

```
; Timing code assumes 1.2MHz Clock
```

```
#include < Bitdef.h>
#include < Io51.h>
```

```
;LCD Registers addresses
Lcd_cmd_wr Equ 0
Lcd_data_wr Equ 1
Lcd_busy_rd Equ 2
Lcd_data_rd Equ 3
```

```
;LCD Commands
Lcd_cls Equ 1
Lcd_home Equ 2
Lcd_setmode Equ 4
Lcd_setvisible Equ 8
Lcd_shift Equ 16
Lcd_setfunction Equ 32
Lcd_setcgaddr Equ 64
Lcd_setddaddr Equ 128
```

```
;Reset vector
Org 0000h
jmp start
```

```
;Start of the program
Org 0100h
```

```
String1a : Db                                'P r o j E c T '
Db 0
String1b : Db                                'T a b L o R a V a n'
Db 0
String2 : Db                                'A m i R i Y a N'
Db 0

String3 : Db                                'Z e M e S T A N'
Db 0
String4 : Db                                '1393..'
Db 0
```

```
Start : Mov A , #038h
Call Wrcmd
```

```
Loop : Mov A , #lcd_setvisible + 6 ; Make The Display & Blink Visible:
Call Wrcmd
```

```
mov R7,#2
Loop2:
mov DPTR,#string1a
Call Wrstr
```

```
mov DPTR,#200
Call Wtms
```

```
mov A,#LCD_CLS ;Clear screen
Call Wrcmd
```

```
mov DPTR,#string1b
Call Wrstr
```

```
mov DPTR,#200
Call Wtms
```

```
mov A,#LCD_CLS ;Clear screen
Call Wrcmd
```

```
Djnz R7 , Loop2
```

```
mov DPTR,#string1a
Call Wrstr
```

```
mov DPTR,#400
Call Wtms
```

```
mov A,#LCD_SETDDADDR+64
Call Wrcmd
```

```

mov DPTR,#string2
Call Wrslow

    mov DPTR,#200
    Call Wtms

mov A,#LCD_CLS    ;Clear screen
Call Wrcmd

mov DPTR,#string3
Call Wrslow

    mov A,#LCD_SETDDADDR+64
    Call Wrcmd

mov DPTR,#string4
Call Wrslow

mov A,#LCD_SETVISIBLE+7    ;Show the blink cursor as well.
Call Wrcmd

    mov DPTR,#2000
    Call Wtms

mov A,#LCD_CLS    ;Clear screen
Call Wrcmd

jmp loop

;Sub routine to write null terminated string at DPTR in program ram.
Wrstr : Mov R0 , #lcd_data_wr
Wrstr1 : Clr A
    Movc A , @a + Dptr
    Jz Wrstr2
    Movx @r0 , A
    Call Wtbusy
    inc DPTR
    push DPL
    push DPH
    pop DPH
    pop DPL
    jmp wrstr1
Wrstr2 : Ret

;Sub routine to write null terminated string at DPTR in program ram. Slowly
Wrslw : Mov R0 , #lcd_data_wr
Wrslw1 : Clr A
    Movc A , @a + Dptr
    Jz Wrslw2
    Movx @r0 , A
    Call Wtbusy
    inc DPTR
    push DPL
    push DPH
        mov DPTR,#100
        Call Wtms
    pop DPH
    pop DPL
    jmp wrslw1
Wrslw2 : Ret

;Sub routine to write command:
Wrcmd : Mov R0 , #lcd_cmd_wr
    Movx @r0 , A
    jmp wtbusy

;Sub routine to write character:
Wrchar : Mov R0 , #lcd_data_wr
    Movx @r0 , A

;Subroutine to wait for busy clear
Wtbusy : Mov R1 , #lcd_busy_rd
    Movx A , @r1
    Jb Acc.7 , Wtbusy
    ret

```

```
;Wait for number of seconds in A
Wtsec : Push Acc
       Call Wtms
       pop ACC
       dec A
       Jnz Wtsec
       ret
```

```
;Wait for number of milliseconds in DPTR
Wtms : Xrl Dpl , #0ffh ; Can 't do DEC DPTR, so do the
loop by forming 2's complement
       Xrl Dph , #0ffh ; And Incrementing Instead.
       inc DPTR
Wtms1 : Mov Tl0 , #09ch ; 100 Ticks Before Overflow = 1ms At 1.2mhz Clock
       mov TH0,#0FFh
       mov TMOD,#1 ;Timer 0 mode 1
       Setb Tcon.4 ; Timer 0 Runs
Wtms2 : Jnb Tcon.5 , Wtms2
       clr TCON.4 ;Timer 0 stops
       clr TCON.5
       inc DPTR
       mov A,DPL
       Orl A , Dph
       Jnz Wtms1
       ret
```

End