;File: pll42dly.a51

;Author: Eckart Hartmann Date:05/10/2003

;Development progress: Pll842.df

;

;PllDly==========Software delay

;C Function prototype: char PllDly(uint uDlyMs);

;Description of Function: Causes software delay of uDlyMs milliseconds.

;User interface: Put delay in parameter1. Call PllDly.

; Returns with 1 (always).

;Robustness: Due to unavoidable rounding the delay times have typical

; errors of up to 3%.

; An offset of about 15us\*2^CD (as much as 2ms at CD=7)

; becomes dominant for large CD and small delays.

; Delay will be increased if normal interrupts occur during delay.

; Delay can be decreased if fast interrupts occur during delay.

; Delay will be incorrect if CD is changed by an interrupt.

; Delay will be incorrect inside an interrupt unless FINT=0 or CD=0.

;Side effects: Overwrites a, p and ip2l.

;

NAME PLLDLY

$NOMOD51

$IC(..kei842.inc) ; Parameter passing registers for Keil .

$IC(..kei842.dat) ; SFR definition for Keil .

;

public \_PllDly

?PR?\_PllDly?PLLDLY SEGMENT CODE

RSEG ?PR?\_PllDly?PLLDLY

\_PllDly:

mov a,iP1l ;Start on low byte.

jnz PlDLr7

PlDCh6: mov a,iP1h

jz PlDR ;High byte with low byte full scale.

dec iP1h

PlDLr7: mov a,PLLCON

anl a,#7 ;a = 2^(7-CD);

inc a

inc a

movc a,@a+pc ;//Use CD in reverse lookup table.

sjmp PlDCd

db 128

db 64

db 32

db 16

db 8

db 4

db 2

db 1

PlDCd: mov iP2li,a ;a

LpC5: mov iP2hi,#42 ; \* 44

djnz iP2hi,$ ; \* 3 cycles

djnz iP2li,LpC5

djnz iP1l,PlDCd

sjmp PlDCh6

PlDR: mov cP1l,#1 ;return(1); // Always.

ret

;

;Function End==========================================================Function End

end

[Session]

FileCount=2

Active=1

Filespec1=PLL42DLY.A51

Pos1=0,1,-1,-1,-1,-1,160,295,832,1045

Loc1=27,0,9,85,0

Filespec2=PLL842.C

Pos2=0,1,-1,-1,-1,-1,3,6,675,756

Loc2=0,0,0,0,0