Interfacing the Serial Input Dot Addressable Intelligent Displays with the Intel 5031/8051 Microcontroller Appnote 70

This application note is intended to serve as a design and application guide for users of the SCF574X and other similar display families in Infineon Opto Intelligent Displays data book, such as, SCE578X, SCD558X and SCD5510X with some minor modifications.

Description

The SCF5740 (Standard Red), SCF5742 (High Efficiency Red), and SCF5744 (High Efficiency Green) are 0.270 four digit 5x7 dot matrix, serial input dot addressable intelligent displays. The on-board CMOS is the heart of the display. It accepts decoded serial data and stores it in the internal RAM. The four characters are row multiplexed with RAM resident column data. The strobe rate is established by the internal or external MUX clock rate. The maximum external clock rate should be limited to 4 MHz.

Microprocessor Interface

In this application note, the 8031/8051 microcontroller family is used for demonstration, and the program is written in Assembly language. The display interface between SCF574X and Intel 8031 microcontroller is via serial port in mode 0, so the serial port control register will be a simple shift register. Serial data enters and exits through RXD, TXD outputs the shift clock. 8 bits are transmitted/received. Look at the Figure 1 for details.
Loading data into the display

In the following example, the word “ABCD” is used for demonstration of displaying a message:

```asm
.ORG 00H

BEGIN: CALL RE_SET
CALL LAMTST
CALL CLEAR
CALL BRGHNESS
NEXT: CALL DISPLAY
JMP NEXT

1. RESET ROUTINE: Reset the display to clear the Multiplex Counter, Address Register, Control Word Register, internal RAM and Data Register. The display will be blank, and brightness is set to 100%.

   RE_SET: CLR P1.2
   NOP
   SETB P1.2
   MOV SCON, #00H ;SET SERIAL MODE 0.
   RET

2. LAMP TEST ROUTINE: This routine is used to test all LEDs by turning them on for a short time.

   LAMTST: MOV R0, #A0 ;DIGIT ADDRESS 0.
   MOV R1, #1FH ;TURN ON ALL LEDs.
   MOV R3, #04H ;4-DIGIT COUNTER.
   MOV R2, #07H ;7-ROW COUNTER.
   LOOP_1: MOV A, R0 ;DIGIT ADDRESS
   CALL DISP ;LOAD DATA TO THE DISPLAY.
   LOOP_2: MOV A, R1 ;LOAD DATA TO RAM
   CALL DISP ;DISPLAY ROUTINE
   DJNZ R2, LOOP_2 ;TO DISPLAY ALL 7 ROWS.
   INC R0 ;NEXT DIGIT ADDRESS.
   DJNZ R3, LOOP_1 ;TO FINISH LAST DIGIT ADDRESS.
   CALL DELAY ;TIME DURATION OF DISPLAY.
   RET

3. CLEAR: Clear the display. The display is blank and character address register will be set to A0.

   CLEAR: MOV A, #C0H ;SET CONTROL WORD C0 TO CLEAR THE DISPLAY.
   CALL DISP ;LOAD DATA TO THE DISPLAY.
   RET

4. BRIGHTNESS: Set the desired brightness level. In this example, the brightness is set to 53% with maximum peak current.

   BRGHNESS: MOV R1, #E1H ;SET BRIGHTNESS LEVEL TO 53%.
   CALL DISP ;LOAD DATA TO THE DISPLAY.
   RET

5. DISPLAY: This routine is written to display the word “ABCD.”

   DISPLAY: MOV DPTR, #TABLE
   MOV R0, #A0H ;DIGIT ADDRESS 0
   MOV R3, #04H ;TO COUNT FOR 4 DIGITS.
   LOOP_A: MOV A, R0 ;INITIALIZE DIGIT ADDRESS FOR DISPLAY
   CALL DISP
```

© 2000 Infineon Technologies Corp. • Optoelectronics Division • San Jose, CA
www.infineon.com/opto • 1-888-Infineon (1-888-463-4636)
OSRAM Opto Semiconductors GmbH & Co. OHG • Regensburg, Germany
6. DISP routine: To output data byte from the accumulator to the display

```
CALL DIG_DPLY ;LOAD DATA TO THE APPROPRIATE DIGIT
INC R0 ;UPDATE TO THE NEXT DIGIT ADDRESS
DJNZ R3, LOOP_A
RET
```

7. DIG_DPLY routine: To output data bytes from the table to the display

```
6. DISP routine: To output data byte from the accumulator to the display

DISP: CLR P1.0 ;PULL LOAD LINE LOW
       MOV SBUF, A ;LOAD SBUF WITH DATA BYTE
       JNB SCON.1, $ ;TRANSMISSION NOT COMPLETE. WAIT.
       CLR SCON.1 ;TRANSMISSION COMPLETE.
       SETB P1.0 ;LATCH DATA TO DISPLAY.
       RET
```

8. DELAY routine: Duration of the display

```
8. DELAY routine: Duration of the display

DELAY: MOV R6, #1AH
       NOP
DELAY_1: MOV R5, #EFH
       NOP
DELAY_2: MOV R7, #1FH
       NOP
       DJNZ R7, $  ;A
       NOP
       DJNZ R5, DELAY_2
       NOP
       DJNZ R6, DELAY_1
       NOP
       RET
```

       DB 1EH,11H,11H,1EH,11H,11H,1EH ; “B”
       DB 07H,08H,10H,10H,10H,08H,07H ; “C”

.END