Using the ADC and PWM of the 83C752/87C752 AN428

The Philips 83C752/87C752 is a single-chip control-oriented microcontroller. It is an 80C51 derivative, having the same basic architecture and powerful instruction set in a small 28-pin package. As “add-on” functions to a standard microcontroller, it offers an I^2C small area network port, a five-channel multiplexed 8-bit analog-to-digital converter (ADC), and a pulse width modulation (PWM) output. The part is essentially the popular 8XC751 with the addition of the ADC and the PWM output.

There are many control applications for which this microcontroller can provide an almost-complete, low-cost solution. The A/D converter can monitor analog voltages of up to five sources. The PWM output can be used to generate an analog control voltage with the addition of a simple integrator circuit. Another potential use for the PWM output is as a driver of power-switching circuits for DC motor speed control.

The analog-to-digital converter has 8-bit resolution, and the conversion takes 40 machine cycles. A multiplexer selects one out of five input pins. The operation of the A/D converter and the multiplexer is controlled by the ADCON register.

The repetition frequency of the PWM output pulses is determined by an 8-bit prescaler, programmed at register PWMP. The duty cycle of these pulses is determined by the contents of a compare register, PWM. In order to implement the pulse width modulator, the prescaler output drives an 8-bit counter. When the counter value matches the contents of the compare (PWM) register, the PWM output is set high, and when the counter reaches zero, the output is set low. The counter is modulo 255, so the duty cycle generated will be the PWM contents multiplied by 1/255.

The enclosed listing demonstrates usage of the A/D converter and the PWM. In order to communicate with the outside world, the program sends messages on a software-driven RS-232 port. The routines for sending messages via a software-controlled serial port can be quite useful, and for further discussion on those, please refer to Application Note 423: “Software Driven Serial Communication Routines for the 83C751 and 83C752 Microcontrollers.”

Bit 5 of port 1 is used for the RS-232 communications, and in order to hook the microcontroller to a terminal, a buffer (e.g., MC1488) is needed. Timer 0 is used as the baud rate generator, where the timer value is defined by the symbol BaudVal. The programmed value will generate a 9600 baud rate with a 16MHz crystal.

The program, after initialization and sending a message to the terminal, scans all five A/D channel inputs and outputs the voltage read on the serial port, as a hexadecimal value. Circuit operation can be verified by comparing channel voltages with the reading at the terminal. The program follows with an infinite loop in which channel 0 of the A/D converter is read, and its value is used to program the PWM. A simple verification of the duty cycle can be done with a voltmeter: since it acts as an integrator, its reading will be proportional to the duty cycle. Reading of a voltmeter on the PWM output should be proportional to the channel 0 input voltage. If the analog reference voltage AV_{CC}, which is full-scale of the A/D measurement, is set to be exactly as V_{CC}, the PWM output will track channel 0 within about 20mV.

```
DEMO752C                    87C752 A/D and PWM Demonstration                    12/03/90    PAGE 1

                     1 ;
                     2
                     3    ;*************************************************************************
                     4
                     5    ;                 87C752 A/D and PWM Demonstration Program
                     6
                     7    ;  This program first reads all five A/D channels and outputs the values in
                     8    ;  hexadecimal as RS-232 data. Next, the PWM output is set to reflect the
                     9    ;  value on A/D channel 0, and again outputs the A/D value to RS-232. Note
                    10    ;  that the A/D value is inverted before being moved to the PWM compare
                    11    ;  register in order to compensate for the inversion on the PWM output pin.
                    12    ;  This process is repeated continuously.
                    13
                    14    ;  Thus, a voltage may be applied to ADC0 (P1.0, pin 13) to vary the PWM pulse
                    15    ;  width.  A simple test of this function is to measure the voltage on ADC0
                    16    ;  and PWM with a voltmeter.  A typical voltmeter will integrate the waveform
                    17    ;  on the PWM output and show a voltage within about 20mV of that on ADC0.
                    18
                    19    ;  The RS-232 output appears on Port 1 pin 5, which must be buffered with an
                    20    ;  MC1488 or perhaps a MAX232 chip prior to being connected to a terminal.
                    21    ;  The transmission rate will  be 9600 baud when the 87C752 is operated from
                    22    ;  16MHz crystal.
                    23
                    24    ;*************************************************************************
                    25
                    26    $Title(87C752 A/D and PWM Demonstration)
                    27    $Date(12/03/90)
                    28    $MOD752
                    29
                    30    ;*************************************************************************
                    31
  FF75              32    BaudVal  EQU       -139              ;Timer value for 9600 baud @ 16 MHz.
                    33                                         ;(one bit cell time)
                    34
  0010              35    XmtDat   DATA      10h               ;Data for RS-232 transmit routine.
  0012              36    BitCnt   DATA      12h               ;RS-232 transmit bit count.
  0013              37    PWMVal   DATA      13h               ;Holds next value for updating the PWM.
  0014              38    ADVal    DATA      14h               ;Holds last A/D conversion result.
                    39
  0020              40    Flags    DATA      20h
  0000              41    TxFlag   BIT       Flags.0           ;Transmit-in-progress flag.
  0001              42    ADFlag   BIT       Flags.1           ;Indicates A/D conversion complete.
                    43
  0095              44    TxD      BIT       P1.5              ;Port bit for RS-232 transmit.
                    45
                    46    ;*************************************************************************
                    47
                    48    ; Interrupt Vectors
                    49
0000                50             ORG       0                 ;Reset vector.
0000 0135           51             AJMP      Reset
                    52
000B                53             ORG       0BH               ;Timer 0 interrupt.
000B 01C5           54             AJMP      Timr0             ;(used as a baud rate generator)
                    55
002B                56             ORG       2Bh               ;A/D conversion complete interrupt.
002B 0199           57             AJMP      ADInt
                    58
```

```
DEMO752C                    87C752 A/D and PWM Demonstration                    12/03/90    PAGE 2


0033                 59             ORG       33h                 ;PWM interrupt.
0033 01A3            60             AJMP      PWMInt
                     61
                     62
                     63     ;*************************************************************************
                     64
0035 758130          65     Reset:  MOV       SP,#30h
0038 752000          66             MOV       Flags,#0            ;Clear RS-232 flags.
003B 758800          67             MOV       TCON,#00h           ;Set up timer controls.
003E 75A882          68             MOV       IE,#82h             ;Enable timer 0 interrupt.
                     69
0041 90011B          70             MOV       DPTR,#Msg1          ;Point to message string.
0044 310A            71             ACALL     Mess                ;Send message.
                     72
0046 7900            73             MOV       R1,#0               ;Start with A/D channel 0.
0048 E9              74     Loop1:  MOV       A,R1
0049 118D            75             ACALL     ADConv              ;Start A/D conversion.
004B FA              76             MOV       R2,A
                     77
004C 900152          78             MOV       DPTR,#Msg2          ;Point to message string.
004F 310A            79             ACALL     Mess                ;Send message.
0051 E9              80             MOV       A,R1
0052 11EC            81             ACALL     PrByte              ;Print channel #.
0054 900161          82             MOV       DPTR,#Msg3          ;Point to message string.
0057 310A            83             ACALL     Mess                ;Send message.
                     84
0059 EA              85             MOV       A,R2
005A 11EC            86             ACALL     PrByte              ;Print A/D value.
005C 09              87             INC       R1                  ;Advance R1 value.
005D B905E8          88             CJNE      R1,#5,Loop1
0060 90014F          89             MOV       DPTR,#CRLF          ;Point to message string.
0063 310A            90             ACALL     Mess
                     91
                     92     ; Now use A/D channel 0 value to control the PWM.
                     93
0065 758FFF          94             MOV       PWMP,#0FFh          ;Set PWM slow frequency.
0068 758E00          95             MOV       PWCM,#0             ;Set initial PWM value.
006B 751300          96             MOV       PWMVal,#0           ;Default starting value for the PWM.
006E 75FE01          97             MOV       PWENA,#1            ;Start PWM
0071 75A8CA          98             MOV       IE,#0CAh           ;Now enable the A/D and PWM interrupts.
                     99
0074 7400           100     Loop2:  MOV       A,#0                ;Read A/D channel 0.
0076 1186           101             ACALL     ADStart             ;Start A/D conversion.
0078 3001FD         102             JNB       ADFlag,$            ;Wait for A/D conversion complete.
007B E514           103             MOV       A,ADVal             ;Get A/D result to print.
007D 11EC           104             ACALL     PrByte              ;Print PWM value.
007F 900165         105             MOV       DPTR,#Msg4          ;Point to message string.
0082 310A           106             ACALL     Mess
0084 80EE           107             SJMP      Loop2
                    108
                    109
                    110     ; A/D Conversion Routines.
                    111     ;   The following shows two ways to use the A/D. Both routines are used by
                    112     ;   different portions of the sample program.
                    113
                    114     ;   Method 1: This version of the routine starts the conversion and then
                    115     ;   returns. The mainline program can detect when the conversion is
                    116     ;   complete by checking the A/D conversion complete flag (ADFlag) which is
```

```
DEMO752C                    87C752 A/D and PWM Demonstration                    12/03/90    PAGE 3


                    117    ;   set by the A/D interrupt service routine. A/D data must be read by the
                    118    ;   calling routine.
                    119
0086 C201           120    ADStart: CLR       ADFlag              ;Clear A/D conversion complete flag.
0088 4428           121             ORL       A,#28h              ;Add control bits to channel #.
008A F5A0           122             MOV       ADCON,A             ;Start conversion.
008C 22             123             RET
                    124
                    125
                    126    ;   Method 2: This is an alternative version of the A/D routine which
                    127    ;   starts the conversion and then waits for it to complete before
                    128    ;   returning. A/D data is returned in the ACC.
                    129
008D 4428           130    ADConv:  ORL       A,#28h              ;Add control bits to channel #.
008F F5A0           131             MOV       ADCON,A             ;Start conversion.
0091 E5A0           132    ADC1:    MOV       A,ADCON
0093 30E4FB         133             JNB       ACC.4,ADC1          ;Wait for conversion complete.
0096 E584           134             MOV       A,ADAT              ;Read A/D.
0098 22             135             RET
                    136
                    137
                    138    ; A/D interrupt service routine.
                    139
0099 E584           140    ADInt:   MOV       A,ADAT              ;Read A/D data.
009B F514           141             MOV       ADVal,A             ;Save A/D data for print routine.
009D F4             142             CPL       A                   ;Complement the value for the PWM.
009E F513           143             MOV       PWMVal,A            ;Set new value for PWM update.
00A0 D201           144             SETB      ADFlag              ;Tell main that new A/D data is ready.
00A2 32             145             RETI
                    146
                    147
                    148    ; PWM interrupt service routine allows updating the PWM synchronously.
                    149
00A3 85138E         150    PWMInt:  MOV       PWCM,PWMVal         ;Update PWM duty cycle.
00A6 32             151             RETI
                    152
                    153
                    154    ; Send a byte out RS-232 and wait for completion before returning.
                    155
00A7 11AD           156    XmtByte: ACALL     RSXmt               ;Send ACC to RS-232 output.
00A9 2000FD         157             JB        TxFlag,$            ;Wait for transmit complete.
00AC 22             158             RET
                    159
                    160
                    161    ; Begin RS-232 transmit.
                    162
00AD F510           163    RSXmt:   MOV       XmtDat,A            ;Save data to be transmitted.
00AF 75120A         164             MOV       BitCnt,#10          ;Set bit count.
00B2 758CFF         165             MOV       TH,#High BaudVal    ;Set timer for baud rate.
00B5 758A75         166             MOV       TL,#Low BaudVal
00B8 758DFF         167             MOV       RTH,#High BaudVal   ;Also set timer reload value.
00BB 758B75         168             MOV       RTL,#Low BaudVal
00BE D28C           169             SETB      TR                  ;Start timer.
00C0 C295           170             CLR       TxD                 ;Begin start bit.
00C2 D200           171             SETB      TxFlag              ;Set transmit-in-progress flag.
00C4 22             172             RET
                    173
                    174
```

```
DEMO752C                    87C752 A/D and PWM Demonstration                    12/03/90    PAGE 4


                  175    ; Timer 0 timeout: RS-232 receive bit or transmit bit.
                  176
00C5 C0E0         177    Timr0:   PUSH      ACC
00C7 C0D0         178             PUSH      PSW
00C9 200007       179             JB        TxFlag,TxBit        ;Is this a transmit timer interrupt?
00CC C28C         180    T0Ex1:   CLR       TR                  ;Stop timer.
00CE D0D0         181    T0Ex2:   POP       PSW
00D0 D0E0         182             POP       ACC
00D2 32           183             RETI
                  184
                  185
                  186    ; RS-232 transmit bit routine.
                  187
00D3 D51204       188    TxBit:   DJNZ      BitCnt,TxBusy       ;Decrement bit count, test for done.
00D6 C200         189             CLR       TxFlag              ;End of stop bit, release timer.
00D8 80F2         190             SJMP      T0Ex1               ;Stop timer and exit.
                  191
00DA E512         192    TxBusy:  MOV       A,BitCnt            ;Get bit count.
00DC B40104       193             CJNE      A,#1,TxNext         ;Is this a stop bit?
00DF D295         194             SETB      TxD                 ;Set stop bit.
00E1 80EB         195             SJMP      T0Ex2               ;Exit.
                  196
00E3 E510         197    TxNext:  MOV       A,XmtDat            ;Get data.
00E5 13           198             RRC       A                   ;Advance to next bit.
00E6 F510         199             MOV       XmtDat,A
00E8 9295         200             MOV       TxD,C               ;Send data bit.
00EA 80E2         201             SJMP      T0Ex2               ;Exit.
                  202
                  203
                  204    ; Print byte routine: print ACC contents as ASCII hexadecimal.
                  205
00EC C0E0         206    PrByte:  PUSH   ACC
00EE C4           207             SWAP   A
00EF 11FA         208             ACALL  HexAsc
00F1 11A7         209             ACALL  XmtByte
00F3 D0E0         210             POP    ACC
00F5 11FA         211             ACALL  HexAsc            ;Print nibble in ACC as ASCII hex.
00F7 11A7         212             ACALL  XmtByte
00F9 22           213             RET
                  214
                  215
                  216    ; Hexadecimal to ASCII conversion routine.
                  217
00FA 540F         218    HexAsc:  ANL    A,#0FH            ;Convert a nibble to ASCII hex.
00FC 30E308       219             JNB    ACC.3,NoAdj
00FF 20E203       220             JB     ACC.2,Adj
0102 30E102       221             JNB    ACC.1,NoAdj
0105 2407         222    Adj:     ADD    A,#07H
0107 2430         223    NoAdj:   ADD    A,#30H
0109 22           224             RET
                  225
                  226
                  227    ; Message string transmit routine.
                  228
010A C0E0         229    Mess:    PUSH      ACC
010C 7800         230             MOV       R0,#0               ;R0 is character pointer (string
010E E8           231    Mesl:    MOV       A,R0                ; length is limited to 256 bytes).
010F 93           232             MOVC      A,@A+DPTR           ;Get byte to send.
```

```
 DEMO752C                    87C752 A/D and PWM Demonstration                    12/03/90    PAGE 5

0110 B40003        233             CJNE      A,#0,Send           ;End of string is indicated by a 0.
0113 D0E0          234             POP       ACC
0115 22            235             RET
                   236
0116 11A7          237     Send:   ACALL     XmtByte             ;Send a character.
0118 08            238             INC       R0                  ;Next character.
0119 80F3          239             SJMP      Mesl
                   240
011B 0D0A          241     Msg1:   DB        0Dh, 0Ah,
011D 54686973      242             DB        'This is a demonstration of the 87C752 A/D and PWM.'
0121 20697320
0125 61206465
0129 6D6F6E73
012D 74726174
0131 696F6E20
0135 6F662074
0139 68652038
013D 37433735
0141 3220412F
0145 4420616E
0149 64205057
014D 4D2E
014F 0D0A00        243     CRLF:   DB        0Dh, 0Ah, 0
                   244
0152 0D0A412F      245     Msg2:   DB        0Dh, 0Ah, 'A/D Channel ', 0
0156 44204368
015A 616E6E65
015E 6C2000
                   246
0161 203D2000      247     Msg3:   DB        ' = ', 0
                   248
0165 202000        249     Msg4:   DB        '  ', 0
                   250
                   251             END

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

```
 DEMO752C                    87C752 A/D and PWM Demonstration                    12/03/90    PAGE 6

ACC. . . . . . . . . . . . . . . .  D ADDR  00E0H  PREDEFINED
ADAT . . . . . . . . . . . . . . .  D ADDR  0084H  PREDEFINED
ADC1 . . . . . . . . . . . . . . .  C ADDR  0091H
ADCON. . . . . . . . . . . . . . .  D ADDR  00A0H  PREDEFINED
ADCONV . . . . . . . . . . . . . .  C ADDR  008DH
ADFLAG . . . . . . . . . . . . . .  B ADDR  0001H
ADINT. . . . . . . . . . . . . . .  C ADDR  0099H
ADJ. . . . . . . . . . . . . . . .  C ADDR  0105H
ADSTART. . . . . . . . . . . . . .  C ADDR  0086H
ADVAL. . . . . . . . . . . . . . .  D ADDR  0014H
BAUDVAL. . . . . . . . . . . . . .    NUMB  FF75H
BITCNT . . . . . . . . . . . . . .  D ADDR  0012H
CRLF . . . . . . . . . . . . . . .  C ADDR  014FH
FLAGS. . . . . . . . . . . . . . .  D ADDR  0020H
HEXASC . . . . . . . . . . . . . .  C ADDR  00FAH
IE . . . . . . . . . . . . . . . .  D ADDR  00A8H  PREDEFINED
LOOP1. . . . . . . . . . . . . . .  C ADDR  0048H
LOOP2. . . . . . . . . . . . . . .  C ADDR  0074H
MESL . . . . . . . . . . . . . . .  C ADDR  010EH
MESS . . . . . . . . . . . . . . .  C ADDR  010AH
MSG1 . . . . . . . . . . . . . . .  C ADDR  011BH
MSG2 . . . . . . . . . . . . . . .  C ADDR  0152H
MSG3 . . . . . . . . . . . . . . .  C ADDR  0161H
MSG4 . . . . . . . . . . . . . . .  C ADDR  0165H
NOADJ. . . . . . . . . . . . . . .  C ADDR  0107H
P1 . . . . . . . . . . . . . . . .  D ADDR  0090H  PREDEFINED
PRBYTE . . . . . . . . . . . . . .  C ADDR  00ECH
PSW. . . . . . . . . . . . . . . .  D ADDR  00D0H  PREDEFINED
PWCM . . . . . . . . . . . . . . .  D ADDR  008EH  PREDEFINED
PWENA. . . . . . . . . . . . . . .  D ADDR  00FEH  PREDEFINED
PWMINT . . . . . . . . . . . . . .  C ADDR  00A3H
PWMP . . . . . . . . . . . . . . .  D ADDR  008FH  PREDEFINED
PWMVAL . . . . . . . . . . . . . .  D ADDR  0013H
RESET. . . . . . . . . . . . . . .  C ADDR  0035H
RSXMT. . . . . . . . . . . . . . .  C ADDR  00ADH
RTH. . . . . . . . . . . . . . . .  D ADDR  008DH  PREDEFINED
RTL. . . . . . . . . . . . . . . .  D ADDR  008BH  PREDEFINED
SEND . . . . . . . . . . . . . . .  C ADDR  0116H
SP . . . . . . . . . . . . . . . .  D ADDR  0081H  PREDEFINED
T0EX1. . . . . . . . . . . . . . .  C ADDR  00CCH
T0EX2. . . . . . . . . . . . . . .  C ADDR  00CEH
TCON . . . . . . . . . . . . . . .  D ADDR  0088H  PREDEFINED
TH . . . . . . . . . . . . . . . .  D ADDR  008CH  PREDEFINED
TIMR0. . . . . . . . . . . . . . .  C ADDR  00C5H
TL . . . . . . . . . . . . . . . .  D ADDR  008AH  PREDEFINED
TR . . . . . . . . . . . . . . . .  B ADDR  008CH  PREDEFINED
TXBIT. . . . . . . . . . . . . . .  C ADDR  00D3H
TXBUSY . . . . . . . . . . . . . .  C ADDR  00DAH
TXD. . . . . . . . . . . . . . . .  B ADDR  0095H
TXFLAG . . . . . . . . . . . . . .  B ADDR  0000H
TXNEXT . . . . . . . . . . . . . .  C ADDR  00E3H
XMTBYTE. . . . . . . . . . . . . .  C ADDR  00A7H
XMTDAT . . . . . . . . . . . . . .  D ADDR  0010H
```