

Description

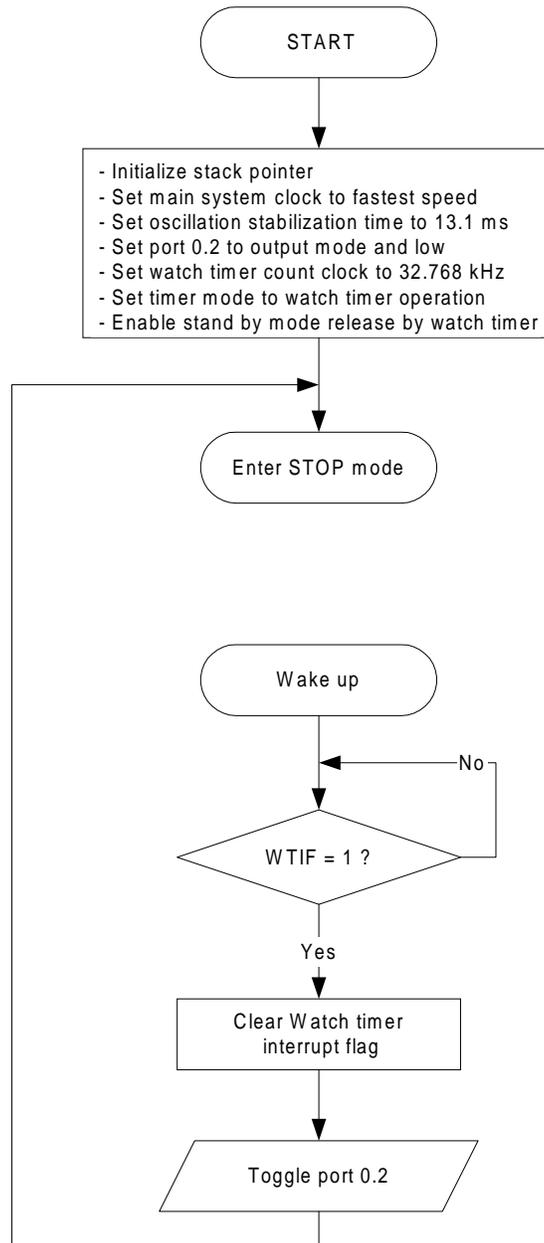
Stop mode is a standby function on the μ PD7805x/78005x subseries that is used to reduce CPU power consumption. Execution of the STOP instruction initiates stop mode, suspending operation of the CPU and oscillation of the main system clock. Stop mode is exited upon any interrupt request. If interrupt handling is enabled (EI), the wakeup starts executing the code in the ISR. If interrupt handling is disabled (DI), the next instruction after the STOP instruction is executed.

In this program, the watch timer generates a nonvectored interrupt every 0.5 seconds to wake up the microcontroller. After a preprogrammed oscillator stabilization time of 13.1 ms elapses, the CPU toggles port 0.2 and puts the microcontroller back into stop mode.

Program Specifications

- CPU runs from the main system clock at fastest speed
- Watch timer clock is subsystem clock (32.768 kHz)
- CPU wakes up every 0.5 seconds from a nonvectored interrupt of the watch timer
- Pins used in program: P02/INTP2 (toggles every 0.5 seconds)

Flowchart



Assembly Language Program

```

;*****
; Date:          08/26/1999
;
; Parameters: - fastest CPU clock,
;              (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;              - Watch timer clock source is subsystem clock (32.768 kHz)
;              - Watch timer interval time is 0.5 s
;              - Port 0.2 toggles every time (0.5 s) when STOP mode is exited
;*****

;=====
; Specify Interrupt Vectors      =
;=====

Res_Vec CSEG AT 0000h           ; Set main program start vector
        DW      Start

;*=====
; Main Program                   =
;=====

MAIN    CSEG
Start:  DI                      ; Disable interrupts
        MOVW   AX, #0FE20h      ; Load stack pointer address
        MOVW   SP, AX           ; Set stack pointer
        MOV    OSMS,#01h        ; Don't use scaler
        MOV    PCC, #00h        ; Main system clock at fastest setting
        MOV    OSTS, #03h       ; Set oscillation stabilization time to 13.1 ms
        CLR1   P0.2             ; Latch port 0.2 to low
        CLR1   PM0.2           ; Set port 0.2 to output mode
        MOV    TCL2,#10h        ; Watch timer counter clock is 32.768 kHz
        MOV    TMC2,#56h       ; Set timer mode register to watch timer operation
        CLR1   WTMK            ; Enable stand by mode release by watch timer
Loop:   STOP                   ; Enter STOP mode
NOTSET: NOP                    ; Needed NOP for possible interrupt acknowledge
        BF     WTIF, $NOTSET    ; Test if interrupt came form watch timer
        CLR1   WTIF            ; Clear watch timer interrupt flag
        XOR    P0, #04h        ; Toggle port 0.2 to indicate wake up
        BR     $Loop           ; Branch back to Loop
        END

```

C Language Program

```

/*****
; Date:          08/26/1999
;
; Parameters: - fastest CPU clock,
;              (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;              - Watch timer clock source is subsystem clock (32.768 kHz)
;              - Watch timer interval time is 0.5 s
;              - Port 0.2 toggles everytime (0.5 s) when STOP mode is exited
;*****
/* extension functions in K0/K0S compiler */

#pragma sfr      /* key word to allow SFR names in C code */
#pragma asm     /* key word to allow ASM statements in C code */
#pragma STOP    /* key word for STOP instruction in C code */
#pragma NOP     /* key word for NOP instruction in C code */

/*=====
;          Specify Interrupt Vectors          =
;=====*/

/*=====
;          Constants and Variables           =
;=====*/

#define TRUE    1
#define FALSE   0

/*=====
;          Main Program                      =
;=====*/

void main(void)
{
    OSMS = 0x01;      /* Don't use scaler */
    PCC = 0x00;      /* Main system clock at fastest setting */
    OSTS = 0x03;     /* Set oscillation stabilization time to 13.1 ms */
    P0.2 = 0;        /* Latch port 0.2 to low */
    PM0.2 = 0;       /* Set port 0.2 to output mode */
    TCL2 = 0x10;     /* Watch timer counter clock is 32.768 kHz */
    TMC2 = 0x56;     /* Set timer mode register to watch timer operation */
    WTMK = 0;        /* Enable stand by mode release by watch timer */
    while(TRUE)
    {
        STOP();      /* Enter STOP mode */
        NOP();        /* Needed delay for wake up cycle */
        while(!WTIF); /* Test if interrupt came from watch timer */
        WTIF = 0;     /* Clear watch timer interrupt flag */
        P0 ^= 0x04;   /* Toggle port 0.2 to indicate wake up */
    }
}
/* end of function main() */

```



For literature, call **1-800-366-9782** 7 a.m. to 6 p.m. Pacific time
or FAX your request to **1-800-729-9288**
or visit our web site at **www.necel.com**

In North America: No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics Inc. (NECEL). The information in this document is subject to change without notice. All devices sold by NECEL are covered by the provisions appearing in NECEL Terms and Conditions of Sales only. Including the limitation of liability, warranty, and patent provisions. NECEL makes no warranty, express, statutory, implied or by description, regarding information set forth herein or regarding the freedom of the described devices from patent infringement. NECEL assumes no responsibility for any errors that may appear in this document. NECEL makes no commitments to update or to keep current information contained in this document. The devices listed in this document are not suitable for use in applications such as, but not limited to, aircraft control systems, aerospace equipment, submarine cables, nuclear reactor control systems, and life support systems. "Standard" quality grade devices are recommended for computers, office equipment, communication equipment, test and measurement equipment, machine tools, industrial robots, audio and visual equipment, and other consumer products. For automotive and transportation equipment, traffic control systems, anti-disaster and anti-crime systems, it is recommended that the customer contact the responsible NECEL salesperson to determine the reliability requirements for any such application and any cost adder. NECEL does not recommend or approve use of any of its products in life support devices or systems or in any application where failure could result in injury or death. If customers wish to use NECEL devices in applications not intended by NECEL, customer must contact the responsible NECEL salespeople to determine NECEL's willingness to support a given application.