Interfacing the PD243X Alphanumeric Programmable Display™ with the SAB80515/SAB80535 Microcontroller

To Produce a Bidirectional, Speed Regulated Moving Message Display by Using the SAB80515/SAB80535's Timer 2 & 8-Bit Converter

Appnote 49

This application note introduces the user to one of the features of Timer 2 and A/D converter of the SAB 80515/535. Included in this application note is a description of both the software and hardware implementations of the SAS 80515/535 to use its Timer 2 and 8-bit A/D converter for the bidirectional, speed regulated moving message display. The program listing demonstrates how the Timer 2 and the 8-bit A/D converter of the SAB 80515/535 can be combined to generate time delays controlled by analog levels. The hardware circuitry shows an interface of the SAP 80515/535 with a simulated analog input, a 2 kbyte EPROM, and intelligent display chips of Siemens used in memory mapped 1/0 scheme.

The SAB 80515/535 microcontroller with on-chip A/D converter and a 16-bit Timer (Timer 2) with reload capability offers a solution which can be applied to a wide range of industrial applications. These applications vary from analog controlled digital delays to controlled frequency converters for pulse width modulation.

In the present application example, the above features of the SAB 80515/535 are used in conjunction to generate the software delays. The software delay results in varying the voltage level of the analog signal applied to the A/D converter of the SAB 80515/535.

A/D Converter

The SAB 80515/535 provides an 8-bit A/D converter with eight multiplexed analog input channels on-chip. In addition, the A/D converter has a sample and hold circuit and offers the feature of software programmable reference voltages. For the conversion, the method of successive approximation with a capacitor network is used.

Figure 1 shows a block diagram of the A/D converter. There are three user-accessible special function registers:

—ADCON (A/D converter control register)
—ADDAT (A/D converter data register)
—DAPR (D/A converter program register) for the programmable reference voltages.

Special function register ADCON is used to select one of the eight analog input channels to be converted, to specify a single or continuous conversion, and to check the status bit BSY which signals whether a conversion is in progress or not.

The special function register ADDAT holds the converted digital 8-bit data result. The data remains in ADDAT until it is overwritten by the next converted data. The new converted value will appear in ADDAT in the 15th machine cycle after a conversion has been started. ADDAT can be read and written to under software control. If the A/D converter of the SAB 80515/535 is not used, register ADDAT can be used as an additional general-purpose register.

The special function register DAPR is provided for programming the internal reference voltages IVAREF and IVAGND. In the present application DAPR holds a value of 00H. For this value of DAPR, IVAREF and IVAGND are the same as VAREF and VAGND respectively.

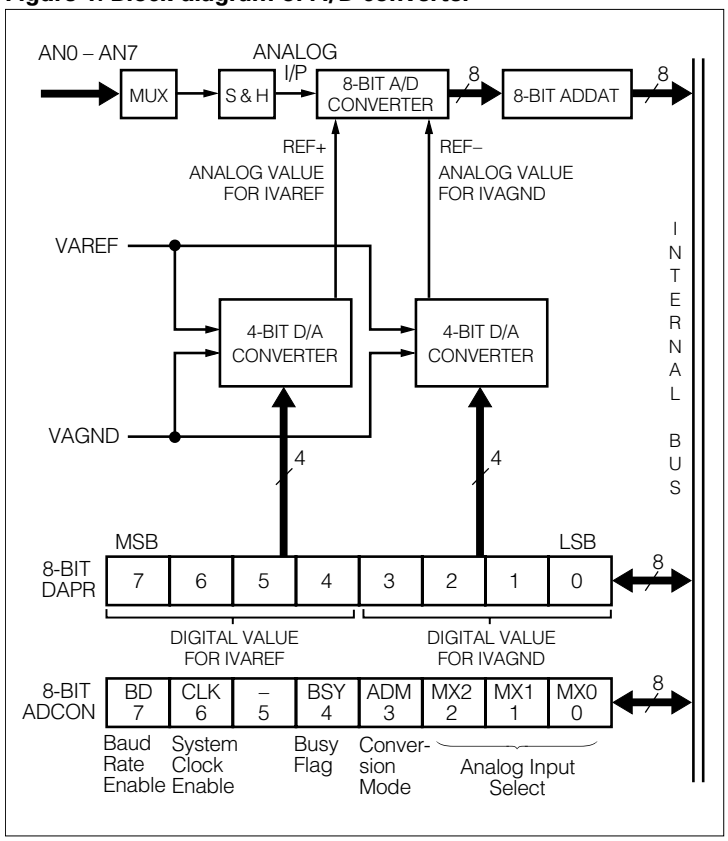
A/D Conversion

A conversion is started by writing to the special function register DAPR. A "Write-to-DAPR" will start a new conversion even if a conversion is currently in progress. The conversion begins with the next machine cycle. The busy flag BSY will be set in the same machine cycle as the "write-to-DAPR" operation occurs. If the value written to DAPR is 00H, meaning that no adjustment of the internal reference voltages is desired, the conversion needs 15 machine cycles to be completed. Thus, the conversion time is 15 µs for 12 MHz oscillator frequency.

After a conversion has been started by writing into the special function register DAPR, the analog voltage at the selected input channel is sampled for 5 machine cycles (5 µs at 12 MHz oscillator frequency), which will then be held at the sampled level for the rest of the conversion time.

The external analog source must be strong enough to source the current in order to load the sample & hold capacitance, being 25 pF, within those 5 machine cycles.

Figure 1. Block diagram of A/D converter



Conversion of the sampled analog voltage takes place between the 6th and 15th machine cycle after sampling has been completed. In the 15th machine cycle the converted result is moved to ADDAT.

Timer 2

The SAB 80515 has three 16-bit Timer/Counters: Timer 0, Timer 1 and Timer 2. These Timers can be configured to operate either as timers or event counters. Timer 2 is the time base of the programmable Timer/Counter Register Array (PTRA) unit. In addition to the operational modes "Timer" or "counter", Timer 2, being the time base for the PTRA unit, provides the features of:

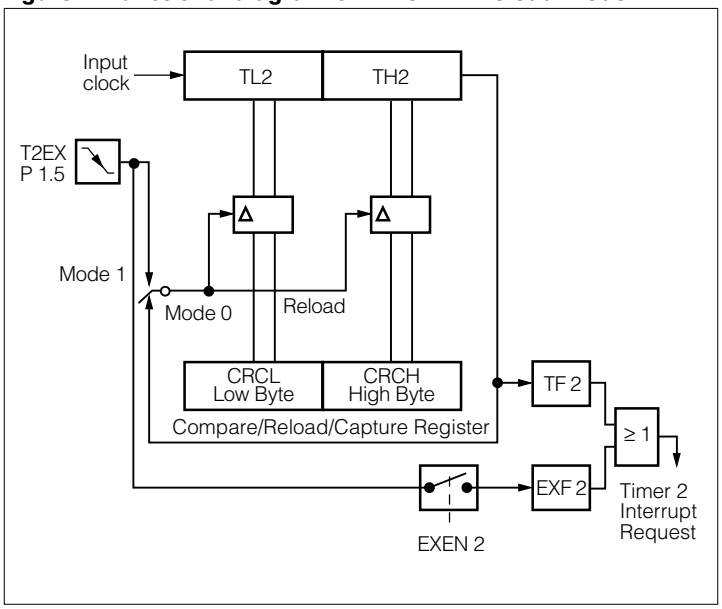
—16-bit reload

—16-bit compare

—16-bit capture

The reload mode of Timer 2 is used in this application to generate software delays. For explanation of the other modes please refer to the users' manual.

Figure 2. Functional diagram of Timer 2 in reload mode



Reload

The reload mode for Timer 2 is selected by bits T2R0 and T2R1 in special function register T2CON as illustrated in Table 1. In mode 0, when Timer 2 rolls over from all 1s to all 0s, it not only sets TF2 but also causes the Timer 2 registers to be loaded with the 16-bit value in the CRC (compare/reload/capture) register which is preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000H.

Table 1. Timer 2 reload mode selection

T2RI	T2R0	Mode
0	X	Reload Disabled
1	0	Mode 0: Auto-Reload upon Timer 2 Overflow (TF2)
1	1	Mode 1: Reload upon Falling Edge at Pin T2EX/P1.5

PD2435

The PD2435 is a CMOS 4-character 5x7 dot matrix alphanumeric programmable display with ROM to decode 128 ASCII alphanumeric characters and enough RAM to store the display's complete four digit ASCII message with software programmable attributes. The CMOS IC incorporates special interface control circuitry to allow the user to control the module as a fully supported microprocessor peripheral.

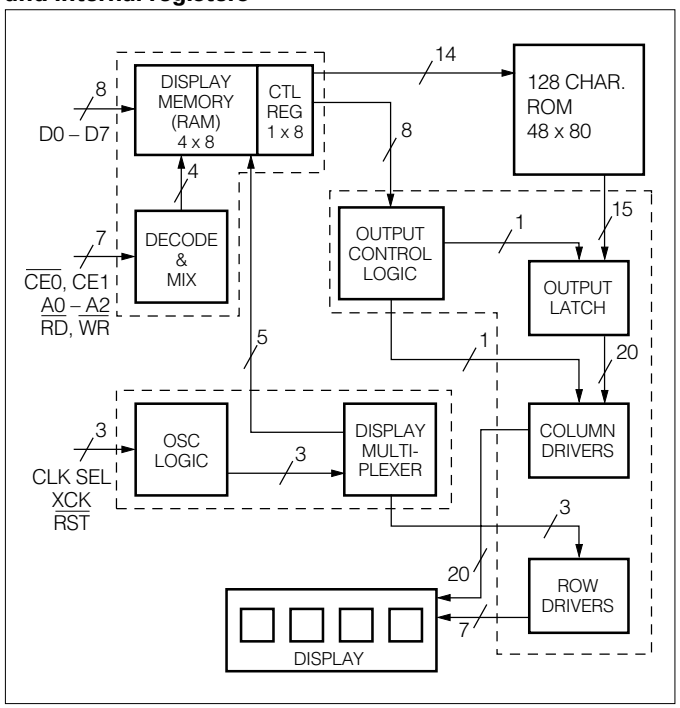
Microprocessor Interface

The interface to the microprocessor is through the address lines (A0-A2), the data bus (D0-D7), two chip select lines ($\overline{CE0}$, CE1), and ($\overline{RD}$) and ($\overline{WR}$) lines. The $\overline{CE0}$ should be held low and CE1 held high when executing a read or write to a specific PD243X device. The read and write lines are both active low. A valid write will enable the data as input lines.

Programming the PD2435

There are five registers within the PD2435. Four of the registers are used to hold the ASCII code of the four display characters. The fifth register is the Control Word, which is used to blink, blank, clear or dim the entire display to change the presentation (attributes) of individual characters.

Figure 3. PD2435 block diagram showing the major blocks and internal registers



Application

The speed regulated moving message display is an example where a digitized value of the controlling analog signal is used to compute a reload value for the Timer 2. The Timer 2 is operated in mode 0 where this reload value becomes a starting point for the Timer to count up. On overflow the Timer automatically takes the restart value for counting from reload register CRC. While the Timer is counting up, a new reload value is computed using the present A/D value.

Hardware

The circuit used in this application has the advantage of requiring a minimum of components. The single chip microcomputer SAB 80535 operates in conjunction with four alphanumeric programmable display chips PD 2435 to form a 16-digit long display.

The ASCII-coded data is transferred from the SAB 80535 to the display ICs via the data port P0 and using the control signal WR (P3.6) of the SAB 80535. The address pins from the ports P0 and P2 of the SAB 80535 are used to address the EPROM as well as the display chips in a memory-mapped I/O scheme. The display chips are addressed as memory locations with the following addresses.

Display Chip	Control Register Address	Digits Address
1	1000H	1004H-1007H
2	2000H	2004H-2007H
3	4000H	4004H-4007H
4	8000H	8004H-8007H

A push button is interfaced to port P3.2 of the SAB 80535 to provide an external interrupt to the microcontroller.

Firmware Description

Besides controlling speed of the moving message, there is a provision to interrupt the moving message and roll it backwards to the beginning of the message. The microcontroller reads the code and the message to display from an EPROM 2716A interfaced to the ports P0 and P2 of the SAB 80535. A virtual image of the message is created in the internal RAM of the SAB 80535. Four display chips PD2435 are interfaced to the SAB 80535 in a memory-mapped scheme and can be addressed as external memory to the SAB 80535. The virtual image of the message in internal RAM of the SAB 80535 is used to manipulate data to be displayed on the display chips. The internal RAM used for the display can be viewed as an area divided into two portions:

1. For active display
2. As a data buffer

The active display area is the replica of the data being displayed on the display chips. In this case the 16-digit display would need 16 RAM locations which correspond to 16 digits currently being displayed. The data buffer contains the rest of the message which is not being displayed. The message is shifted character by character in the RAM area. When the message on the display moves from right to left, the RAM buffer acts in "First In First Out" mode, and when the message on the display moves from left to right, the data to the display from the microcontroller RAM buffer is supplied in the "Last In First Out" scheme.

Between display of every character there is a software delay which depends upon the level of the analog signal supplied to the ANO pin of the SAB 80535. The external interrupt 0 (at port P3.2) is used to interrupt the microcontroller to inform it that the message needs to be scrolled backwards. On getting this interrupt the software sets the flag bit 0 which remains set until the message is scrolled back to the beginning of the message.

List of Components

Name	Number
SAB 80535	1
271 6A	1
PD2435	4
12 MHz Crystal	1
74LS373	1
22 pF Capacitors	2
100 nF Capacitor	1
4.7 µf Capacitor	1
1 k Resistor	1
10 k Pot	1

Reference Material for ICs

1. SAB 80515/80535 User's Manual.
2. PD2435 Data-Sheet or Optoelectronic Data Book (1990).

Figure 4. Interface circuit

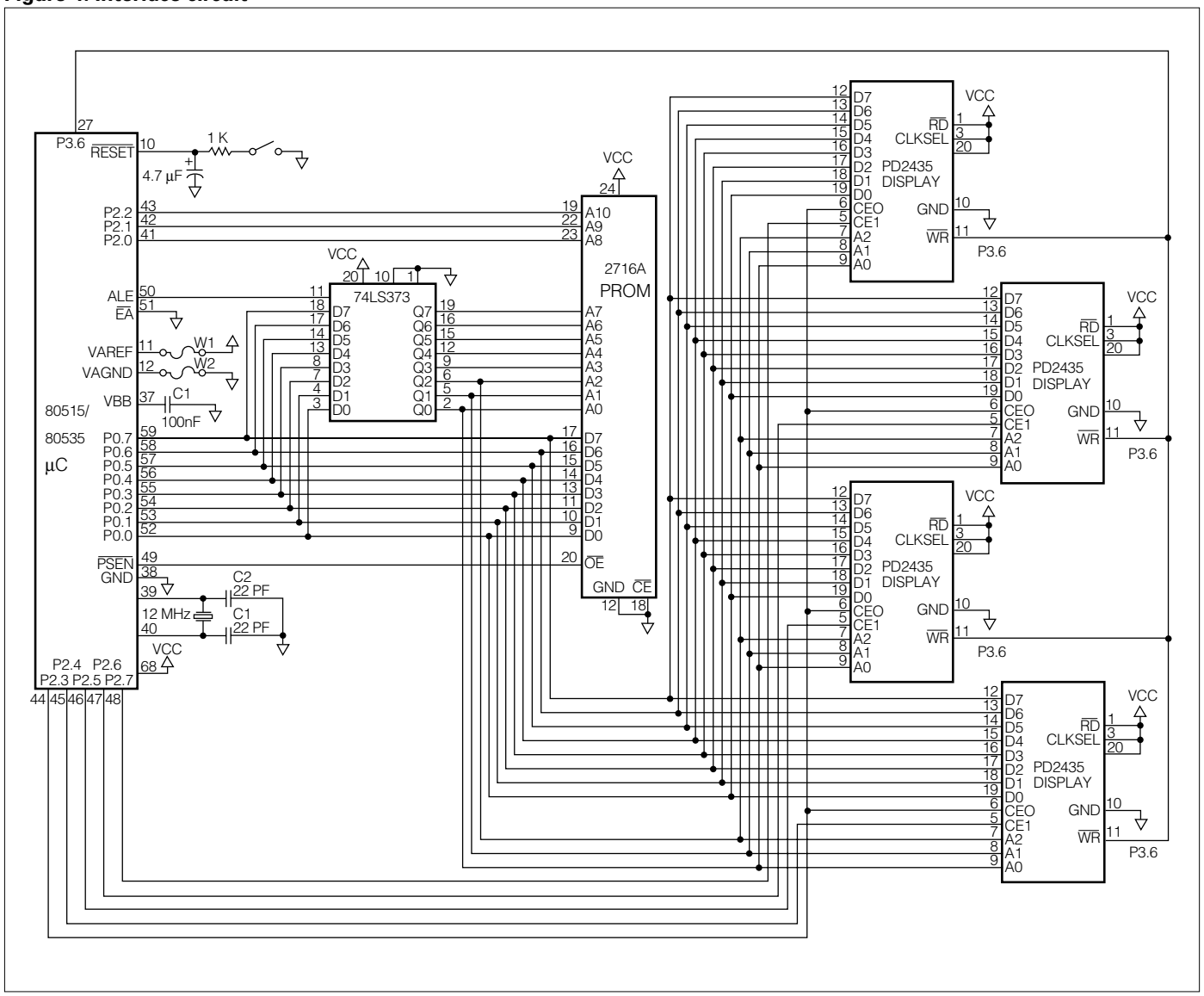
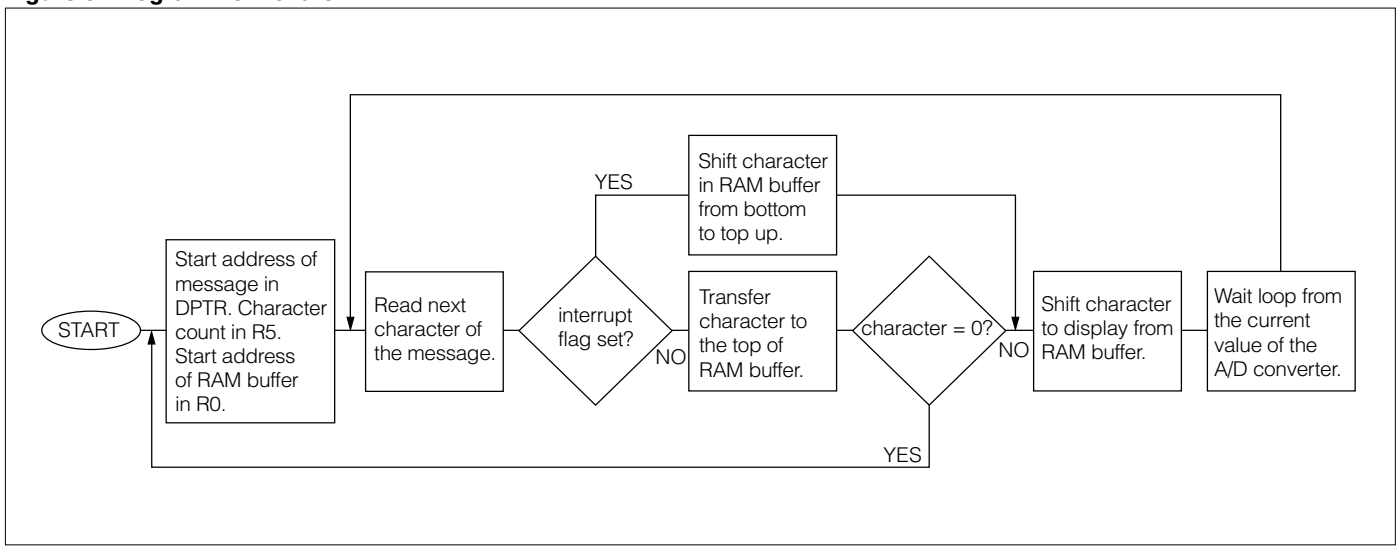


Figure 5. Program flow chart



Program listing

```
UDISP                'PD 2435 Display PROGRAM'

                  1          $TITLE ('PD 2435 DISPLAY PROGRAM')
                  2          $MOD515
                  3          $NOSYMBOLS
                  4
....              5          CSEG
                  6          $DEBUG
                  7
                  8
0000              9                     ORG       00H
                 10
0000 02000C      11                     LJMP      BEGIN          ;Jump on reset
                 12
                 13                     ;-------------------------------------------------------
                 14                     ; This is the interrupt subroutine for INTO. This is used to set a flag
                 15                     ; which then indicates that the message needs to be rolled back.
                 16                     ;-------------------------------------------------------
                 17                     ;
                 18
0003             19                     ORG       03H
                 20
0003 C0E0        21                     PUSH      ACC
0005 D2D5        22                     SETB      F0             ;Set flag for external interrupt
0007 D0E0        23                     POP       ACC
0009 C289        24                     CLR       IE0
000B 32          25                     RETI
                 26
                 27                     ;-------------------------------------------------------
                 28                     ; MAIN PROGRAM
                 29                     ;-------------------------------------------------------
                 30
000C D282        31          BEGIN:     SETB      P3.2           ;Set bit for INT0
000E 758110      32                     MOV       SP,#10H
0011 75D800      33                     MOV       ADCON, #00H    ;Select analog channel 0
                 34
0014 C2D5        35          OPTS:      CLR       F0             ;Clear flag 0
0016 7800        36                     MOV       R3,#00H        ;Character pointer in the message
0018 79FF        37                     MOV       R1,#0FFH       ;R1 used as a flag
001A 90F000      38                     MOV       DPTR,#0F000H   ;Control register of all displays
001D 7403        39                     MOV       A,#03H         ;Control word for display
001F F0          40                     MOVX      @DPTR,A
0020 9000C2      41                     MOV       DPTR,#(TEXT-1) ;Beginning of the text
0023 7820        42                     MOV       R0,#20H        ;Internal RAM location
0025 7D65        43                     MOV       R5,#101        ;A count for 101 characters
0027 7420        44                     MOV       A,#20H         ;ASCII for space
0029 F6          45          BLANK:     MOV       @R0,A          ;Fill all locations with blank
002A 08          46                     INC       R0
002B DDFC        47                     DJNZ      R5, BLANK
                 48
002D 12006C      49          SHIF:      CALL      NEXTC          ;Read the next character
0030 20D501      50                     JB        F0,TEMP        ;Check if the interrupt was raised
0033 0B          51                     INC       R3             ;If no interrupt
0034 7D65        52          TEMP:      MOV       R5,#101        ;Character count in message
0036 7820        53                     MOV       R0,#20H        ;RAM location 20H
0038 20D506      54                     JB        F0,REV0
003B C6          55          SHFT:      XCH       A,@R0          ;If no interrupt
003C 08          56                     INC       R0             ;Add the character
003D DDFC        57                     DJNZ      R5,SHFT        ;To the top of the RAM buffer
003F 0158        58                     AJMP      CONT0
0041 7421        59          REV0:      MOV       A,#21H         ;If there is no interrupt
0043 2B          60                     ADD       A,R3           ;Offset for the RAM buffer
```

```
0044 F8      61                 MOV     R0,A          ;Pointer in the RAM buffer
0045 7600    62                 MOV     @R0,#00H      ;Displayed so far
0047 7820    63                 MOV     R0,#20H       ;Beginning of the RAM buffer
0049 E6      64                 MOV     A,@R0         ;Read the character
004A C0E0    65                 PUSH    ACC           ;Save it
004C 08      66     AGAIN:      INC     R0            ;Next location in RAM buffer
004D E6      67                 MOV     A,@R0         ;Read the next character
004E 18      68                 DEC     R0            ;Back to first character
004F F6      69                 MOV     @R0,A         ;Replace with second character
0050 08      70                 INC     R0            ;Process repeats
0051 DDF9    71                 DJNZ    R5,AGAIN      ;Moving character backwards
0053 08      72                 INC     R0
0054 7600    73                 MOV     @R0,#00H      ;End of character buffer
0056 D0E0    74                 POP     ACC           ;Restore character
0058 7820    75     CONT0:      MOV     R0,#20H       ;Beginning of character buffer
005A E9      76                 MOV     A,R1          ;Check if end of character buffer
005B 6087    77                 JZ      OPTS
005D 120071  78                 CALL    OUTC
0060 C2AF    79                 CLR     IEN0.7        ;Disable interrupt
0062 1200A4  80                 CALL    WAITA         ;Before delay
0065 75A881  81                 MOV     IEN0,#81H     ;Enable interrupt
0068 D288    82                 SETB    IT0           ;INT0 control bit
006A 012D    83                 AJMP    SHIF
             84
             85                 ;  ---------------------------------------------------
             86                 ;  The routine moves a character of the message to ACC.
             87                 ;  ---------------------------------------------------
             88
006C A3      89     NEXTC:      INC     DPTR
006D 7400    90                 MOV     A,#0
006F 93      91                 MOVC    A,@A+DPTR     ;Move the character to Acc.
0070 22      92                 RET
             93
             94                 ;  ---------------------------------------------------
             95                 ;  This routine displays and moves a character over the four digits of
             96                 ;  the PD2435 and then repeats for the next display chip and so on.
             97                 ;  ---------------------------------------------------
             98                 ;
             99
0071 C0E0    100    OUTC:       PUSH    ACC
0073 C082    101                PUSH    DPL
0075 C083    102                PUSH    DPH
0077 7A04    103                MOV     R2,#4         ;For four digits (0 to 3) in a chip
0079 901004  104                MOV     DPTR,#1004H   ;Digit 0 in first display chip
007C 120098  105                CALL    OUTC0
007F 902004  106                MOV     DPTR,#2004H   ;Digit 0 in second display chip
0082 120098  107                CALL    OUTC0
0085 904004  108                MOV     DPTR,#4004H   ;Digit 0 in third display chip
0088 120098  109                CALL    OUTC0
008B 908004  110                MOV     DPTR,#8004H   ;Digit 0 in fourth display chip
008E 120098  111                CALL    OUTCO
0091 D083    112                POP     DPH
0093 0082    113                POP     DPL
0095 D0E0    114                POP     ACC
0097 22      115                RET
             116
             117                ;  ---------------------------------------------------
             118                ;  This is a nested subroutine. It moves a nonzero hex value (ASCII)
             119                ;  from left to right of the four digit display.
             120                ;  ---------------------------------------------------
             121                ;
             122
0098 E6      123    OUTC0:      MOV     A,@R0
```

```
0099 6007   124                 JZ      FIN
0098 F0     125                 MOVX    @DPTR,A
009C 08     126                 INC     R0
009D A3     127                 INC     DPTR
009E DAF8   128                 DJNZ    R2,OUTC0
00A0 7A04   129                 MOV     R2,#4
00A2 F9     130     FIN:        MOV     R1,A
00A3 22     131                 RET
            132
            133                 ;-------------------------------------------------------
            134                 ; This subroutine generates the software delay. The delay is
            135                 ; generated by the timer 2. The start count of the timer 2 is
            136                 ; computed from the present value of the A/D converter.
            137                 ;-------------------------------------------------------
            138                 ;
            139
00A4 7E03   140     WAITA:      MOV     R6,#03H
00A6 7D10   141     WAITB:      MOV     R5,#10H
00A8 75DA00 142     WAITC:      MOV     DAPR,#00H
00AB E5D9   143                 MOV     A,ADDAT
00AD 75F0FF 144                 MOV     B,#255          ;For computing reload value
00B0 A4     145                 MUL     AB              ;Reload value is computed
00B1 F5CA   146                 MOV     CRCL,A          ;Load the reload value low
00B3 85F0C8 147                 MOV     CRCH,B          ;Load the reload value high
00B6 75C811 148                 MOV     T2CON,#11H
00B9 10C602 149     WAITD:      JBC     TF2,WAITE
00BC 01B9   150                 AJMP    WAITD
00BE DDE8   151     WAITE:      DJNZ    R5,WAITC
00C0 DEE4   152                 DJNZ    R6,WAITB
00C2 22     153                 RET
            154
            155                 ;-------------------------------------------------------
            156                 ; MESSAGE
            157                 ;-------------------------------------------------------
            158
00C3 20202020 159   TEXT:       DB      '                   '
00C7 20202020
00CB 20202020
00CF 20202020
00D3 5349454D 160               DB      'SIEMENS MICROCONTROLLER SAB 80515/535'
00D7 454E5320
00D8 4D494352
00DF 4F434F4E
00E3 54524F4C
00E7 4C455220
00EB 53414220
00EF 38303531
00F3 352F3533
00F7 35
00F8 20202020 161               DB      '       SAB 80515/535       ',0
00FC 20202020
0100 20202020
0104 53414220
0108 38303531
010C 352F3533
0110 35202020
0114 20202020
0118 20202020
011C 20202020
0120 00
            162                 END
```

ASSEMBLY COMPLETE, 0 ERRORS FOUND