

On-Chip Peripheral Program Example

August 1999

Description

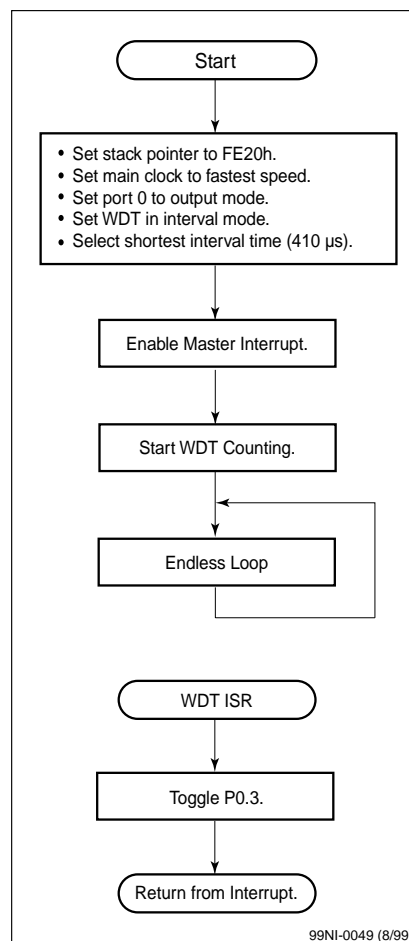
The watchdog timer (WDT) in the μ PD7805x/78005x subseries can be used in watchdog timer mode or interval timer mode.

This program demonstrates interval timer mode, where the WDT functions as a standard 8-bit timer that generates interrupt requests repeatedly at intervals set to the timer clock select register (TCL2). In the interrupt service routine (ISR), the program continuously toggles port 0 bit 3.

Program Specifications

- ❑ Watch dog timer count: $f_{xx}/2^3 = 625 \text{ kHz}$ at 5-MHz main system clock
- ❑ Interval time: $t = 410 \mu\text{s}$
- ❑ Toggle frequency: $f = 1220 \text{ Hz}$
- ❑ Pins used in program: P03/INTP3 (toggles every $410 \mu\text{s}$)

Flowchart



Assembly Language Program

```

;*****
; Date:          07/14/1999
;
; Parameters:    - fastest CPU clock
;                (fx = 5.00 MHz; 1 CPU clock cycle = 200 ns)
;                - interval time is 2^11/fx (410 µs)
;                - use WDT in Interval timer mode
;                - port 0.3 toggles every 410 µs
;
;*****

;=====
;=      Specify Interrupt Vectors      =
;=====
Res_Vec      CSEG AT 0000h      ; Set main program start vector
              DW      Start

WDT_Vec      ORG    0004h      ; Set interrupt vector for WDT
              DW      WDT_ISR

;=====
;=      Main Program                    =
;=====
MAIN         CSEG
Start:       DI                ; Disable interrupts
              MOVW    AX,      #0FE20h    ; Load SP address
              MOVW    SP,      AX        ; Set Stack Pointer

              MOV     OSMS, #01h        ; Don't use scaler
              MOV     PCC, #00h        ; Main system clock at fastest setting

              MOV     P0, #00h          ; Clear port 0 latch
              MOV     PM0, #00h        ; P0.0 to P0.3 are outputs
              MOV     TCL2, #00h        ; WDT clock will be 2^11/fx (410 µs)
              MOV     WDTM, #00h        ; Set WDT into Interval timer mode
              CLR1     TMMK4            ; Unmask the WDT interrupt mask bit
              SET1     RUN              ; Start WDT operation
              EI                ; Enable interrupts

Loop1:       BR      $Loop1            ; Endless loop

;=====
;=      Interrupt Routine                =
;=====
ISR          CSEG
WDT_ISR:     XOR     P0, #08h          ; Toggle port 0.3
              RETI                    ; Return from interrupt

END

```

C Language Program

```

/*****
; Date:      07/14/1999
;
; Parameters: - fastest CPU clock
;              (fx = 5.00 MHz; 1 CPU clock cycle = 200 ns)
;              - interval time is 2^11/fx (410 µs)
;              - use WDT in Interval timer mode
;              - port 0.3 toggles every 410 µs
;
*****/
/* extension functions in K0/K0S compiler */

#pragma sfr /* key word to allow SFR names in C code */
#pragma DI /* key word for DI instruction in C code */
#pragma EI /* key word for EI instruction in C code */

/*;=====
;      Specify Interrupt Vectors      =
;=====*/
/* Set interrupt vector for the Watchdog timer */

#pragma interrupt INTWDT WDT_ISR
/*;=====
;=      Constants/Variables      =
;=====*/
#define TRUE      1
#define FALSE     0

/*;=====
;=      Main Program      =
;=====*/
void main(void)
{
    DI(); /* Disable interrupts */

    OSMS = 0x01; /* Don't use scaler */
    PCC = 0x00; /* Main system clock at fastest setting */
    P0 = 0x00; /* Latch port 0 low */
    PM0 = 0x00; /* Set P0.0 - P0.3 to outputs */
    TCL2 = 0x00; /* interval timer = 2^11/fx (410 µs) */
    WDTM = 0x00; /* Set WDT into interval timer mode */
    TMMK4 = 0; /* Unmask the WDT interrupt mask bit */
    RUN = 1; /* Start WDT running */
    EI(); /* Enable interrupts */
    while(TRUE)
    {
        /* endless loop */
    }
    /* end of function main() */
/*;=====
;      Interrupt Routine      =
;=====*/
void WDT_ISR(void)
{
    P0 ^= 0x08; /* toggle port 0.3 */
}
/* end of WDT_ISR */

```



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.