```
; Program to read the I/P from 2x2 keypad and display on LED. Also if more than 1 key is pressed than all LEDs blink indicating error condition.
; codeRSHDS: 03.30.2011                    DE: 04.03.2011

ORG 00H

MOV P1, #00H        ; Make P1 O/P port
MOV P2, #0FFH       ; Make P2 I/P port

REPT:
MOV R0, P2                  ; Save I/P in R0

MOV A, R0                   ; Lower 4 bits indicate I/P.
XRL A, #0F7H        ; XOR with 11110111 (4th key pressed)
JZ CRRCT                    ; Zero indicates that A contains I/P for 4th key pressed
MOV A, R0
XRL A, #0FBH
JZ CRRCT                    ; 0 indicates that A contain I/P for 3rd key press. Go to label CRRCT to display
MOV A, R0
XRL A, #0FDH        ; All inputs and outputs are active low. Hence comparing with "11111101" rather than "00000010"
JZ CRRCT
MOV A, R0
XRL A, #0FEH
JZ CRRCT
MOV A, R0
XRL A, #0FFH        ; If no key pressed.
JZ CRRCT

ACALL INCRCT        ; If more than 1 key pressed, go to procedure INCRCT
SJMP REPT

CRRCT: MOV P1, R0
SJMP REPT


ORG 50H
INCRCT:
MOV A, #0FFH

MOV R5, #02                 ; Triple loop inside loop used here.
AGAIN: CPL A
MOV P1, A
MOV R3, #01
THERE: MOV R4, #01
HERE: DJNZ R4, HERE
DJNZ R3, THERE
DJNZ R5, AGAIN
RET

END



;Program to generate a 2 Hz wave of 50% duty cycle using Timer 1 in Mode 1
; codeRSHDS: 03.31.2011         DE: 04.03.2011


ORG 00H

MOV TMOD, #10H              ; TMOD register used to set Timer Mode

REPT:
MOV R0, #7                          ; Repeat loop 7 times to generate ON period of .25 secs
SETB P1.0                           ; Set ON
MOV TH1, #00H
MOV TL1, #00H

BACK1: CLR TF1
SETB TR1                            ; Start the timer
AGN1: JNB TF1, AGN1         ; Timer Flag gets set when the timer rolls over
CLR TR1
DJNZ R0, BACK1


MOV R0, #7                          ; To keep OFF for 0.25 sec loop 7 times
CLR P1.0                    ; Set OFF
MOV TH1, #00H
MOV TL1, #00H

BACK2: CLR TF1              ; Clear the timer flag
```

```
SETB TR1
AGN2: JNB TF1, AGN2
CLR TR1
DJNZ R0, BACK2

SJMP REPT                               ; Repeat the pulse again to create the wave

END
```

```
;Program to generate a 0.5 Hz wave of 75% duty cycle using Timer 0 in Mode 1.
; codeRSHDS: 03.31.2011          DE: 04.03.2011


ORG 00H

MOV TMOD, #01H                  ; TMOD register used to set Timer Mode

REPT:
MOV R0, #42                             ; Repeat loop 42 times to generate ON period of 1.5 secs
SETB P1.1                               ; Set ON
MOV TL0, #00H
MOV TH0, #00H

BACK1: CLR TF0
SETB TR0                                ; Start the timer
AGN1: JNB TF0, AGN1             ; Timer Flag gets set when the timer rolls over
CLR TR0
DJNZ R0, BACK1


MOV R0, #14                             ; To keep OFF for 0.5 sec loop 14 times
CLR P1.1                        ; Set OFF
MOV TL0, #00H
MOV TH0, #00H

BACK2: CLR TF0                  ; Clear the timer flag
SETB TR0
AGN2: JNB TF0, AGN2
CLR TR0
DJNZ R0, BACK2

SJMP REPT                               ; Repeat the pulse again to create the wave

END
```

```
; Program to send the name serially from uC to PC.
; codeRSH               DS: 04.03.2011                          DE: 04.03.2011

ORG 00H

MOV A, PCON         ; Since EdSim's UART doesn't support baud rate..
SETB ACC.7          ; .. of 9600. So doubling it to 19200 using PCON reg
MOV PCON, A         ; This is the way to use, since PCON is not bit-addressable

MOV DPTR, #60H   ;Value to be displayed is stored at 60h location
MOV TMOD, #20H  ; Set Timer 1 Mode 2
MOV TH1, #-3        ; Load T1 to set baud rate of 9600 * 2
MOV SCON, #50H   ; 1 start, 1 stop bit with no parity mode.
SETB TR1                        ; start the timer

RPT: CLR A
MOVC A, @A+DPTR                 ; Load character from ROM in Accumulator
JZ EXIT                         ; If last character then exit

MOV SBUF, A         ; Send the character
AGAIN: JNB TI, AGAIN            ; Loop till last bit of character not sent serially.
CLR TI              ; Clear Transmit Interrupt flag to enable transmission of next character

INC DPTR            ; DPTR now points to next character
SJMP RPT            ; Repeat again

EXIT: SJMP EXIT
```

```
; ------------------
ORG 60H
DB "codeRSH"                    ;Text to send
DB 0

END




; Program to send an ASCII value from PC to uC, convert to binary and then display it on LEDs connected to P0. Also generate a square wave of 1Hz
on P2.0
; codeRSHDS: 04.03.2011                     DE: 04.04.2011

ORG 00H
LJMP MAIN

ORG 0BH             ; ISR for Timer 0
DJNZ R0, OVER
MOV R0, #14
CPL P1.0
OVER: RETI

ORG 23H             ; ISR for Serial should be stored at this location
LJMP SERIAL

ORG 30H             ; Look up table
DB 0H, 1H, 2H, 3H, 4H, 5H, 6H, 7H, 8H, 9H

ORG 60H
MAIN:                           ; Main Program
MOV R0, #14                     ; Timer 0 loops 14 times to get a wave of 1Hz.
MOV DPTR, #00H
MOV IE, #10010010B              ; Enable Serial and Timer 0 Interrupt
MOV TMOD, #21H
MOV TH1, #-6
MOV TL0, #00H
MOV TH0, #00H
MOV SCON, #50H

SETB TR1            ; Start Timer 1
SETB TR0            ; Start Timer 0

AGAIN: SJMP AGAIN               ; Continuosly loop here until an interrupt occurs


ORG 90H
SERIAL:                         ; Serial Reception ISR
MOV A, SBUF
MOVC A, @A+DPTR                 ; Conversion from ASCII to binary using look up table
MOV P1, A
CLR RI
RETI

END






; Program to display counter from 0 to 9 using a 7 segment display.
; codeRSHDS: 04.03.2011         DE: 04.03.2011

ORG 00H

MOV R0, #0
MOV DPTR, #60H
CLR P3.3
CLR P3.4

RPT:
            MOV A, R0
            MOVC A, @A+DPTR
            MOV P1, A
            ACALL DELAY
            CJNE R0, #9, THERE
            MOV R0, #0FFH
            THERE: INC R0
            SJMP RPT

DELAY:
            MOV R2, #1
            MORE: MOV R3, #1
            AGAIN: DJNZ R3, AGAIN
            DJNZ R2, MORE
            RET

ORG 60H
DB C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 83H, F8H, 80H, 98H

END
```

```
ORG 00H

MOV R0, #0          ; Pointer for lower counter
MOV R1, #0          ; Pointer for upper counter
MOV DPTR, #60H      ; 7 seg values for digit saved at 60H M/M location
SETB P3.3
CLR     P3.4


STRT:
            ;SETB P3.3          ; Set the 2nd..
            ;CLR     P3.4       ; ..7-seg display
            MOV A, R1
            MOVC A, @A+DPTR
            MOV R7,A
            ;MOV P1, A          ; Show value on the display

            ;ACALL DELAY
            ;MOV P1, #0FFH      ; Clears P1 i.e. LED of previous content.
            CJNE R1, #9, HERE
            MOV R1, #0FFH       ; If R1 reaches 9 then roll over. FFH = -1
            HERE: INC R1

RPT:

            SETB P3.3           ; Set the 2nd..
            CLR     P3.4        ; ..7-seg display
            MOV P1,R7

            ACALL DELAY
            MOV P1, #0FFH

            CLR P3.3  ; Set the 1st..
            CLR P3.4  ; ..7-seg Display
            MOV A, R0
            MOVC A, @A+DPTR             ; Load in A value stored at specified M/M location.
            MOV P1, A           ; Only P1 is connected to 7-seg display, so counters will have to be multiplexed.

            ACALL DELAY
            MOV P1, #0FFH
            CJNE R0, #9, THERE
            MOV R0, #0H
            SJMP STRT                   ; If smaller counter rolls over go to higher counter
            THERE: INC R0
            SJMP RPT

DELAY:
            MOV R2, #1          ; Loop inside loop used for delay
            MORE: MOV R3, #1
            AGAIN: DJNZ R3, AGAIN
            DJNZ R2, MORE
            RET

ORG 60H
; These are hex codes for digits to be displayed on 7-seg Display. These are active-low values
DB C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 83H, F8H, 80H, 98H

END
```