```
$NOMOD51
$include (c8051f120.inc)

;---------------- INIT ----------------;
;- Setup board
;-
;- Pin config:
;- P0.0 = dir
;- P0.1 = step
;-
Init:       mov    SFRPAGE,#CONFIG_PAGE
            mov    XBR2,#40h                ; Enable the Port I/O Crossbar

            mov    WDTCN,#0ADh
            mov    R3,#0D
            mov    R7,#250D


;---------------- MAIN ----------------;
;- Invert Step pin every 20ms and Invert Direction pin every 5 seconds at a rate of 250
step/s
;-

Main:

MainLoop:   mov     R4,#226D     ; load number of steps
            mov     R5,#4D      ; R4 = LB, R5 = HB

            acall   StepN       ; step 1250 times, at speed of 250 steps/s, takes 5 sec
onds to complete

            cpl     P0.0
            sjmp    MainLoop


;---------------- STEPN ---------------;
;- a function that executes n steps.
;- load HB of n in R5, and LB of n in R4
;-

StepN:      mov     R6,#255D
StepL1:     acall   Step        ; invert step pin
            djnz    R6,StepL1
            mov     R6,#255D
            djnz    R5,StepL1

StepL2:     acall   Step
            djnz    R4,StepN



;---------------- STEP ---------------;
;- a function that steps once.

Step:       acall   Delay1ms
            acall   Delay1ms
            acall   Delay1ms
            acall   Delay1ms
            cpl     P0.1
            acall   Delay1ms
            acall   Delay1ms
            acall   Delay1ms
            acall   Delay1ms
            cpl     P0.1

            ret

;---------------- DELAY ---------------;
```

```
;- Consumes 1333.333 MC per call to delay for 1ms.
;-
;- Crystal Freq = 16MHz  =>  MC = 10*3us * 16MHz / 12 = 1333.333...
;- MC from Main,  X =  acall + djnz = 2 + 2 = 4
;- MC from Delay, Y =  mov*3 + djnz*(255 + 255 + 151) + cjne + ret = 3*1 + 2*3(255 + 25
5 + 151) + 2 + 1 = 1328
;- Y + X = 132, 1.333 missing MC / cycle
;-
;- MC in Error,   z = mov + djnz*6 = 1 + 2*6 = 13 / 0.1*cycles
;-
;- Software error = 0MC (in theory)
;-

Delay1ms:   mov     R0,#0ffh                ; 1
            mov     R1,#0ffh                ; 1
            mov     R2,#151D                ; 1
Loop1:      djnz    R0,Loop1                ; 2*255
Loop2:      djnz    R1,Loop2                ; 2*255
Loop3:      djnz    R2,Loop3                ; 2*151
            inc     R3
            cjne    R3,#10D,Out             ; 2 | jump out if no error correction needed
 (due to .3333)

Error:      mov     R3,#6D                  ; 1
ErrLoop:    djnz    R3, ErrLoop             ; 2*6
                                            ; (error correction every 10 cycles = 1 + 12
 = 13)

Out:        ret                             ; 1



;---------------- EOF ----------------;

END
```