NEC

D/A Converter in Normal Output Mode

On-Chip Peripheral Program Example August 1999

Description

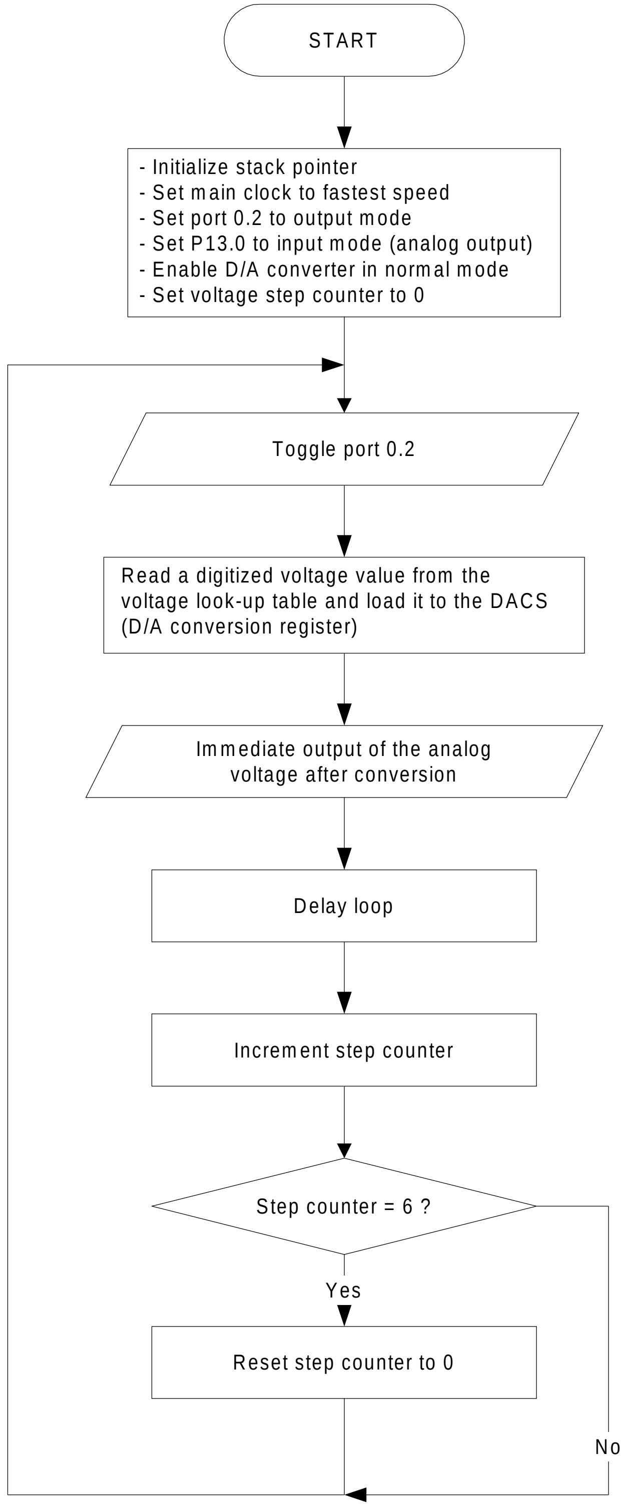
The 8-bit D/A converter in the µPD7805x/78005x subseries can be used in normal output mode or real-time output mode.

This program demonstrates the D/A converter in normal mode, where the conversion result is output immediately after the conversion is completed. The program outputs a voltage ladder with 0-, 1-, 2-, 3-, 4- and 5-volt steps. After reaching the 5-volt step, the program starts outputting from 0 volts again. Each step is 36 µs (Assembly program) or 170 µs (C program) long.

Program Specifications

- D/A converter channel 0 in normal output mode
- D/A output triggered after conversion is completed
- D/A outputs analog voltages from 0 to 5 volts in 1-volt steps
- Pins used in program:
  - P02/INTP2: toggles every time, a new analog voltage is output
  - ANO0/P130: outputs the 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 P13.0 to input mode (analog output)
- Enable D/A converter in normal mode
- Set voltage step counter to 0
Toggle port 0.2
Read a digitized voltage value from the voltage look-up table and load it to the DACS (D/A conversion register)
Immediate output of the analog voltage after conversion
Delay loop
Increment step counter
Step counter = 6 ?
Yes
Reset step counter to 0
No

Assembly Language Program

```
;**************************************************************************
; Date:       08/31/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;             - 8-bit D/A channel 0
;             - Normal output operation mode (Writing to DACS0 register)
;             - Output ladder voltage:         0 V, 1 V, 2 V, 3 V, 4 V, 5 V, 0 V…
;             - Output ladder frequency:       13.9 kHz (each step is 36 µs)
;             - Reference voltages:            5 V (must be attached to Vref1 pin)
;             - Port 0.2:                      Toggles before each conversion
;*************************************************************************

;========================================
;=      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              equu     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

;========================================
;=      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 to low
         CLR1     PM0.2                        ; Set port 0.2 to output mode
         CLR1     P13.1                        ; Latch port 13.1 to low
         CLR1     PM13.1                       ; Set port 13.1 to output
         SET1     PM13.0                       ; Set port 13.0 to input mode
                                               ;  (analog output)
         MOV      DAM,#01H                     ; Enable D/A conversion in normal mode
         MOV      StepCounter,#0               ; Set voltage output step counter to 0

Loop:    XOR      P0, #04h                     ; Toggle port 0.2
         MOV      A,StepCounter                ; Load step counter value to C register
         MOV      C,A
         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 DACS0 register
         MOV      B, #10h                      ; Load 16d in loop counter
DELAY:   NOP                                   ; Delay sets the step width in µs
         DBNZ     B,$DELAY                     ; Branch back, if B is not 0
         INC      StepCounter                  ; Increment step counter
         CMP      StepCounter,#TableSize       ; Compare step counter with size of table
         BNZ      $Loop                        ; Branch back if step counter is not 4
         MOV      StepCounter,#0               ; Clear step counter
         BR       Loop                         ; Branch back
         END
```

C Language Program

```
/***************************************************************************
; Date:       08/31/1999
;
; Parameters: - fastest CPU clock
;               (fx = 5 MHz; 1 CPU clock cycle = 200 ns)
;             - 8-bit D/A channel 0
;             - Normal output operation mode (Writing to DACS0 register)
;             - Output ladder voltage:         0 V, 1 V, 2 V, 3 V, 4 V, 5 V, 0 V…
;             - Output ladder frequency:       2.9 kHz (each step is 170 µs)
;             - Reference voltages:            5 V (must be attached to Vref1 pin)
;             - Port 0.2:                      Toggles before each conversion
;************************************************************************/
/* 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 NOP        /* key word for NOP instruction in C code */

/*======================================
;=      Constants/Variables            =
;=====================================*/
#define TRUE       1
#define FALSE      0
#define Vref1      5                           /* Reference voltage value */
#define Volts(n)   255 * n / Vref1
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 */
};
unsigned char StepCounter;                     /* Voltage output step counter */
unsigned char I;                               /* Variable for delay loop */

/*=======================================
;=      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 */
         P13.1  = 0;                  /* Latch  port 13.1 to low */
         PM13.1 = 0;                  /* Set port 13.1 to output mode */
         PM13.0 = 1;                  /* Set port 13.0 to input mode (analog output) */
         DAM = 0x01;                  /* Enable D/A conversion in normal mode */
         StepCounter = 0;             /* Set voltage output step counter to 0 */
         while(TRUE)                                        /* Loop here */
         {
            P0 ^= 0x04;                                     /* Toggle port 0.2 */
            DACS0 = VoltageTable[StepCounter];              /* Output current step */
            for(i=1; i<20; i++);                            /* Delay loop for conversion
```

```
    time */
            StepCounter++;                             /* increment step counter */
            if(StepCounter == sizeof VoltageTable )    /* Test step counter */
                StepCounter = 0;                       /* clear step counter */
         }                                             /* end of while(TRUE) loop */
}                                                      /* 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.

50885