

ภาคผนวก

โปรแกรมระบบตรวจสอบการเข้าใช้ห้องคอมพิวเตอร์

KPAD_ROW0	BIT	P2.0	
KPAD_ROW1	BIT	P2.1	
KPAD_ROW2	BIT	P2.2	
KPAD_ROW3	BIT	P2.3	
KPAD_COL2	BIT	P2.4	
KPAD_COL1	BIT	P2.5	
KPAD_COL0	BIT	P2.6	
FLAG	EQU	02FH	
KEYPRESSED	BIT	FLAG.0	
KPAD_DATA	EQU	032H	
LCD_EN	BIT	P3.6	
LCD_RS	BIT	P3.7	
LCD_ADD	EQU	030H	
LCD_DATA	EQU	031H	
INTERUPT	EQU	032H	
CHS	EQU	050H	
RXDINPUT	EQU	051H	
TXDS	EQU	052H	;TXD
RXDS	EQU	053H	;READ
SENSOR	EQU	54H	
INPUT	EQU	55H	;inputkeypad
;-----			
; Main Program.			
;-----			
ORG 0000H			

```
TTT:      MOV P0,#00H
          MOV P3,#0FFH
          MOV P1,#0FFH
          MOV  P2,#0FFH
          MOV IE,#00B
          CLR F0
          CLR P1.7
          MOV INTERRUPT,#80H
          CLR P1.4
          CLR P1.5
          SETB P3.4
          ;---- set port 232
          MOV IE,#00H
          MOV TMOD,#021H
          MOV TL1,#0FDH
          MOV TH1,#0FDH
          SETB TR1
          MOV SCON,#050H
          ;---- LCD
          CLR LCD_EN
          CLR LCD_RS
          ACALL INIT_LCD
MAIN:     SETB P3.2
          SETB P3.3
NNX:     MOV LCD_ADD,#00H
          ACALL SET_ADD
          MOV DPTR,#TABLE
          ACALL WRITE_LCD
          MOV TXDS,#'R'
          ACALL TXD
KEY_START: ACALL KEY
```

```
CJNE A,#'X',MAIN
MOV LCD_ADD,#00H
ACALL SET_ADD
MOV DPTR,#TABLE_BAR
ACALL WRITE_LCD

RS232_RXD:    JNB RI,$
               MOV A,SBUF
               MOV RXDS,A
               CLR RI
               CJNE A,#'C',CH

LOOP:         MOV LCD_ADD,#00H
               ACALL SET_ADD
               MOV DPTR,#TABLE_B
               LCALL WRITE_LCD
               MOV CHS,#0

START:        ACALL KEY
               MOV LCD_ADD,#00H
               ACALL SET_ADD
               ACALL POSION
               ACALL WRITE_LCD
               INC CHS
               MOV R1,CHS
               CJNE R1,#4,START
               MOV CHS,#0
               JNB RI,$
               MOV A,SBUF
               CLR RI

CH:           CJNE A,#'A',CHOOSE
```

```
SETB P1.4
SETB P1.0
ACALL SOUND
ACALL DELAY_100ms
ACALL SOUND
MOV LCD_ADD,#00H
ACALL SET_ADD
MOV DPTR,#TABLE_5
ACALL WRITE_LCD
ACALL DELAY_1s
ACALL DELAY_1s
ACALL DELAY_1s
ACALL DELAY_1s
JB P1.0,$
ACALL SOUND
ACALL SOUND
ACALL DELAY_1s
CLR P1.4 ;CLOSE
ACALL DELAY_100ms
MOV LCD_ADD,#00H
ACALL SET_ADD
MOV DPTR,#TABLE_6
ACALL WRITE_LCD
ACALL DELAY_1s
ACALL DELAY_1s
AJMP MAIN
CHOOSE: CJNE A,#'Z',CHOOSE_1
MOV LCD_ADD,#00H
ACALL SET_ADD
MOV DPTR,#TABLE_ERROR
ACALL WRITE_LCD
ACALL DELAY_1s
```

```

        ACALL DELAY_1s
        AJMP MAIN

CHOOSE_1:    CJNE A,#'B',CHOOSE_2
              MOV LCD_ADD,#00H
              ACALL SET_ADD
              MOV DPTR,#TABLE_F
              ACALL WRITE_LCD
              ACALL DELAY_1s
              AJMP MAIN

CHOOSE_2:    AJMP MAIN

POSITION:    MOV A,CHS

PS_1:        CJNE A,#0,PS_2
              MOV DPTR,#TABLE_1
              RET

PS_2:        CJNE A,#1,PS_3
              MOV DPTR,#TABLE_2
              RET

PS_3:        CJNE A,#2,PS_4
              MOV DPTR,#TABLE_3
              RET

PS_4:        MOV DPTR,#TABLE_4
              RET

;-----
;----- keypad

KEY:         JNB P3.2,STUDENT
              JNB P3.3,TEACHER
              ACALL GET_KPAD
              MOV A,KPAD_DATA
              JZ NEXT
              JB KEYPRESSED,SHOW

```

```

SETB  KEYPRESSED
AJMP  KEY
NEXT: CLR  KEYPRESSED
AJMP  KEY
SHOW: MOV DPTR,#TABLE_CODE
MOVC A,@A+DPTR
ACALL SOUND
ACALL DELAY_400ms
MOV  SBUF,A
JNB  TI,$
CLR  TI
MOV  P2,#0FFH
RET
STUDENT:
ACALL SOUND
SETB P3.2
ACALL DELAY_1s
SETB P1.5
MOV  CHS,#0
MOV  TXDS,#'O'
ACALL TXD
OUT_1: ACALL KEY
MOV  R0,CHS
INC  CHS
CJNE R0,#7,OUT_1
;-----
JNB  RI,$
MOV  A,SBUF
MOV  RXDS,A      ;VALUE RXD
CLR  RI
ACALL ICH
ACALL DELAY_1s

```

```

ACALL DELAY_1s

CLR P1.5
AJMP MAIN

TEACHER:
SETB P3.3
ACALL DELAY_1s
SETB P1.5
MOV CHS,#00H
MOV TXDS,#'T'
ACALL TXD

TEA:
ACALL KEY
MOV R0,CHS
INC CHS
CJNE R0,#3,TEA
JNB RI,$
MOV A,SBUF
CLR RI
ACALL ICH
ACALL DELAY_1s
ACALL DELAY_1s
CLR P1.5
AJMP MAIN

;----- keypad
; Keypad Scan key Subroutine
;-----
GET_KPAD:      MOV P2,#0FFH
               MOV KPAD_DATA,#0

CHK_COLO:     CLR KPAD_COLO
               MOV      A,P2
               ANL      A,#00FH
               CJNE  A,#00FH,COLO_DETECT

```



```
                                AJMP  CHK_COL1

COL0_DETECT:                   MOV   KPAD_DATA,#01H
                                AJMP  GET_ROW

CHK_COL1:                      SETB  KPAD_COL0
                                CLR    KPAD_COL1
                                MOV    A,P2
                                ANL    A,#00FH
                                CJNE  A,#00FH,COL1_DETECT
                                AJMP  CHK_COL2

COL1_DETECT:                   MOV   KPAD_DATA,#02
                                AJMP  GET_ROW

CHK_COL2:                      SETB  KPAD_COL1
                                CLR    KPAD_COL2
                                MOV    A,P2
                                ANL    A,#00FH
                                CJNE  A,#00FH,COL2_DETECT
                                RET

COL2_DETECT:MOV   KPAD_DATA,#03

GET_ROW:                       CLR    KPAD_COL0
                                CLR    KPAD_COL1
                                CLR    KPAD_COL2

                                JB     KPAD_ROW0,CHK_ROW1
                                RET

CHK_ROW1:                      JB    KPAD_ROW1,CHK_ROW2
                                MOV    A,KPAD_DATA
                                ADD    A,#3
                                MOV    KPAD_DATA,A
                                RET
```

```

CHK_ROW2:      JB    KPAD_ROW2,CHK_ROW3
                MOV   A,KPAD_DATA
                ADD   A,#6
                MOV   KPAD_DATA,A
                RET

```

```

CHK_ROW3:      MOV   A,KPAD_DATA
                ADD   A,#9
                MOV   KPAD_DATA,A
                RET

```

```

;----- INPUT TXD

```

```

TXD:           MOV   SBUF,TXDS
                JNB   TI,$
                CLR   TI
                RET

```

```

;----- LCD

```

```

INIT_LCD:      CLR   LCD_RS
                MOV   P0,#00111000B
                ACALL LCD_CLK
                ACALL DELAY_10ms
                MOV   P0,#00111000B
                ACALL LCD_CLK
                ACALL LCD_OFF
                ACALL LCD_CLR
                MOV   P0,#00000110B
                ACALL LCD_CLK
                ACALL LCD_HOME

```

```

LCD_CLR:       CLR   LCD_RS
                MOV   P0,#01H
                ACALL LCD_CLK
                RET

```

```
LCD_OFF:      CLR LCD_RS
              MOV P0,#08H
              ACALL LCD_CLK
              RET

LCD_CLK:      SETB LCD_EN
              ACALL DELAY_LCD
              CLR LCD_EN
              ACALL DELAY_LCD
              RET

LCD_ON:       CLR LCD_RS
              MOV P0,#0CH
              ACALL LCD_CLK
              RET

SET_ADD:     CLR LCD_RS
              MOV A,LCD_ADD
              SETB ACC.7
              MOV P0,A
              ACALL LCD_CLK
              RET

LCD_HOME:    CLR LCD_RS
              MOV P0,#00000010B
              ACALL LCD_CLK
              RET

WRITE_LCD:   MOV R0,#0
W:           CLR A
              MOVC A,@A+DPTR
              MOV P0,A
              ACALL LCD_CLK
              INC R0
              INC DPTR
              CJNE R0,#8,W
```

```
ACALL LCD_ON
RET
ICH: CJNE A,#A',ICHOOSE
      SETB P1.4 ; OPEN
      ACALL SOUND
      ACALL SOUND
      SETB P1.0
      MOV LCD_ADD,#00H
      ACALL SET_ADD
      MOV DPTR,#TABLE_5
      ACALL WRITE_LCD
      ACALL DELAY_1s
      SETB P1.5
      ACALL DELAY_1s
      ACALL DELAY_1s
      ACALL DELAY_1s
      CA: JB P1.0,CA
      ACALL SOUND
      ACALL SOUND
      ACALL DELAY_1s
      CLR P1.5
      CLR P1.4
      ACALL DELAY_100ms
      MOV LCD_ADD,#00H
      ACALL SET_ADD
      MOV DPTR,#TABLE_6
      ACALL WRITE_LCD
      ACALL DELAY_1s
      MOV LCD_ADD,#00H
      ACALL SET_ADD
      MOV DPTR,#TABLE
      ACALL WRITE_LCD
```

```

                RET
ICHOOSE:       CJNE A,#'Z',ICHOOSE_1
                ACALL SOUND
                MOV LCD_ADD,#00H
                ACALL SET_ADD
                MOV DPTR,#TABLE_ERROR
                ACALL WRITE_LCD
                ACALL DELAY_1s
                ACALL DELAY_1s
                RET

```

```

ICHOOSE_1:     ACALL SOUND
                MOV LCD_ADD,#00H
                ACALL SET_ADD
                MOV DPTR,#TABLE_F
                ACALL WRITE_LCD
                ACALL DELAY_1s
                ACALL DELAY_1s
                RET

```

```

;----- SUOND

```

```

SOUND:         MOV R4,#200
SOUND_1:       SETB P1.7
                ACALL DELAY_125us
                CLR P1.7
                ACALL DELAY_125us
                DJNZ R4,SOUND_1
                RET

```

```

;----- DELAY

```

```

DELAY_125us:   MOV R6,#52
DELAY_100u:    NOP
                NOP

```

```
                                DJNZ R6,DELAY_100u
                                RET
DELAY_LCD:                      MOV  R7,#002
LCD_DELAY_1:                    MOV  R6,#0E6H
LCD_DELAY_2:                    NOP
                                NOP
                                DJNZ  R6,LCD_DELAY_2
                                DJNZ  R7,LCD_DELAY_1
                                RET

DELAY_10ms:                     MOV  R7,#010
DELAY_10ms_1:                  MOV  R6,#0E6H
DELAY_10ms_2:                  NOP
                                NOP
                                DJNZ  R6,DELAY_10ms_2
                                DJNZ  R7,DELAY_10ms_1
                                RET

DELAY_500ms                     MOV  R4,#5
DELAY_1:                       ACALL DELAY_100ms
                                DJNZ  R4,DELAY_1
                                RET

DELAY_400ms:                   MOV  R4,#4
DELAY_300m:                    ACALL DELAY_100ms
                                DJNZ  R4,DELAY_300m
                                RET

DELAY_100ms:                   MOV  R7,#100
DELAY_100ms_1:                 MOV  R6,#0E6H
DELAY_100ms_2:                 NOP
                                NOP
                                DJNZ  R6,DELAY_100ms_2
                                DJNZ  R7,DELAY_100ms_1
                                RET
```

```
DELAY_1s:      MOV R5,#100
DELAY_1s_1:    ACALLDELAY_10ms
               DJNZ R5,DELAY_1s_1
               RET
```

```
;----- TABLE
```

```
TABLE:        DB 'START >#'
TABLE_1:      DB ' * '
TABLE_2:      DB ' ** '
TABLE_3:      DB ' *** '
TABLE_4:      DB ' **** '
TABLE_5:      DB ' OPEN '
TABLE_6:      DB ' CLOSE '
TABLE_ERROR:  DB ' ERROR '
TABLE_BAR:    DB ' BARCODE'
TABLE_RESET:  DB ' RESET '
TABLE_B:      DB 'KEY CODE'
TABLE_F:      DB 'NO PASS'
TABLE_CODE:   DB 00H,'1','2','3','4'
               DB '5','6','7','8','9'
               DB '*','0','X'
               END
```