

#### Description

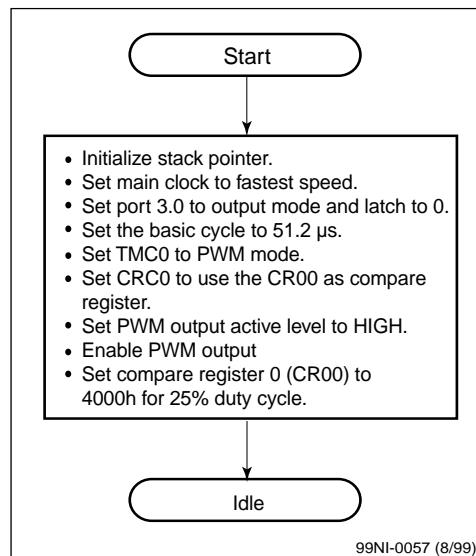
The 16-bit timer/event counter (TM0) in the  $\mu$ PD7805x/78005x subseries can be used as an interval timer, external event counter, pulse-width modulation (PWM) output, square-wave output, one-shot pulse output, or for pulse-width measurement.

This program demonstrates the 16-bit timer/event counter in PWM output mode. One PWM cycle consists of 64 cycles (or basic cycles). Each basic cycle has a period of 51.2  $\mu$ s. In this program, the duty of each basic cycle is 12.8  $\mu$ s (25%) and the pulses are output to the timer's TO0/P30 output pin.

#### Program Specifications

- Pulse-width modulation output resolution: 14-bit
- Basic pulse-width modulation cycle: 51.2  $\mu$ s ( $2^8/f_x$ ) at  $f_x = 5$  MHz
- Duty cycle: 25 % form basic pulse-width modulation cycle (12.8  $\mu$ s)
- Active level: high
- Output port: TO0/P30
- Pins used in program: TO0/P30 (output of the PWM signal)

#### Flowchart



## Assembly Language Program

```

;*****
; Date:          03/08/1999
;
; Parameters: - fastest CPU clock,
;              (fx=5 MHz; 1 CPU clock cycle = 200 ns)
;              - PWM output resolution: 14-bit
;              - PWM output basic cycle: 51.2 μs (2^8/fx) fx = 5 MHz
;              - Count clock:      fx = 5 MHz
;              - Duty cycle:      12.8 μs (25 %)
;              - Active level:    High
;              - Output port:    T00/P30
;*****

;=====
;=          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 SP address
          MOVW       SP, AX           ; Set Stack Pointer
          MOV        OSMS,#01h        ; Don't use scaler
          MOV        PCC, #00h        ; Main system clock at fastest setting
          CLR1       P3.0             ; Latch port 3.0 low
          CLR1       PM3.0           ; Set port as output
          MOV        TCL0,#20h        ; Set counter clock to fx = 5 MHz to get a
          ;          basic cycle of 51.2 μs (2^8/counter_clock)
          MOV        TMC0,#02h        ; Set to PWM output mode
          MOV        CRC0,#00H        ; Set TM0 operation to compare register
          CLR1       TOC0.1          ; Active level = HIGH
          MOVW       CR00,#4000h      ; Set Compare register to 4000h for
          ;          25 % duty cycle
          SET1       TOE0             ; 16-bit timer counter output enable
Loop1:    NOP                          ; Needed NOP for interrupt acknowledge
          BR         $Loop1          ; Branch back to Loop1
          END

```

### C Language Program

```

/*****
; Date:          03/08/1999
;
; Parameters: - fastest CPU clock,
;              (fx=5 MHz; 1 CPU clock cycle = 200 ns)
;              - PWM output resolution: 14-bit
;              - PWM output basic cycle: 51.2 μs (2^8/fx) fx = 5 MHz
;              - Count clock:          fx = 5 MHz
;              - Duty cycle:          12.8 μs (25 %)
;              - Active level:        High
;              - Output port:         T00/P30
;*****/

/* 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 */

/*=====
;          Constants/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 */
    P3.0 = 0;        /* Latch port 3.0 low */
    PM3.0 = 0;       /* Set port 3.0 an Output */
    TCL0 = 0x20;     /* Select counter clock to fx = 5 MHz to get basic
                       cycle of 51.2 μs (2^8/counter_clock) */
    TMC0 = 0x02;     /* Set to PWM output mode */
    CRC0 = 0x00;     /* Set TM0 operation to compare register */
    TOC0.1 = 0;     /* Active level = HIGH */
    CR00 = 0x4000;   /* Set Compare register to 4000h for 25 % duty cycle */
    TOE0 = 1;       /* 16-bit timer/event counter output Enable */
    while(TRUE);    /* Loop here */
}
/* 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](http://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.