On-Chip Peripheral Program Example August 1999

Description

The µPD7805x/78005x subseries has an 8-bit real-time output port that can be triggered by either of two internal 8-bit timers or an external interrupt. Using real-time output, a signal can be output without jitter, which is suitable for control of stepper motors.

This program demonstrates the real-time output port triggered by an internal timer interrupt. Each time timer 1 times out, an interrupt automatically triggers the real-time output. Previously toggled data in the port latch is output immediately.

Program Specifications

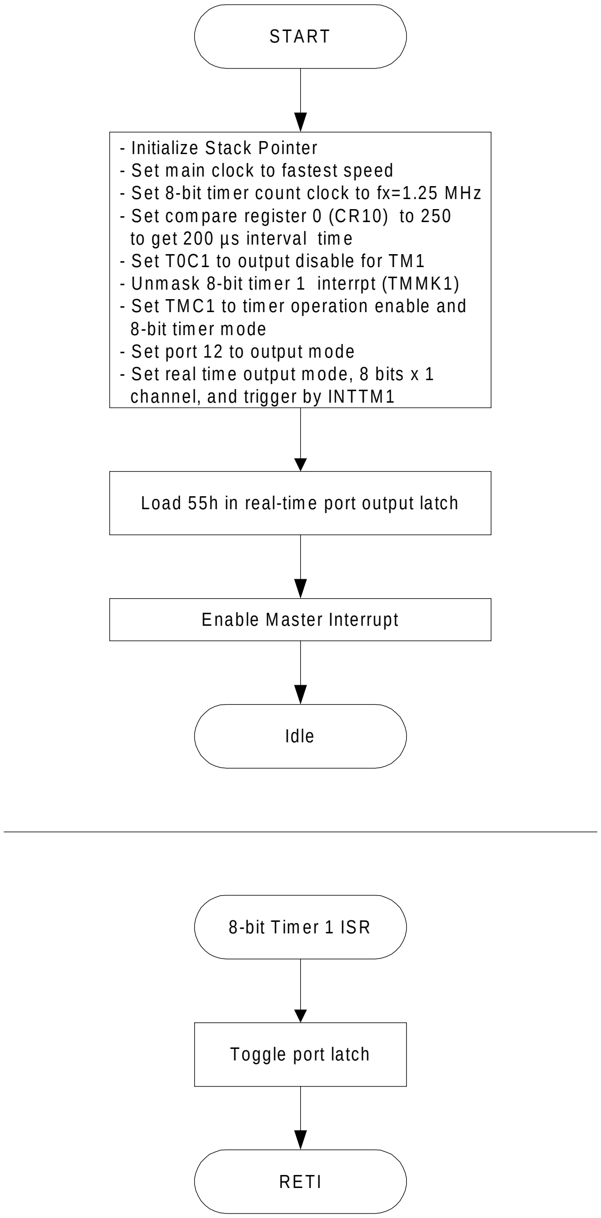
❑ Operation mode: 8 bits x 1 channel

❑ Output trigger: 8-bit timer 1 (TM1)

❑ Output period: 200 µs

❑ Pins used in program: port 12 (output port)

Flowchart


START
- Initialize Stack Pointer
- Set main clock to fastest speed
- Set 8-bit timer count clock to fx=1.25 MHz
- Set compare register 0 (CR10) to 250
to get 200 µs interval time
- Set T0C1 to output disable for TM1
- Unmask 8-bit timer 1 interrpt (TMMK1)
- Set TMC1 to timer operation enable and
8-bit timer mode
- Set port 12 to output mode
- Set real time output mode, 8 bits x 1
channel, and trigger by INTTM1
Load 55h in real-time port output latch
Enable Master Interrupt
Idle
8-bit Timer 1 ISR
Toggle port latch
RETI

Assembly Language Program

```
;**************************************************************************
; Date:       08/23/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;             - Real-time output mode:  8-bit x 1 channel
;             - Output trigger:         8-bit timer 1 (TM1)
;             - Output data pattern:    AAh and 55h by turn
;             - Output frequency:       5 kHz (200 μs)
;*************************************************************************

;========================================
;       Specify Interrupt Vectors       =
;========================================
Res_Vec  CSEG AT 0000h                ; Set main program start vector
         DW        Start
         ORG       0024h              ; Set interrupt vector
         DW        TM1_ISR            ; for 8-bit timer 1

;========================================
;       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       TCL1,#07h          ; Select counter clock to fx = 1.25 MHz
         MOV       CR10,#250          ; Set compare register to 250 (200 μs)
         MOV       TOC1,#00h          ; Disable timer output function
         MOV       TMC1,#01h          ; Set timer 1 operation enable
         CLR1      TMMK1              ; Unmask 8-bit timer interrupt
         MOV       PM12,#00h          ; Set port 12 to output mode
         MOV       RTPM,#0FFh         ; Set port 12 to real time output mode
         MOV       RTPC,#02h          ; Set real-time output mode:
                                      ;   - 8-bits x 1 channel
                                      ;   - Timer 1 as trigger input
         MOV       RTBL,#55h          ; Load value into real-time output port latch
         EI                           ; Enable interrupts

Loop:    BR        Loop               ; CPU idle

;========================================
;=      8-bit timer ISR                 =
;========================================
ISR      CSEG
TM1_ISR:
         MOV       A, RTBL            ; Read back the port latch
         XOR       A, #0FFh           ; Toggle all bits
         MOV       RTBL, A            ; Write back into port latch
         RETI                         ; Return from interrupt

         END
```

C Language Program

```
/***************************************************************************
; Date:       08/23/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;             - Real-time output mode:  8-bit x 1 channel
;             - Output trigger:         8-bit timer 1 (TM1)
;             - Output data pattern:           AAh and 55h by turn
;             - Output frequency:    5 kHz (200 µs)
;*************************************************************************/

/* 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 DI        /* key word for DI instruction in C code */
#pragma EI        /* key word for EI instruction in C code */

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

#pragma interrupt INTTM1 TM1_ISR     /* TM1 interval timer vector */

/*======================================
;       Main Program                   =
;=====================================*/
void main(void)
{
         OSMS = 0x01;                /* Don't use scaler */
         PCC = 0x00;                 /* Main system clock at fastest setting */
         TCL1 = 0x07;                /* Select counter clock to fx = 1.25 MHz */
         CR10 = 250;                 /* Set compare register to 200 µs interval */
         TOC1 = 0x00;                /* Disable timer output function */
         TMC1 = 0x01;                /* Set timer 1 operation enable */
         TMMK1= 0;                   /* Unmask 8-bit timer 1 interrupt */

         PM12 = 0x00;                /* Set port 12 to output mode */
         RTPM = 0xFF;                /* Set port 12 to real time port mode */
         RTPC = 0x02;                /* Set real-time output mode
                                               - 8 bits x 1 channel
                                               - Timer 1 as trigger input */
         RTBL = 0x55;                /* Load value into real-time output latch */
         EI();                       /* Enable interrupts */

         while(1);                   /* loop here */
}                                    /* end of function main() */

/*=======================================
;       8-bit timer 1 ISR               =
;======================================*/
void TM1_ISR(void)
{
         RTBL ^= 0xFF;               /* Toggle each port bit */

}
```

NEC

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.