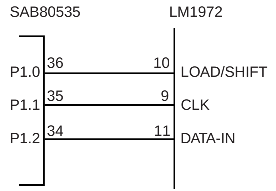
μC controls digital potentiometer

LUKASZ SLIWCZYNSKI, UNIVERSITY OF MINING AND METALLURGY, KRAKOW, POLAND

Many digitally controlled potentiometers (for example, the LM1971/2/3 from National Semiconductor, www.national.com) incorporate a three-wire serial digital interface, using data, clock, and enable lines. In **Figure 1**, the potentiometer's nomenclature for these lines is Data-In, Clk, and Load/Shift, respectively. The assembler program in **Listing 1** provides an interface to an SAB80535 μC. The main idea of the method is to use the capture/compare capability of Timer 2 in the μC to provide the timing relationship between the Data-In and Clk signals. The principal control of the interface comes from subroutine S16BIT in **Listing 1**.

To program the potentiometer, the μC must send 2 bytes to it—the "channel address," followed by the attenuation value with its most significant bit first. Sending a byte starts by loading the number of bits to send into the μC's BCOUNT register and initiating the P1 lines (setting P1.0 through P1.3 to a low state). Next, Timer 2 starts with overload enabled. The routine sets two digital compare/capture units (CC1 and

FIGURE 1



The rising edge of the clock signal validates data loading into the Data-In and Load/Shift pins of the potentiometer.

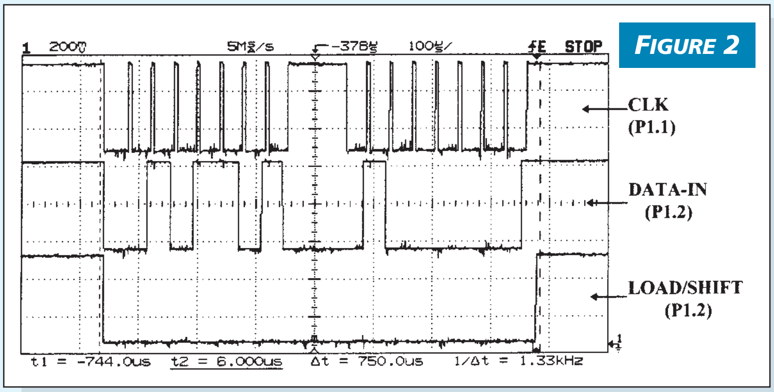
CC3 in the µC) to the "compare" mode by writing 88H into the CCEN register. The contents of the register decrease after each interrupt. In this way, eight consecutive interrupts occur, each to send 1 bit of data. The interrupt subroutine at address 006BH manages the transmission of the data bits via P1.2.

The CC1 unit generates the Clk signal on pin P1.1 when the contents of the Timer 2 count register (composed of TH2 and TL2 8-bit registers) equal the values set in registers CCL1 and CCH1. This value depends on the length of the interrupt subroutine. After transmission of the last bit, the routine stops Timer 2 by setting the value 0ECH in register T2CON. During data transmission, the main program spends its time waiting in the loop, as long as the bit FLAGS.0 is at logic 1. This bit clears in the last pass of the subroutine and sets just before Timer 2 starts. Transmission of the second byte of data occurs in exactly the same way after the routine reprograms the related registers. The Channel and Volume registers hold the 2 bytes to send.

Figures 2 and **3** show the timing relationships of the interface. In **Figure 2**, the 2 bytes Channel and Volume are 05AH and 081H, respectively. Data is valid on the rising edge of the Clk signal. **Figure 3** shows the time dependence in the interrupt-routine calls and the Clk rising edges. In this design, it takes approximately 740 µsec to program 2 bytes into the potentiometer with an 8-MHz clock frequency (a 1.5-µsec machine-cycle time). (DI #2256).

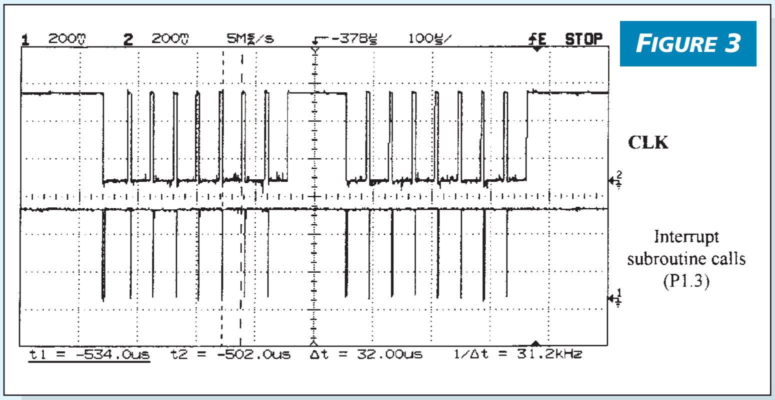
To Vote For This Design, Circle No. 400

FIGURE 2



The rising edge of the clock signal appears approximately 32 µsec after the CC3 compare/capture unit in the µC generates an interrupt.

FIGURE 3



The subroutine in Listing 1 provides a simple serial digital interface to National's (and others') digital potentiometers.

LISTING 1—S16BIT SUBROUTINE

```
;***************************REGISTERS ADDRESSES
        DEFINITIONS**************************

.EQU FLAGS,002FH
.EQU BCOUNT,004FH
.EQU CHANNEL,004DH
.EQU VOLUME,004EH

;****************************INTERRUPT SUBROUTINE
        HANDLER***************************
;THE NUMBER OF MACHINE CYCLES TO EXECUTE THE COMMAND ARE
GIVEN AT THE ;RIGHT-HAND SIDE. THREE ADDITIONAL CYCLES ARE ADDED
FOR THE RESPONSE TO ;THE INTERRUPT REQUEST

.ORG 006BH
          CLR IEN0.7; DISABLE ALL INTERRUPTS          1
          PUSH ACC;                               2
          PUSH PSW;                               2
          MOV A,BCOUNT;                           1
          JZ ALL_BY; TEST IF THE LAST BIT TO SEND     2 -> 8 + 3 = 11
          DEC A;                                  1
          MOV BCOUNT,A;                           1
          MOV A,CHANNEL;                          1
          RLC A;                                  1
          MOV P1.2,C; SEND THE BIT                2 -> 14 + 3 = 17
          MOV CHANNEL,A;                          1
          SETB IEN0.7;                            1
          POP PSW;                                2
          POP ACC;                                2 -> 22 + 3 = 25
          RETI;                                   2
ALL_BY:   MOV A,CHANNEL; THE LAST BIT TO SEND     1
          RLC A;                                  1
          MOV P1.2,C; SEND THE BIT                2 -> 12 + 3 = 15
          CLR FLAGS.0; INFORM S16BIT FUNCTION     1
          POP PSW;                                2
          POP ACC;                                2
          ANL T2CON,#0ECH; STOP TIMER 2           2
          SETB IEN0.7; ENABLE INTERRUPTS          1
          RETI;                                   2 -> 22 + 3 = 25
```

```
;*******************************************************************************
;THIS COMMANDS SHOULD BE PLACED SOMEWHERE IN THE BEGINING OF
THE MAIN ;PROGRAMM
          MOV SP,#06H; SET THE STACK POINTER
          MOV IP0,#020H; SET THE INTERRUPT PRIORITY LEVEL OF CC3
          MOV FLAGS,#00H; CLEAR ALL BITS
```

```
;**********************LM1972 INTERFACE CONTROLLING
        FUNCTION*********************

S16BIT:   MOV TL2,#(255 - 25);SET TIMER 2 COUNT REGISTER
          MOV TH2,#0FFH
          MOV CRCL,#(255 - 25);SET TIMER 2 RELOAD REGISTER
          MOV CRCH,#0FFH
          MOV CCL1,#0FCH; SET VALUE FOR CC1 - CLK SIGNAL
          MOV CCH1,#0FFH;
          MOV CCL3,#(255 - 25); SET THE BEGINING OF INTERRUPT
          MOV CCH3,#0FFH
          MOV BCOUNT,#07H; NUMBER OF BITS TO SEND - 1
          SETB FLAGS.0
          ANL P1,#070H; INITIATE PORT 1
          MOV CCEN,#088H; ENABLE COMPARE MODE FOR CC1 AND CC
          SETB IEN1.5; ENABLE INTERRUPT FROM CC3
          ORL T2CON,#011H; START TIMER 2 WITH OVERLOAD ENABLED
S16_1:    JB FLAGS.0,S16_1; WAIT UNTIL THE FIRST BYTE IS SEND
          MOV TL2,#(255 - 25); REINITIATE TIMER 2 COUNT REGISTER
          MOV TH2,#0FFH
          MOV BCOUNT,#07H; REINITIATE NUMBER OF BITS TO SEND
          MOV A,VOLUME; SWAP VOLUME AND CHANNEL CONTENTS
          MOV CHANNEL,A
          SETB FLAGS.0
          ORL T2CON,#011H; START TIMER 2 WITH OVERLOAD ENABLED
S16_2:    JB FLAGS.0,S16_2; WAIT UNTIL THE SECOND BYTE IS SEND
          MOV CCEN,#00H; DISABLE CC1 AND CC3 UNITS
          CLR IEN1.5; DISABLE CC3 INTERRUPT
          ORL P1,#07H; SET OUTPUT PINS TO THE HIGH STATE
S_END:    RET
```