NEC

D/A Converter in Real-Time Output Mode

On-Chip Peripheral Program Example August 1999

Description

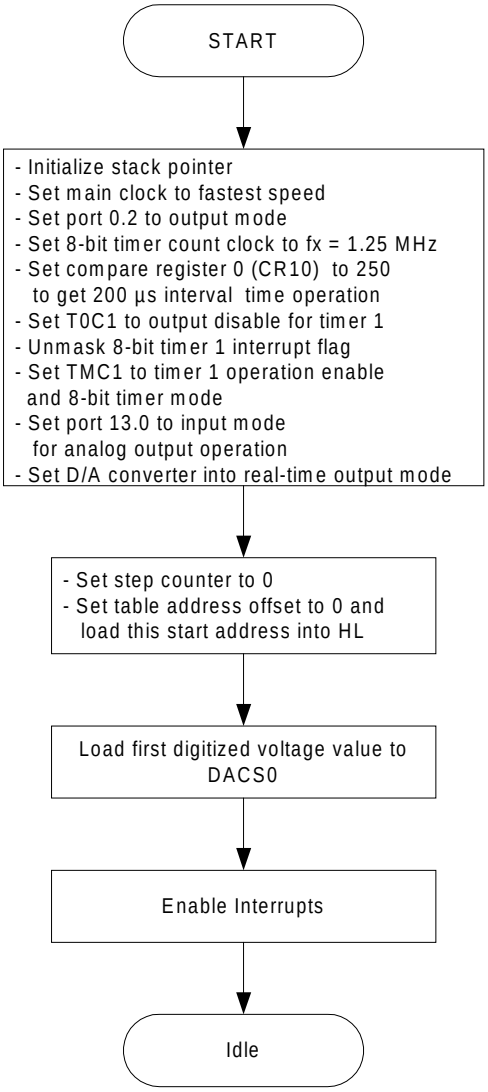
The 8-bit D/A converter in the μPD7805x/78005x subseries can be used in the normal output mode or the real-time output mode.

This program demonstrates the D/A converter in real-time output mode. Every 200 μs, an interrupt toggles port 0.2 and outputs a voltage step at pin ANO0/P130. The 0-, 1-, 2-, 3-, 4-, and 5-volt steps are predefined in a lookup table. After reaching the 5-volt step, the program starts outputting from 0 volts again.

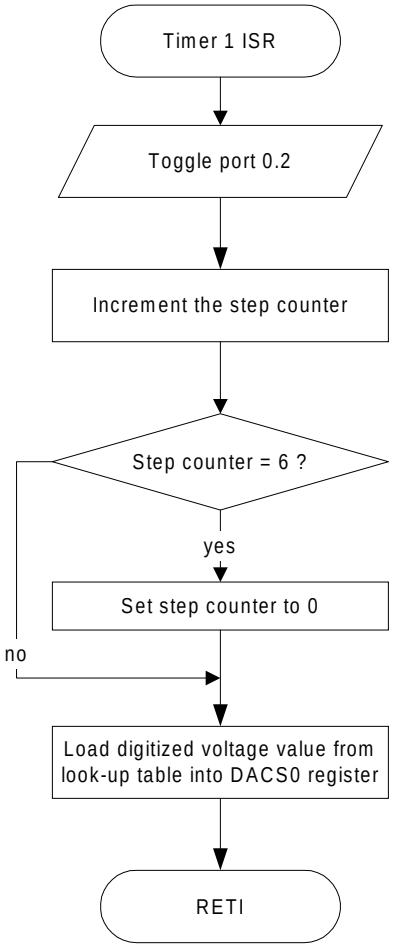
Program Specifications

- D/A converter channel 0 in real-time mode
- D/A output triggered every 200 μs by timer 1 interrupt
- D/A outputs analog voltages form 0 to 5 volts in 1-volt steps
- Pins used in program:
  - P02/INTP2: toggles every 200 μs
  - ANO0/P130: outputs analog voltage from D/A converter

Flowchart


START
- Initialize stack pointer
- Set main clock to fastest speed
- Set port 0.2 to output mode
- Set 8-bit timer count clock to fx = 1.25 MHz
- Set compare register 0 (CR10) to 250
to get 200 µs interval time operation
- Set T0C1 to output disable for timer 1
- Unmask 8-bit timer 1 interrupt flag
- Set TMC1 to timer 1 operation enable
and 8-bit timer mode
- Set port 13.0 to input mode
for analog output operation
- Set D/A converter into real-time output mode
- Set step counter to 0
- Set table address offset to 0 and
load this start address into HL
Load first digitized voltage value to
DACS0
Enable Interrupts
Idle

Flowchart ISR



Assembly Language Program

```
;*************************************************************************
; Date:       08/27/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;             - 8-bit D/A channel 0 (ANO0)
;             - Real-time output operation mode
;             - Timer 1 as output trigger
;             - Output ladder voltage:         0 V, 1 V, 2 V, 3 V, 4 V, 5 V
;             - Output ladder frequency:       5 kHz (200 µs steps)
;             - Reference voltage:             5 V (must be attached to Vref1 pin)
;             - Output port:                   Port 0.2 toggles every 200 µs
;*************************************************************************

;=======================================
;=      Digitized voltage values        =
;=======================================

VoltageTable:      db 0 * 255 /Vref1           ; 0 volt
                   db 1 * 255 /Vref1           ; 1 volt
                   db 2 * 255 /Vref1           ; 2 volt
                   db 3 * 255 /Vref1           ; 3 volt
                   db 4 * 255 /Vref1           ; 4 volt
                   db 5 * 255 /Vref1           ; 5 volt

;=======================================
;=      Constants/Variables             =
;=======================================

Vref1           equ         5                  ; Reference voltage value
TableSize       equ         $ - VoltageTable

;=======================================
;=      Data Segment                     =
;=======================================

Data               DSEG     saddr
StepCounter:       ds       1         ; Voltage step counter

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

Res_Vec   CSEG AT 0000h               ; Set main program start vector
          DW       Start
Tmr_Vec   CSEG AT 0024h               ; Set interrupt vector for 8-bit timer 1
          DW       TM1_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
          CLR1      P0.2               ; Latch port 0.2 low
          CLR1      PM0.2              ; Set port 0.2 to output mode
          MOV       TCL1,#07h          ; Select counter clock to fx = 1.25 MHz
          MOV       CR10,#250          ; Set Compare register to 250
                                       ;  200 µs interval
          MOV       TOC1,#00h          ; Disable output function
          MOV       TMC1,#01h          ; Set timer 1 operation enable
                                       ;  and 8-bit timer mode
          CLR1      TMMK1              ; Unmask the 8-bit timer 1 interrupt
          CLR1      PM13.1             ; Set port 13.1 to output mode
          CLR1      P13.1              ; Latch port 13.1 to low
          SET1      PM13.0             ; Set port 13.0 to input mode
                                       ;  (analog output)
          MOV       DAM,#11h           ; D/A channel 0 conversion enable
                                       ;  in real-time output mode
          MOV       StepCounter,#0     ; Set voltage output step counter to 0
          MOV       C,#0               ; Set address offset to 0
          MOVW      HL,#VoltageTable   ; Load look-up table start address
          MOV       A,[HL+C]           ; Read digital value of the voltage
          MOV       DACS0,A            ;  and write value to DAC register
          EI                           ; Enable interrupts
Loop:     BR        Loop               ; Endless loop
;=======================================
;=      8-bit timer 1 ISR               =
;=======================================
ISR       CSEG
TM1_ISR:
          XOR       P0, #04h                    ; Toggle port 0.2
          INC       StepCounter                 ; Increment step counter
          CMP       StepCounter,#TableSize      ; Compare step counter with
                                                ;   size of table
          BNZ       $ISR10                      ; Branch if step counter is not 6
          MOV       StepCounter,#0              ; Clear step counter
ISR10:    MOV       A,StepCounter               ; Load step counter to A register
          MOV       C,A                         ;  to store in C register
          MOVW      HL,#VoltageTable            ; Load look-up table start address
          MOV       A,[HL+C]                    ; Read digital value of the voltage
          MOV       DACS0,A                     ; and write value to DAC register
          RETI                                  ; Return from interrupt
          END
```

C Language Program

```
/************************************************************************
; Date:       08/27/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;             - 8-bit D/A channel 0 (ANO0)
;             - Real-time output operation mode
;             - Timer 1 as output trigger
;             - Output ladder voltage:         0 V, 1 V, 2 V, 3 V, 4 V, 5 V
;             - Output ladder frequency:       5 kHz (200 µs steps)
;             - Reference voltage:             5 V (must be attached to Vref1 pin)
;             - Output port:                   Port 0.2 toggles every 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 EI         /* key word for EI instruction in C code */

/*====================================
;=      Specify Interrupt Vectors      =
;====================================*/
#pragma interrupt INTTM1 TM1_ISR              /* TM1 interval timer vector */

/*====================================
;=      Constants/Variables            =
;====================================*/
#define  TRUE               1
#define  FALSE              0
#define  Vref1              5                 /* Reference voltage */
#define  Volts(n)           255 * n / Vref1
unsigned char               StepCounter;      /* Voltage output step index */

/*====================================
;=      Digitized voltage values       =
;====================================*/
const unsigned char VoltageTable[] =
  {
         Volts(0),                            /* 0 Volt */
         Volts(1),                            /* 1 Volt */
         Volts(2),                            /* 2 Volt */
         Volts(3),                            /* 3 Volt */
         Volts(4),                            /* 4 Volt */
         Volts(5)                             /* 5 Volt */
  };

/*====================================
;=      Main Program                   =
;====================================*/
void main(void)
{
         OSMS = 0x01;               /* Don't use scaler */
         PCC = 0x00;                /* Main system clock at fastest setting */
         P0.2 = 0;                  /* Latch port 0.2 low */
         PM0.2 = 0;                 /* Set port 0.2 to output  mode */
         TCL1 = 0x07;               /* Select counter clock to fx = 1.25 MHz */
```

```
        CR10 = 250;                 /* Set compare register to 250 for
                                        200 µs interval */
        TOC1 = 0x00;                /* Disable output function */
        TMC1 = 0x01;                /* Set timer 1 operation enable
                                              and 8-bit timer mode */
        TMMK1= 0;                   /* Unmask the 8-bit timer 1 interrupt */
        PM13.1 = 0;                 /* Set port 13.1 to output mode */
        P13.1  = 0;                 /* Latch port 13.1 to low */
        PM13.0 = 1;                 /* Set port 13.0 to input mode
                                              (analog output) */
        DAM = 0x11;                 /* D/A channel 0  conversion enable
                                        in real time output mode */
        StepCounter = 0;            /* Set voltage output step counter to 0 */
        DACS0 = VoltageTable[0];    /* Write digital value to DAC register */
        EI();                       /* Enable interrupts */
        while(TRUE);                /* Endless loop */
 }                                  /* End of function main */

/*=====================================
;=      8-bit timer 1 ISR              =
;=====================================*/
void TM1_ISR(void)
{
        P0 ^= 0x04;                                 /* Toggle port 0.2 */
        StepCounter++;                              /* Increment step counter */
        if(StepCounter == sizeof VoltageTable )     /* Step counter equal 6 ? */
                StepCounter = 0;                    /* Clear step counter */
        DACS0 = VoltageTable[StepCounter];          /* Write value to DAC register */
}
```

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.

©1999 NEC Electronics Inc./Printed in U.S.A.

50886