```
;*******************************************************************************
; File: Project1_LED.asm
; 31st MAY, 2012
; Swapnil Christian
;
; This system program is for interfacing a 8 LED bar to Nanocore12 device platform.
; The LED bar is connected to breakout board using current limiting resistors.
; When the system is powered on the LEDs on the bar start lighting up from left
; to right with a 1 second delay between lighting up of each LED. When all the
; LEDs are lit, they flash 4 times before the sequence is repeated.
;*******************************************************************************

PortT     EQU       $240      ; Initialization of port T register
DDRT      EQU       $242      ; Initialization of port T data direction register
INITRG    EQU       $11       ; Initialization of internal registers position register
INITRM    EQU       $10       ; Initialization of internal RAM position register
CLKSEL    EQU       $39       ; Initialization of clock select register
PLLCTL    EQU       $3A       ; Initialization of PLL control register
CRGFLG    EQU       $37       ; Initialization of CRG flags register
SYNR      EQU       $34       ; Initialization of synthesizer register
REFDV     EQU       $35       ; Initialization of reference divider register
COPCTL    EQU       $3C       ; Initialization of COP control register
TSCR1     EQU       $46       ; Initialization of timer system control register 1
TSCR2     EQU       $4D       ; Initialization of timer system control register 2
TIOS      EQU       $40       ; Initialization of timer input capture/output compare select
TCNT      EQU       $44       ; Initialization of timer count register
TC0       EQU       $50       ; Initialization of timer input caputer/output compare register 0
TFLG1     EQU       $4E       ; Initialization of main timer interrupt flag1

;*******************************************************************************
;         Beginning of RAM variables
;*******************************************************************************
          ORG       $3800

;*******************************************************************************
;         Main code
;*******************************************************************************
          ORG       $4000               ; Beginning of flash EEPROM
START:    LDS       #$3FCE              ; Top of the stack
          SEI                           ; Turn off interrupts
          MOVB      #$00, INITRG        ; I/O and control registers start at $0000
          MOVB      #$39, INITRM        ; RAM ends at $3FFF

;*******************************************************************************
;         Set up PLL so that the clock = 24 MHz
;*******************************************************************************
          BCLR      CLKSEL, $80         ; Disengage PLL from system
          BSET      PLLCTL, $40         ; Turn on PLL
          MOVB      #$2, SYNR ; Set PLL multiplier
          MOVB      #$0, REFDV          ; Set PLL divider
          NOP                           ; No operation
          NOP                           ; No operation
PLP:      BRCLR     CRGFLG, $08, PLP    ; while (!(crg.crgflg.bit.lock==1))
          BSET      CLKSEL, $80         ; engage PLL

          CLI                           ; Turn on interrupts

;*******************************************************************************
;         Program code --> Flash memory
;*******************************************************************************
          LDAB      #$FF                ; <B> <-- 11111111
          STAB      DDRT                ; Make port T outbound

REP:      LDAB      #$00                ; <B> <-- 00000000  Loop counter
          STAB      PortT               ; Output the count in <B> to port T --> All LEDs OFF
LOOP:     LDX       #TABLE              ; Index register X points to mem. location $5000
          ABX                           ; X points to mem. address $5000 + <B>
          LDAA      $00, X              ; <A> <-- data located at mem. address pointed by X
          STAA      PortT               ; Output data to light LED
          JSR       DELAY               ; Wait 1 sec
          INCB                          ; <B> <-- <B> + $01
          CMPB      #$08                ; Check if all lights are lit or not
          BNE       LOOP                ; If not then fetch the next sequence from LUT

          ; Case when all LEDs are lit
          ; First turn all the LEDs OFF, and then make them flash 4 times

          LDAA      #$00                ; <A> <-- 00000000
          STAA      PortT               ; Turn off all LEDs
          JSR       DELAY               ; Wait 1 sec
```

```
          LDAB      #$00                ; <B> <-- 00000000  Loop counter
FLASH:    LDAA      #$FF                ; <A> <-- 11111111
          STAA      PortT               ; Turn on all LEDs
          JSR       DELAY               ; Wait 1 sec
          LDAA      #$00                ; <A> <-- 00000000
          STAA      PortT               ; Turn off all LEDs
          JSR       DELAY               ; Wait 1 sec
          INCB                          ; <B> <-- <B> + $01
          CMPB      #$04                ; Check if the LEDs have flashed 4 times or not
          BNE       FLASH               ; Repeat until the lights have flashed 4 time

          JMP       REP                 ; Repeat process



;*******************************************************************************
;         Delay using built-in timer functions for accuracy
;*******************************************************************************
DELAY:    PSHA                          ; Save <A> on stack
          PSHB                          ; Save <B> on stack
          LDY       #10                 ; Repeat subroutine 10 times
          MOVB      #$90, TSCR1         ; Enable TCNT & fast flags clear
          MOVB      #$06, TSCR2         ; Configure prescale factor to 64
          MOVB      #$01, TIOS          ; Enable 0C0
          LDD       TCNT                ; Get current TCNT value
AGAIN:    ADDD      #37500              ; Start an output compare operation
          STD       TC0                 ; With 100ms time delay
WAIT:     BRCLR     TFLG1, $01, WAIT    ; Wait for TCNT to catch up
          LDD       TC0                 ; Get the value in TC0
          DBNE      Y, AGAIN  ; 10 * 100 ms = 1 sec
          PULB                          ; Restore content of B from stack
          PULA                          ; Restore content of A from stack
          RTS

;*******************************************************************************
;         Look up table for lighting the LED
;*******************************************************************************
          ORG       $5000               ; Starting memory address for LUT
TABLE:    DC.B      $80,$C0,$E0,$F0     ;         <-- Data stored for LED sequence
          DC.B      $F8,$FC,$FE,$FF     ;
;
;         LED sequence
;         Port T    | PM7 | PM6 | PM5 | PM4 | PM3 | PM2 | PM1 | PM0 |
;         $80       |  1  |  0  |  0  |  0  |  0  |  0  |  0  |  0  |
;         $C0       |  1  |  1  |  0  |  0  |  0  |  0  |  0  |  0  |
;         $E0       |  1  |  1  |  1  |  0  |  0  |  0  |  0  |  0  |
;         $F0     |  1  |  1  |  1  |  1  |  0  |  0  |  0  |  0  |
;         $F8       |  1  |  1  |  1  |  1  |  1  |  0  |  0  |  0  |
;         $FC       |  1  |  1  |  1  |  1  |  1  |  1  |  0  |  0  |
;         $FE       |  1  |  1  |  1  |  1  |  1  |  1  |  1  |  0  |
;         $FF       |  1  |  1  |  1  |  1  |  1  |  1  |  1  |  1  |

;*******************************************************************************
;         Define power-on reset interrupt vector
;*******************************************************************************
          ORG       $FFFE     ; $FFFE, $FFFF = Power-on reset interrupt vector location
          FDB       START     ; Specify instruction to execute on power up


          END
```