in AL to PORTB
        POP AX ; restore value of AL
        RET

        DATA_CTRL:
        PUSH AX ; preserve value of AL
        MOV DX, PORTA ; set port of LCD data bus
        (PORTA)
        OUT DX, AL ; write data in AL to PORTA
        MOV DX, PORTB ; set port of LCD control lines
        (PORTB)

```

```

        MOV AL, 03H ; E=1, RS=1 (access data register)
        OUT DX, AL ; write data in AL to PORTB
        CALL DELAY_1MS ; delay for 1 ms
        MOV DX, PORTB ; set port of LCD control lines
(PORTB)
        MOV AL, 01H ; E=0, RS=1
        OUT DX, AL ; write data in AL to PORTB
        POP AX ; restore value of AL
        RET

INIT_LCD:
        MOV AL, 38H ; 8-bit interface, dual-line
display
        CALL INST_CTRL ; write instruction to LCD
        MOV AL, 08H ; display off, cursor off, blink
off
        CALL INST_CTRL ; write instruction to LCD
        MOV AL, 01H ; clear display
        CALL INST_CTRL ; write instruction to LCD
        MOV AL, 06H ; increment cursor, display shift
off
        CALL INST_CTRL ; write instruction to LCD
        MOV AL, 0CH ; display on, cursor off, blink
off
        CALL INST_CTRL ; write instruction to LCD
        RET

; MODULE: Convert the digital data from ADC to a
string format and convert to a certain speed
; 16          21          26
; 17          22          27
; 18    Speed 3    23    Speed 2    28    Speed 1
; 19          24          29
; 20          25          30
ADC_DATA_CONVERTER:
        MOV AL, ADC_CURR
        CMP AL, 01BH
        JL BUZZER
        CMP AL, 033H
        JG BUZZER
        MOV AL, 00H
        OUT PORTG, AL

```

```

        MOV AL, 0D4H ; displays
"Temperature: "
        LEA SI, TEMP_STR
        CALL DISPLAY_STR
        MOV AL, 0E3H
        LEA SI, CLEAR_BOTTOM
        CALL DISPLAY_STR

        MOV AL, ADC_CURR
        CMP AL, 01BH
        JE TEMP_16
        CMP AL, 01DH
        JE TEMP_17
        CMP AL, 01FH
        JE TEMP_18
        CMP AL, 021H
        JE TEMP_19
        CMP AL, 022H
        JE TEMP_20
        CMP AL, 024H
        JE TEMP_21
        CMP AL, 026H
        JE TEMP_22
        CMP AL, 027H
        JE TEMP_23
        CMP AL, 029H
        JE TEMP_24
        CMP AL, 02BH
        JE TEMP_25
        CMP AL, 02CH
        JE TEMP_26
        CMP AL, 02EH
        JE TEMP_27
        CMP AL, 030H
        JE TEMP_28
        CMP AL, 032H
        JE TEMP_29
        CMP AL, 033H
        JE TEMP_30
        RET

BUZZER:
        CMP FAN1_FLAG, 1
        JE BUZZER_ON1

```

```

        CMP FAN2_FLAG, 1
        JE BUZZER_ON1
        CMP FAN3_FLAG, 1
        JE BUZZER_ON2
        CMP FAN4_FLAG, 1
        JE BUZZER_ON2
        CMP FAN5_FLAG, 1
        JE BUZZER_ON3
        CMP FAN6_FLAG, 1
        JE BUZZER_ON3
        RET
BUZZER_ON1:
        MOV AL, 0D4H
        LEA SI, WARNING_STR
        CALL DISPLAY_STR
        MOV CX, 03H
        CALL TIMER_CTRL
        MOV ROOM1_WARNING_FLAG, 1
        MOV AL, 02H
        OUT PORTG, AL
        MOV AL, 0C7H ; displays
"[OFF]"
        LEA SI, FAN_OFF_STR
        CALL DISPLAY_STR
        MOV AL, 09BH ; displays
"[OFF]"
        LEA SI, FAN_OFF_STR
        CALL DISPLAY_STR
        RET
BUZZER_ON2:
        MOV AL, 0D4H
        LEA SI, WARNING_STR
        CALL DISPLAY_STR
        MOV CX, 03H
        CALL TIMER_CTRL
        MOV ROOM2_WARNING_FLAG, 1
        MOV AL, 02H
        OUT PORTG, AL
        MOV AL, 0C7H ; displays "[OFF]"
        LEA SI, FAN_OFF_STR
        CALL DISPLAY_STR
        MOV AL, 09BH ; displays "[OFF]"
        LEA SI, FAN_OFF_STR
        CALL DISPLAY_STR

```

```

RET
BUZZER_ON3:
    MOV AL, 0D4H
    LEA SI, WARNING_STR
    CALL DISPLAY_STR
    MOV CX, 03H
    CALL TIMER_CTRL
    MOV ROOM3_WARNING_FLAG, 1
    MOV AL, 02H
    OUT PORTG, AL
    MOV AL, 0C7H                ; displays
    "[OFF]"
    LEA SI, FAN_OFF_STR
    CALL DISPLAY_STR
    MOV AL, 09BH                ; displays
    "[OFF]"
    LEA SI, FAN_OFF_STR
    CALL DISPLAY_STR
RET

HANDLE_ROOM:
    CMP AT_ROOM1_FLAG, 1
    JE ROOM01
    CMP AT_ROOM2_FLAG, 1
    JE ROOM02
    CMP AT_ROOM3_FLAG, 1
    JE ROOM03
    CMP AT_SCHED_FLAG, 1
    JE ROOMS
    JMP CONT1

ROOM01:
    MOV DX, PORTK
    JMP CONT1
ROOM02:
    MOV DX, PORTM
    JMP CONT1
ROOM03:
    MOV DX, PORTO
    JMP CONT1
ROOMS:

CONT1:
RET

```

```

; Getting Temperature sensor value ready to
display
TEMP_16:
    CALL HANDLE_ROOM
    MOV AL, 03H
    OUT DX, AL
    LEA SI, T16
RET
TEMP_17:
    CALL HANDLE_ROOM
    MOV AL, 03H
    OUT DX, AL
    LEA SI, T17
RET
TEMP_18:
    CALL HANDLE_ROOM
    MOV AL, 03H
    OUT DX, AL
    LEA SI, T18
RET
TEMP_19:
    CALL HANDLE_ROOM
    MOV AL, 03H
    OUT DX, AL
    LEA SI, T19
RET
TEMP_20:
    CALL HANDLE_ROOM
    MOV AL, 03H
    OUT DX, AL
    LEA SI, T20
RET
TEMP_21:
    CALL HANDLE_ROOM
    MOV AL, 02H
    OUT DX, AL
    LEA SI, T21
RET
TEMP_22:
    CALL HANDLE_ROOM
    MOV AL, 02H
    OUT DX, AL
    LEA SI, T22

```

```

RET
TEMP_23:
    CALL HANDLE_ROOM
    MOV AL, 02H
    OUT DX, AL
    LEA SI, T23
RET
TEMP_24:
    CALL HANDLE_ROOM
    MOV AL, 02H
    OUT DX, AL
    LEA SI, T24
RET
TEMP_25:
    CALL HANDLE_ROOM
    MOV AL, 02H
    OUT DX, AL
    LEA SI, T25
RET
TEMP_26:
    CALL HANDLE_ROOM
    MOV AL, 01H
    OUT DX, AL
    LEA SI, T26
RET
TEMP_27:
    CALL HANDLE_ROOM
    MOV AL, 01H
    OUT DX, AL
    LEA SI, T27
RET
TEMP_28:
    CALL HANDLE_ROOM
    MOV AL, 01H
    OUT DX, AL
    LEA SI, T28
RET
TEMP_29:
    CALL HANDLE_ROOM
    MOV AL, 01H
    OUT DX, AL
    LEA SI, T29
RET
TEMP_30:

```

```

        CALL HANDLE_ROOM
        MOV AL, 01H
        OUT DX, AL
        LEA SI, T30
    RET

; MODULE: Displays a string from SI
DISPLAY_STR:
CALL INST_CTRL

DISP:
    MOV AL, [SI]
    CMP AL, '$'
    JE EXIT
    CALL DATA_CTRL
    INC SI
    JMP DISP
RET

DISPLAY_HOUR_TENS:
    MOV AL, 08FH
    CALL INST_CTRL
    MOV AL, HR_TENS_DIGIT
    CALL DATA_CTRL
RET

DISPLAY_HOUR_ONES:
    MOV AL, 090H
    CALL INST_CTRL
    MOV AL, HR_ONES_DIGIT
    CALL DATA_CTRL
RET

DISPLAY_COLON:
    MOV AL, 091H
    CALL INST_CTRL
    MOV AL, 3AH
    CALL DATA_CTRL
RET

DISPLAY_MIN_TENS:
    MOV AL, 092H
    CALL INST_CTRL
    MOV AL, MIN_TENS_DIGIT

```

```

        CALL DATA_CTRL
    RET

DISPLAY_MIN_ONES:
    MOV AL, 093H
    CALL INST_CTRL
    MOV AL, MIN_ONES_DIGIT
    CALL DATA_CTRL
    RET

; MODULE: Timer Control
TIMER_CTRL:
    CALL DELAY_1S
    DEC CX
    CMP CX, 00H
    JNZ TIMER_CTRL
    RET

DELAY_1S:
    MOV DX, PORT_T    ; access 8253 timer
    MOV AL, 0A0H
    OUT DX, AL
    MOV AL, 0FH
    OUT DX, AL
    LOCK_INPUT:
        MOV DX, PORTH
        IN AX, DX
        XOR AH, AH
        AND AL, 01H
        CMP AL, 00H    ; checks if remaining time
is 0
        JNE LOCK_INPUT
    RET

DELAY_500MS:  MOV CX, 25
L2:
    CALL DELAY_1MS
    LOOP L2
    RET

DELAY_1MS:  MOV BX, 02CAH
L1:
    DEC BX
    NOP

```

```

        JNZ L1
    RET

    RET

EXIT:
    RET
CODE ENDS
END START

```