CHAPTER 12

REAL-WORLD INTERFACING I: LCD, ADC, AND SENSORS

OBJECTIVES

Upon completion of this chapter, you will be able to:

- ⋙ **List reasons that LCDs are gaining wide-spread use, replacing LEDs**
- ⋙ **Describe the functions of the pins of a typical LCD**
- ⋙ **List instruction command codes for programming an LCD**
- ⋙ **Interface an LCD to the 8051**
- ⋙ **Program an LCD by sending data or commands to it from the 8051**
- ⋙ **Interface ADC (analog-to-digital converter) chips to the 8051**
- ⋙ **Interface temperature sensors to the 8051**
- ⋙ **Explain the process of data acquisition using ADC chips**
- ⋙ **Describe factors to consider in selecting an ADC chip**
- ⋙ **Describe the function of the pins of a typical ADC chip**
- ⋙ **Explain the function of precision IC temperature sensors**
- ⋙ **Describe signal conditioning and its role in data acquisition**

This chapter explores some real-world applications of the 8051. We explain how to interface the 8051 to devices such as an LCD, ADC, and sensors. In Section 12.1, we show LCD interfacing with the 8051. In Section 12.2, we describe analog-to-digital converter (ADC) connection with sensors and the 8051.

SECTION 12.1: INTERFACING AN LCD TO THE 8051

This section describes the operation modes of LCDs, then describes how to program and interface an LCD to an 8051.

LCD operation

In recent years the LCD is finding widespread use replacing LEDs (seven-segment LEDs or other multisegment LEDs). This is due to the following reasons:

1. The declining prices of LCDs.
2. The ability to display numbers, characters, and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.
3. Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU (or in some other way) to keep displaying the data.
4. Ease of programming for characters and graphics.

LCD pin descriptions

The LCD discussed in this section has 14 pins. The function of each pin is given in Table 12-1. Figure 12-1 shows the pin positions for various LCDs.

V_{CC}, V_{SS}, and V_{EE}

While V_{CC} and V_{SS} provide +5V and ground, respectively, V_{EE} is used for controlling LCD contrast.

RS, register select

There are two very important registers inside the LCD. The RS pin is used for their selection as follows. If RS = 0, the instruction command code register is selected, allowing the user to send a command such as clear display, cursor at home, etc. If RS = 1 the data register is selected, allowing the user to send data to be displayed on the LCD.

R/W, read/write

R/W input allows the user to write information to the LCD or read information from it. R/W = 1 when reading; R/W = 0 when writing.

E, enable

The enable pin is used by the LCD to latch information presented to its data pins.

Table 12-1. Pin Descriptions for LCD

Pin	Symbol	I/O	Description
1	V_{SS}	--	Ground
2	V_{CC}	--	+5V power supply
3	V_{EE}	--	Power supply to control contrast
4	RS	I	RS=0 to select command register, RS=1 to select data register
5	R/W	I	R/W=0 for write, R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

When data is supplied to data pins, a high-to-low pulse must be applied to this pin in order for the LCD to latch in the data present at the data pins. This pulse must be a minimum of 450 ns wide.

D0 - D7

The 8-bit data pins, D0 - D7, are used to send information to the LCD or read the contents of the LCD's internal registers.

To display letters and numbers, we send ASCII codes for the letters A - Z, a - z, and numbers 0 - 9 to these pins while making RS = 1.

There are also instruction command codes that can be sent to the LCD to clear the display or force the cursor to the home position or blink the cursor. Table 12-2 lists the instruction command codes.

We also use RS = 0 to check the busy flag bit to see if the LCD is ready to receive information. The busy flag is D7 and can be read when R/W = 1 and RS = 0, as follows: if R/W = 1, RS = 0. When D7 = 1 (busy flag = 1), the LCD is busy taking care of internal operations and will not accept any new information. When D7 = 0, the LCD is ready to receive new information. *Note:* It is recommended to check the busy flag before writing any data to the LCD.

Table 12-2: LCD Command Codes

Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

Note: This table is extracted from Table 12-4.

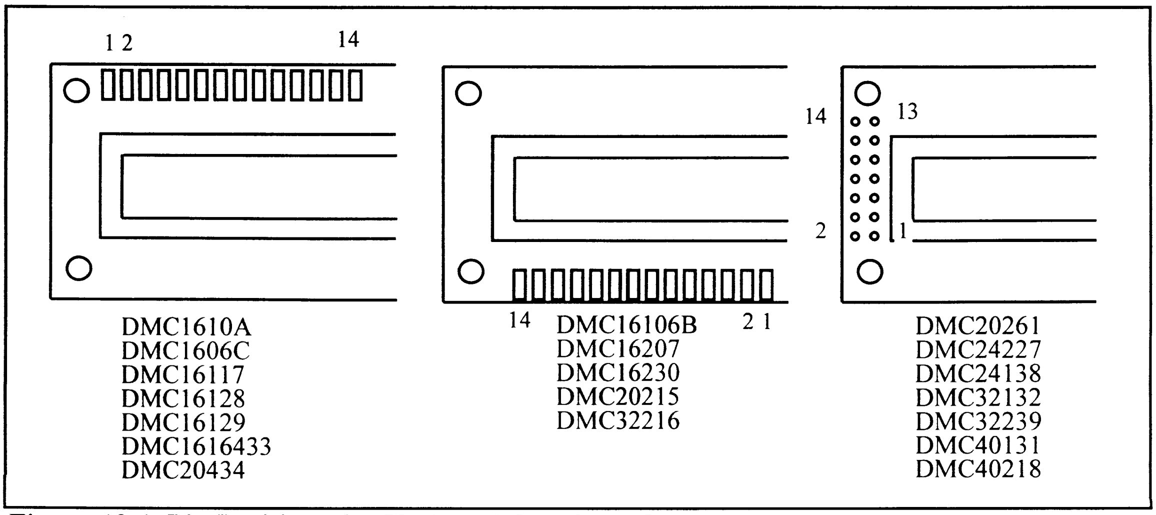


Figure 12-1. Pin Positions for Various LCDs from Optrex

Sending commands and data to LCDs with a time delay

To send any of the commands from Table 12-2 to the LCD, make pin RS = 0. For data, make RS = 1. Then send a high-to-low pulse to the E pin to enable the internal latch of the LCD. This is shown in the code below. See Figure 12-2.

```
;calls a time delay before sending next data/command
; P1.0-P1.7 are connected to LCD data pins D0-D7
; P2.0 is connected to RS pin of LCD
; P2.1 is connected to R/W pin of LCD
; P2.2 is connected to E pin of LCD
         ORG
         MOV     A,#38H  ;init. LCD 2 lines,5x7 matrix
         ACALL   COMNWRT      ;call command subroutine
         ACALL   DELAY        ;give LCD some time
         MOV     A,#0EH       ;display on, cursor on
         ACALL   COMNWRT      ;call command subroutine
         ACALL   DELAY        ;give LCD some time
         MOV     A,#01        ;clear LCD
         ACALL   COMNWRT      ;call command subroutine
         ACALL   DELAY        ;give LCD some time
         MOV     A,#06H       ;shift cursor right
         ACALL   COMNWRT      ;call command subroutine
         ACALL   DELAY        ;give LCD some time
         MOV     A,#84H       ;cursor at line 1,pos. 4
         ACALL   COMNWRT      ;call command subroutine
         ACALL   DELAY        ;give LCD some time
         MOV     A,#'N'       ;display letter N
         ACALL   DATAWRT      ;call display subroutine
         ACALL   DELAY        ;give LCD some time
         MOV     A,#'O'       ;display letter O
         ACALL   DATAWRT      ;call display subroutine
AGAIN:   SJMP    AGAIN        ;stay here
COMNWRT:                      ;send command to LCD
         MOV     P1,A         ;copy reg A to port1
         CLR     P2.0         ;RS=0 for command
         CLR     P2.1         ;R/W=0 for write
         SETB    P2.2         ;E=1 for high pulse
         CLR     P2.2         ;E=0 for H-to-L pulse
         RET
DATAWRT:                      ;write data to LCD
         MOV     P1,A         ;copy reg A to port1
         SETB    P2.0         ;RS=1 for data
         CLR     P2.1         ;R/W=0 for write
         SETB    P2.2         ;E=1 for high pulse
         CLR     P2.2         ;E=0 for H-to-L pulse
         RET
DELAY:   MOV     R3,#50   ;50 or higher for fast CPUs
HERE2:   MOV     R4,#255    ;R4=255
HERE:    DJNZ    R4,HERE    ;stay until R4 becomes 0
         DJNZ    R3,HERE2
         RET
         END
```

Sending code or data to the LCD with checking busy flag

The above code showed how to send commands to the LCD without checking the busy flag. Notice that we must put a long delay in between issuing data or commands to the LCD. However, a much better way is to monitor the busy flag before issuing a command or data to the LCD. This is shown below.

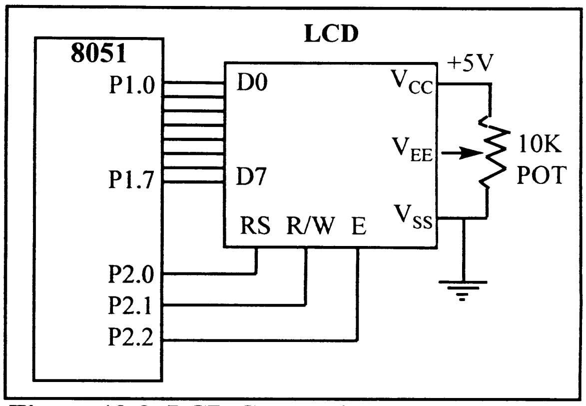


Figure 12-2. LCD Connection

```
;Check busy flag before sending data, command to LCD
;P1=data pin
;P2.0 connected to RS pin
;P2.1 connected to R/W pin
;P2.2 connected to E pin
        ORG
        MOV     A,#38H  ;init. LCD 2 lines,5x7 matrix
        ACALL   COMMAND       ;issue command
        MOV     A,#0EH        ;LCD on, cursor on
        ACALL   COMMAND       ;issue command
        MOV     A,#01H        ;clear LCD command
        ACALL   COMMAND       ;issue command
        MOV     A,#06H        ;shift cursor right
        ACALL   COMMAND       ;issue command
        MOV     A,#86H        ;cursor: line 1, pos. 6
        ACALL   COMMAND       ;command subroutine
        MOV     A,#'N'        ;display letter N
        ACALL   DATA_DISPLAY
        MOV     A,#'O'        ;display letter O
        ACALL   DATA_DISPLAY
HERE:   SJMP    HERE          ;STAY HERE
COMMAND: ACALL  READY         ;is LCD ready?
        MOV     P1,A          ;issue command code
        CLR     P2.0          ;RS=0 for command
        CLR     P2.1          ;R/W=0 to write to LCD
        SETB    P2.2          ;E=1 for H-to-L pulse
        CLR     P2.2          ;E=0 ,latch in
        RET
DATA_DISPLAY:
        ACALL   READY         ;is LCD ready?
        MOV     P1,A          ;issue data
        SETB    P2.0          ;RS=1 for data
        CLR     P2.1          ;R/W=0 to write to LCD
        SETB    P2.2          ;E=1 for H-to-L pulse
        CLR     P2.2          ;E=0, latch in
        RET
```

```
READY:
      SETB      P1.7               ;make P1.7 input port
      CLR       P2.0               ;RS=0 access command reg
      SETB      P2.1               ;R/W=1 read command reg
;read command reg and check busy flag
BACK:CLR        P2.2               ;E=1 for H-to-L pulse
      SETB      P2.2               ;E=0 H-to-L pulse
      JB        P1.7,BACK          ;stay until busy flag=0
      RET
      END
```

Notice in the above program that the busy flag is D7 of the command register. To read the command register we make R/W = 1, RS = 0, and a H-to-L pulse for the E pin will provide us the command register. After reading the command register, if bit D7 (the busy flag) is high, the LCD is busy and no information (command or data) should be issued to it. Only when D7 = 0 can we send data or commands to the LCD. Notice in this method that there are no time delays used since we are checking the busy flag before issuing commands or data to the LCD.

LCD data sheet

In the LCD, one can put data at any location. The following shows address locations and how they are accessed.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A	A	A	A	A	A	A

where AAAAAAA = 0000000 to 0100111 for line 1 and AAAAAAA = 1000000 to 1100111 for line 2. See Table 12-3.

Table 12-3: LCD Addressing

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Line 1 (min)	1	0	0	0	0	0	0	0
Line 1 (max)	1	0	1	0	0	1	1	1
Line 2 (min)	1	1	0	0	0	0	0	0
Line 2 (max)	1	1	1	0	0	1	1	1

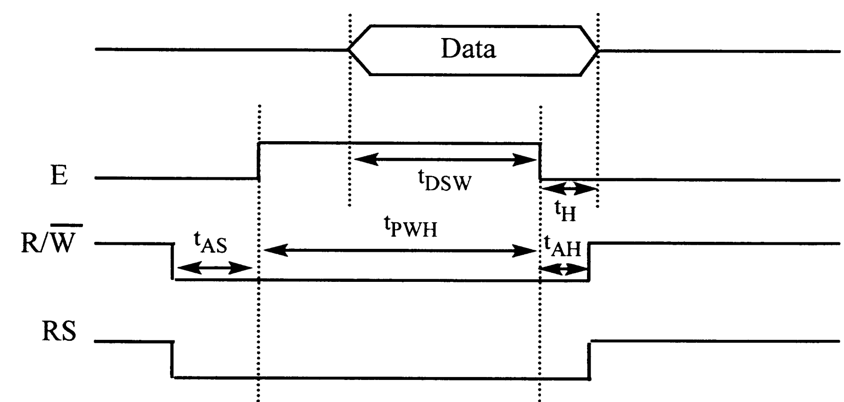
The upper address range can go as high as 0100111 for the 40-character-wide LCD while for the 20-character-wide LCD it goes up to 010011 (19 decimal = 10011 binary). Notice that the upper range 0100111 (binary) = 39 decimal which corresponds to locations 0 to 39 for the LCDs of 40x2 size.

From the above discussion we can get the addresses of cursor positions for various sizes of LCDs. See Figure 12-3. Note that all the addresses are in hex. Figure 12-4 gives a diagram of LCD timing. Table 12-4 provides a detailed list of LCD commands and instructions. Table 12-2 is extracted from this table.

16 x 2 LCD	80	81	82	83	84	85	86 through 8F
	C0	C1	C2	C3	C4	C5	C6 through CF
20 x 1 LCD	80	81	82	83	through 93		
20 x 2 LCD	80	81	82	83	through 93		
	C0	C1	C2	C3	through D3		
20 x 4 LCD	80	81	82	83	through 93		
	C0	C1	C2	C3	through D3		
	94	95	96	97	through A7		
	D4	D5	D6	D7	through E7		
40 x 2 LCD	80	81	82	83	through A7		
	C0	C1	C2	C3	through E7		

Note: All data is in hex.

Figure 12-3 Cursor Addresses for Some LCDs



t_{PWH} = Enable pulse width = 450 ns (minimum)
t_{DSW} = Data set up time = 195 ns (minimum)
t_H = Data hold time = 10 ns (minimum)
t_{AS} = Set up time prior to E (going high) for both RS and R/W = 140 ns (minimum)
t_{AH} = Hold time after E has come down for both RS and R/W = 10 ns (minimum)

Figure 12-4: LCD Timing

Review Questions

1. The RS pin is an _______ (input, output) pin for the LCD.
2. The E pin is an ________ (input, output) pin for the LCD.
3. The E pin requires an _______ (H-to-L, L-to-H) pulse to latch in information at the data pins of the LCD.
4. For the LCD to recognize information at the data pins as data, RS must be set to _____ (high, low).
5. Give the command codes for line 1, first character, and line 2, first character.

Table 12-4: List of LCD Instructions

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (Max)
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DD RAM address 0 in address counter	1.64 ms
Return Home	0	0	0	0	0	0	0	0	1	-	Sets DD RAM address 0 as address counter. Also returns display being shifted to original position. DD RAM contents remain unchanged.	1.64 ms
Entry Mode Set	0	0	0	0	0	0	0	1	1/D	S	Sets cursor move direction and specifies shift of display. These operations are performed during data write and read.	40 μs
Display On/ Off Control	0	0	0	0	0	0	1	D	C	B	Sets On/Off of entire display (D), cursor On/Off (C), and blink of cursor position character (B).	40 μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Moves cursor and shifts display without changing DD RAM contents.	40 μs
Function Set	0	0	0	0	1	DL	N	F	-	-	Sets interface data length (DL), number of display lines (L) and character font (F).	40 μs
Set CG RAM Address	0	0	0	1	AGC						Sets CG RAM address. CG RAM data is sent and received after this setting.	40 μs
Set DD RAM Address	0	0	1	ADD							Sets DD RAM address. DD RAM data is sent and received after this setting.	40 μs
Read Busy Flag & Address	0	1	BF	AC							Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.	40 μs
Write Data CG or DD RAM	1	0	Write Data								Writes data into DD or CG RAM.	40 μs
Read Data CG or DD RAM	1	1	Read Data								Reads data from DD or CG RAM.	40 μs

Notes:

1. Execution times are maximum times when fcp or fosc is 250 kHz.
2. Execution time changes when frequency changes. Ex: When fcp or fosc is 270 kHz: 40 μs × 250 / 270 =37 μs.
3. Abbreviations:

DD RAM	Display data RAM		
CG RAM	Character generator RAM		
ACC	CG RAM address		
ADD	DD RAM address, corresponds to cursor address		
AC	Address counter used for both DD and CG RAM addresses.		
1/D = 1	Increment	1/D = 0	Decrement
S=1	Accompanies display shift		
S/C = 1	Display shift;	S/C = 0	Cursor move
R/L = 1	Shift to the right;	R/L = 0	Shift to the left
DL = 1	8 bits, DL = 0: 4 bits		
N = 1	1line, N = 0 : 1 line		
F = 1	5 x 10 dots, F = 0 : 5 x 7 dots		
BF = 1	Internal operation;	BF = 0 Can accept instruction	

SECTION 12.2: 8051 INTERFACING TO ADC, SENSORS

This section will explore interfacing ADC (analog-to-digital converter) chips and temperature sensors to the 8051. First, we describe ADC chips, then show how to interface an ADC to the 8051. Then we examine the characteristics of the LM35 temperature sensor and show how to interface it to the 8051.

ADC devices

Analog-to-digital converters are among the most widely used devices for data acquisition. Digital computers use binary (discrete) values, but in the physical world everything is analog (continuous). Temperature, pressure (wind or liquid), humidity, and velocity are a few examples of physical quantities that we deal with every day. A physical quantity is converted to electrical (voltage, current) signals using a device called a *transducer*. Transducers are also referred to as *sensors*. Although there are sensors for temperature, velocity, pressure, light, and many other natural quantities, they produce an output that is voltage (or current). Therefore, we need an analog-to-digital converter to translate the analog signals to digital numbers so that the microcontroller can read them. A widely used ADC chip is the ADC804.

ADC804 chip

The ADC804 IC is an analog-to-digital converter in the family of the ADC800 series from National Semiconductor. It is also available from many other manufacturers. It works with +5 volts and has a resolution of 8 bits. In addition to resolution, conversion time is another major factor in judging an ADC. *Conversion time* is defined as the time it takes the ADC to convert the analog input to a digital (binary) number. In the ADC804, the conversion time varies depending on the clocking signals applied to the CLK R and CLK IN pins, but it cannot be faster than 110 μs. The ADC804 pin descriptions follow.

CS

Chip select is an active low input used to activate the ADC804 chip. To access the ADC804, this pin must be low.

RD (read)

This is an input signal and is active low. The ADC converts the analog input to its binary equivalent and holds it in an internal register. RD is used to get the converted data out of the ADC804 chip. When CS = 0, if a high-to-low pulse is applied to the RD pin, the 8-bit digital output shows up at the D0 - D7 data pins. The RD pin is also referred to as output enable.

WR (write; a better name might be "start conversion")

This is an active low input used to inform the ADC804 to start the conversion process. If CS = 0 when WR makes a low-to-high transition, the ADC804 starts converting the analog input value of Vin to an 8-bit digital number. The amount of time it takes to convert varies depending on the CLK IN and CLK R values explained below. When the data conversion is complete, the INTR pin is forced low by the ADC804.

CLK IN and CLK R

CLK IN is an input pin connected to an external clock source when an external clock is used for timing. However, the 804 has an internal clock generator. To use the internal clock generator (also called self-clocking) of the ADC804, the CLK IN and CLK R pins are connected to a capacitor and a resistor, as shown in Figure 12-5. In that case the clock frequency is determined by the equation:

$$f = \frac{1}{1.1\,RC}$$

Typical values are R = 10K ohms and C = 150 pF. Substituting in the above equation, we get f = 606 kHz. In that case, the conversion time is 110 μs.

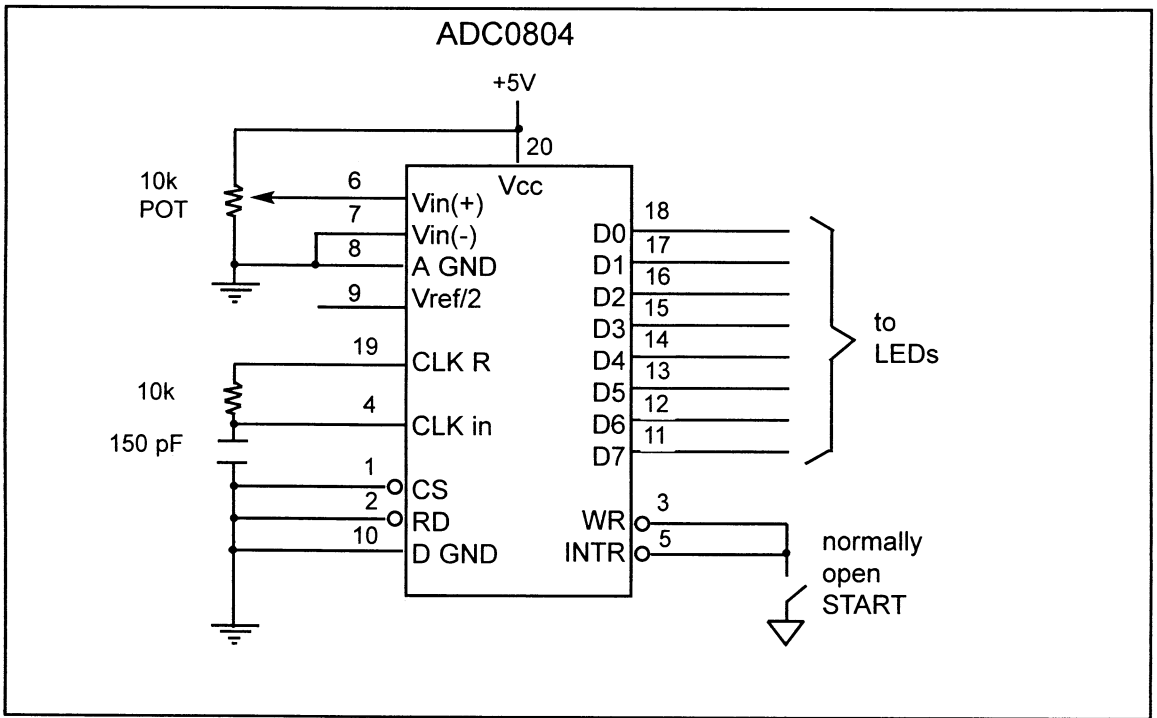


Figure 12-5. Testing ADC804 in Free Running Mode

INTR (interrupt; a better name might be "end of conversion")

This is an output pin and is active low. It is a normally high pin and when the conversion is finished, it goes low to signal the CPU that the converted data is ready to be picked up. After INTR goes low, we make CS = 0 and send a high-to-low pulse to the RD pin to get the data out of the ADC804 chip.

V_{in} (+) and V_{in} (-)

These are the differential analog inputs where $V_{in} = V_{in}(+) - V_{in}(-)$. Often the V_{in} (-) pin is connected to ground and the V_{in} (+) pin is used as the analog input to be converted to digital.

V_{CC}

This is the +5 volt power supply. It is also used as a reference voltage when the $V_{ref}/2$ input (pin 9) is open (not connected). This is discussed next.

$V_{ref}/2$

Pin 9 is an input voltage used for the reference voltage. If this pin is open (not connected), the analog input voltage for the ADC804 is in the range of 0 to 5 volts (the same as the V_{CC} pin). However, there are many applications where the analog input applied to Vin needs to be other than the 0 to +5 V range. $V_{ref}/2$ is used to implement analog input voltages other than 0 – 5 V. For example, if the analog input range needs to be 0 to 4 volts, $V_{ref}/2$ is connected to 2 volts. Table 12-5 shows the V_{in} range for various $V_{ref}/2$ inputs.

Table 12-5: $V_{ref}/2$ Relation to V_{in} Range

$V_{ref}/2$ (V)	V_{in} (V)	Step Size (mV)
not connected*	0 to 5	5/256 = 19.53
2.0	0 to 4	4/255 = 15.62
1.5	0 to 3	3/256 = 11.71
1.28	0 to 2.56	2.56/256 = 10
1.0	0 to 2	2/256 = 7.81
0.5	0 to 1	1/256 = 3.90

Notes: V_{CC} = 5 V

* When not connected (open), $V_{ref}/2$ is measured at 2.5 volts for V_{CC} = 5 V.

Step Size (resolution) is the smallest change that can be discerned by an ADC.

D0 - D7

D0 - D7 (where D7 is the MSB, D0 the LSB) are the digital data output pins. These are tri-state buffered and the converted data is accessed only when CS = 0 and RD is forced low. To calculate the output voltage, use the following formula.

$$D_{out} = \frac{V_{in}}{step\ size}$$

where D_{out} = digital data output (in decimal), V_{in} = analog input voltage, and step size (resolution) is the smallest change, which is $(2 \times V_{ref}/2)/256$ for an 8-bit ADC.

Analog ground and digital ground

These are the input pins providing the ground for both the analog signal and the digital signal. Analog ground is connected to the ground of the analog V_{in} while digital ground is connected to the ground of the V_{CC} pin. The reason that we have two ground pins is to isolate the analog V_{in} signal from transient voltages caused by digital switching of the output D0 - D7. Such isolation contributes to the accuracy of the digital data output. In our discussion, both are connected to the same ground; however, in the real world of data acquisition the analog and digital grounds are handled separately.

From this discussion we conclude that the following steps must be followed for data conversion by the ADC804 chip.

1. Make CS = 0 and send a low-to-high pulse to pin WR to start the conversion.
2. Keep monitoring the INTR pin. If INTR is low, the conversion is finished and we can go to the next step. If INTR is high, keep polling until it goes low.
3. After the INTR has become low, we make CS = 0 and send a high-to-low pulse to the RD pin to get the data out of the ADC804 IC chip. The timing for this process is shown in Figure 12-6.

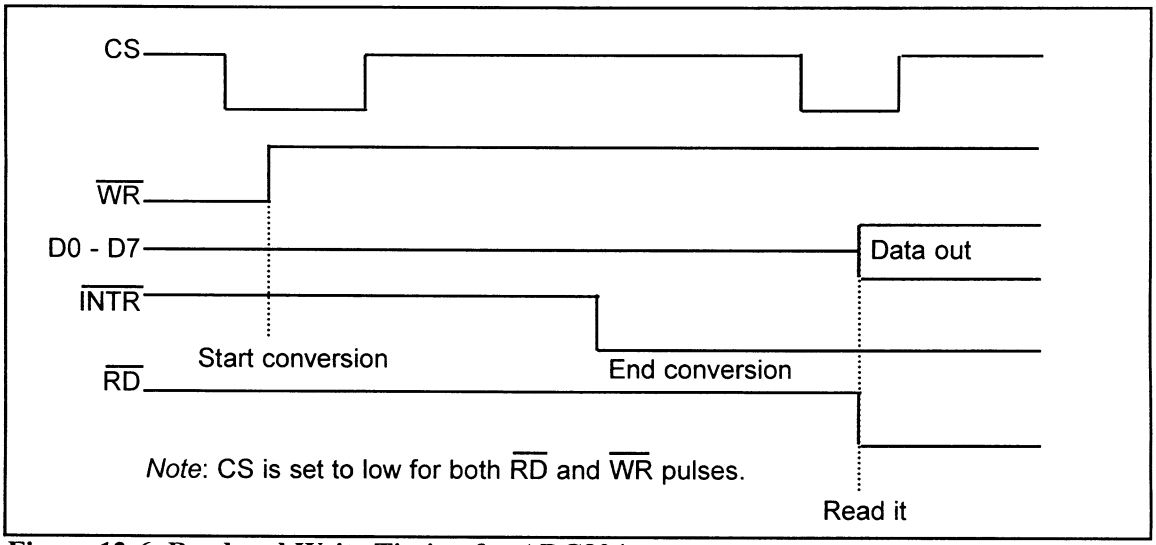


Figure 12-6. Read and Write Timing for ADC804

Testing the ADC804

One can test the ADC804 using the circuit shown in Figure 12-5. This setup is called free running test mode and is recommended by the manufacturer. Figure 12-5 shows a potentiometer used to apply a 0-to-5 V analog voltage to the input V_{in} (+) of the 804 ADC. The binary outputs are monitored on the LEDs of the digital trainer. It must be noted that in free running test mode the CS input is grounded and the WR input is connected to the INTR output. However, according to National Semiconductor's databook "the WR and INTR node should be momentarily forced to low following a power-up cycle to guarantee operation."

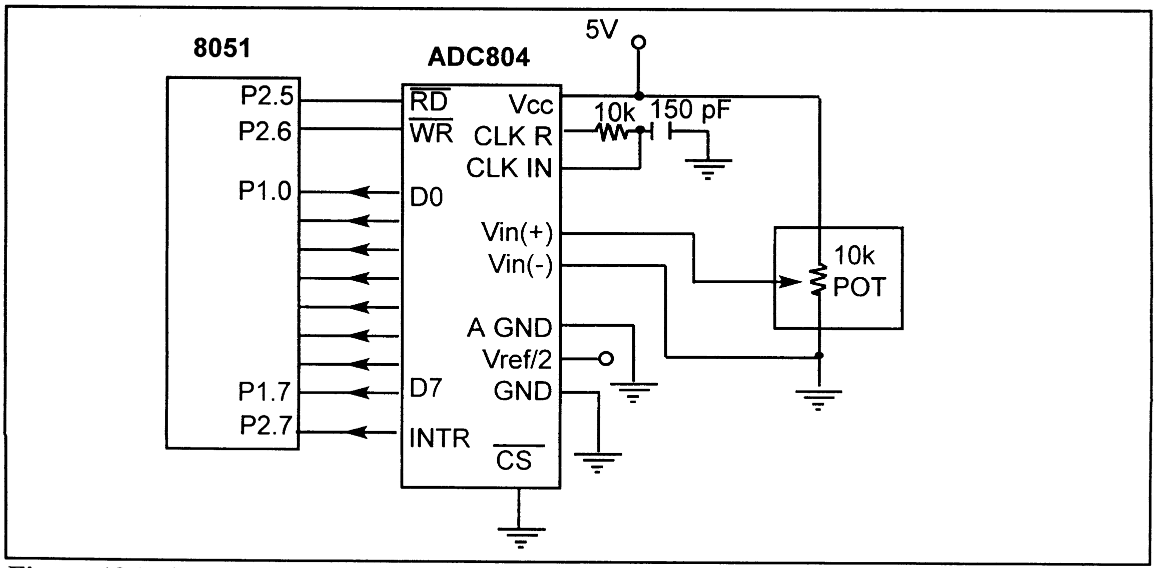


Figure 12-7. 8051 Connection to ADC804 with Self-Clocking

Example 12-1

Examine the ADC804 connection to the 8051 in Figure 12-7. Write a program to monitor the INTR pin and bring an analog input into register A. Then call a hex-to-ASCII conversion and data display subroutines. Do this continuously.

Solution:

```
;P2.6 = WR (start conversion needs to L-to-H pulse)
;P2.7 When low, end-of-conversion)
;P2.5 = RD (a H-to-L will read the data from ADC chip)
;P1,0 - P1.7 = D0 - D7 of the ADC804
;
         MOV  P1,#0FFH   ;make P1 = input
BACK:    CLR  P2.6       ;WR=0
         SETB P2.6       ;WR=1 L-to-H to start conversion
HERE:    JB   P2.7,HERE  ;wait for end of conversion
         CLR  P2.5       ;conversion finished,enable RD
         MOV  A,P1       ;read the data
         ACALL CONVERSION    ;hex-to-ASCII conversion
         ACALL DATA_DISPLAY  ;display the data
         SETB P2.5       ;make RD=1 for next round
         SJMP BACK
```

Note: For a hex-to-ASCII conversion subroutine, see Chapter 7.

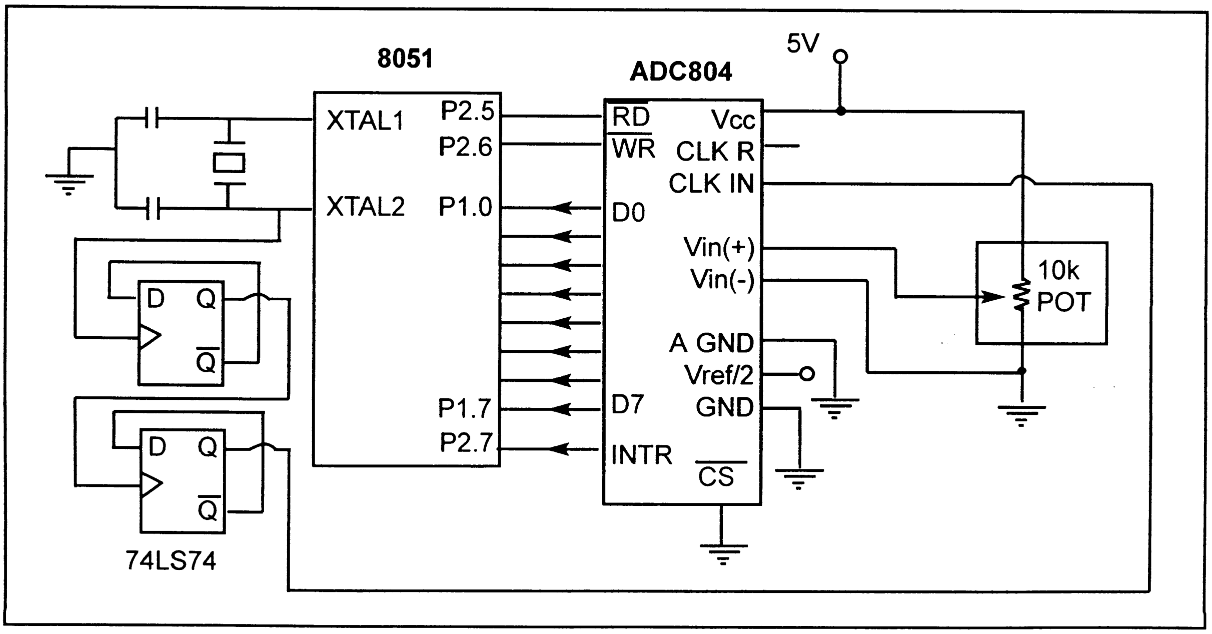


Figure 12-8. 8051 Connection to ADC804 with Clock from XTAL2 of the 8051

In Figure 12-8, notice that the clock in for the ADC804 is coming from the crystal of the microcontroller. Since this frequency is too high, we use two D flip-flops (74LS74) to divide the frequency by 4. A D flip-flop divides the frequency by 2 if we connect its $\overline{Q}$ to the D input. For a higher frequency, use more flip-flops.

Interfacing a temperature sensor to the 8051

Transducers convert physical data such as temperature, light intensity, flow, and speed to electrical signals. Depending on the transducer, the output produced is in the form of voltage, current, resistance, or capacitance. For example, temperature is converted to electrical signals using a transducer called a *thermistor*. A thermistor responds to temperature change by changing resistance, but its response is not linear, as seen in Table 12-6.

Table 12-6. Thermistor Resistance vs. Temperature

Temperature (C)	Tf (K ohms)
0	29.490
25	10.000
50	3.893
75	1.700
100	0.817

From William Kleitz, *Digital Electronics*

Table 12-7. LM34 Temperature Sensor Series Selection Guide

Part Scale	Temperature Range	Accuracy	Output
LM34A	–50 F to +300 F	+2.0 F	10 mV/F
LM34	–50 F to +300 F	+3.0 F	10 mV/F
LM34CA	–40 F to +230 F	+2.0 F	10 mV/F
LM34C	–40 F to +230 F	+3.0 F	10 mV/F
LM34D	–32 F to +212 F	+4.0 F	10 mV/F

Note: Temperature range is in degrees Fahrenheit.

Table 12-8. LM35 Temperature Sensor Series Selection Guide

Part	Temperature Range	Accuracy	Output Scale
LM35A	–55 C to +150 C	+1.0 C	10 mV/F
LM35	–55 C to +150 C	+1.5 C	10 mV/F
LM35CA	–40 C to +110 C	+1.0 C	10 mV/F
LM35C	–40 C to +110 C	+1.5 C	10 mV/F
LM35D	0 C to +100 C	+2.0 C	10 mV/F

Note: Temperature range is in degrees Celsius.

The complexity associated with writing software for such nonlinear devices has led many manufacturers to market the linear temperature sensor. Simple and widely used linear temperature sensors include the LM34 and LM35 series from National Semiconductor Corp. They are discussed next.

LM34 and LM35 temperature sensors

The sensors of the LM34 series are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Fahrenheit temperature. See Table 12-7. The LM34 requires no external calibration since it is inherently calibrated. It outputs 10 mV for each degree of Fahrenheit temperature. Table 12-7 is the selection guide for the LM34.

The LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius (centigrade) temperature. The LM35 requires no external calibration since it is inherently calibrated. It outputs 10 mV for each degree of centigrade temperature. Table 12-8 is the selection guide for the LM35 (For further information see www.national.com.).

Signal conditioning and interfacing the LM35 to the 8051

Signal conditioning is a widely used term in the world of data acquisition. The most common transducers produce an output in the form of voltage, current, charge, capacitance, and resistance. However, we need to convert these signals to voltage in order to send input to an A-to-D converter. This conversion (modification) is commonly called *signal conditioning*. Signal conditioning can be a current-to-voltage conversion or a signal amplification. For example, the thermistor changes resistance with temperature. The change of resistance must be translated into voltages in order to be of any use to an ADC. Look at the case of connecting an LM35 to an ADC804. Since the ADC804 has 8-bit resolution with a maximum of 256 (2^8) steps and the LM35 (or LM34) produces 10 mV for every degree of temperature change, we can condition V_{in} of the ADC804 to produce a V_{out} of 2560 mV (2.56V) for full-scale output. Therefore, in order to produce the full-scale V_{out} of 2.56 V for the ADC804, we need to set $V_{ref}/2$ = 1.28. This makes V_{out} of the ADC804 correspond directly to the temperature as monitored by the LM35. See Table 12-9. $V_{ref}/2$ values are given in Table 12-5.

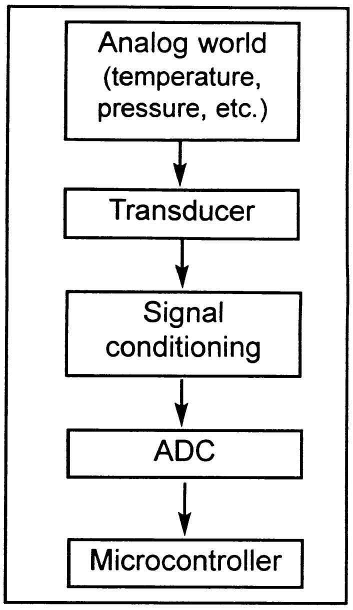


Figure 12-9. Getting Data From the Analog World

Table 12-9. Temperature v. Vout of the ADC804

Temp. (C)	Vin (mV)	Vout (D7 - D0)
0	0	0000 0000
1	10	0000 0001
2	20	0000 0010
3	30	0000 0011
10	100	0000 1010
30	300	0001 1110

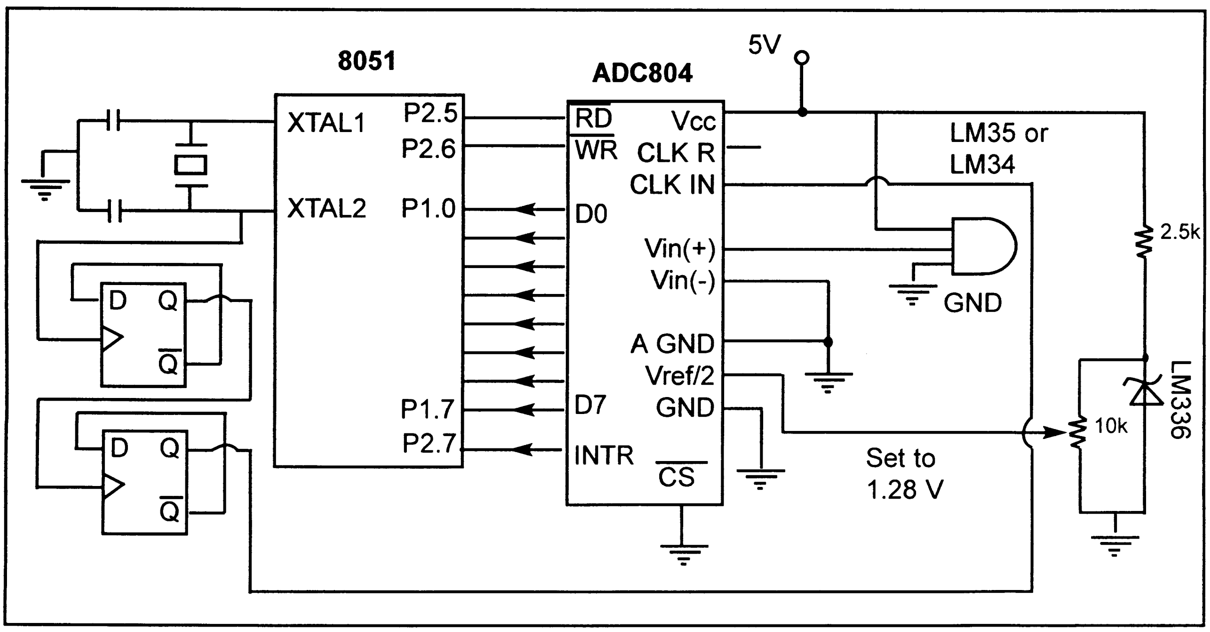


Figure 12-10. 8051 Connection to ADC804 and Temperature Sensor

Figure 12-10 shows connection of a temperature sensor to the ADC804. Notice that we use the LM336-2.5 zener diode to fix the voltage across the 10K pot at 2.5 volts. The use of the LM336-2.5 should overcome any fluctuations in the power supply.

ADC808/809 chip with 8 analog channels

Another useful chip is the ADC808/809 from National Semiconductor. See Figure 12-11. While the ADC804 has only one analog input, this chip has 8 of them. The ADC808/809 chip allows us to monitor up to 8 different transducers using only a single chip. Notice that the ADC808/809 has an 8-bit data output just like the ADC804. The 8 analog input channels are multiplexed and selected according to Table 12-10 using three address pins, A, B, and C.

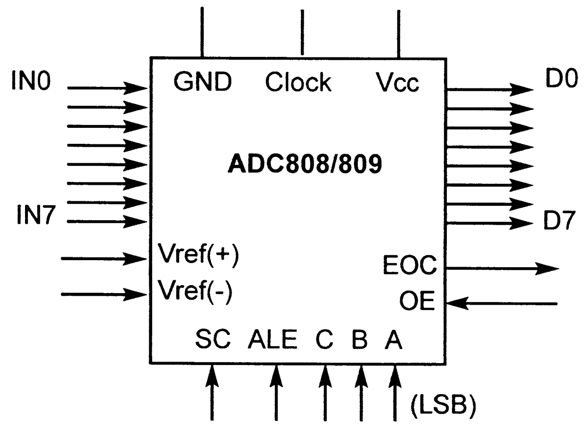


Figure 12-11. ADC808/809

Table 12-10: ADC808 Analog Channel Selection

Selected Analog Channel	C	B	A
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	0	1
IN6	1	1	0
IN7	1	1	1

In the ADC808/809, $V_{ref}(+)$ and $V_{ref}(-)$ set the reference voltage. If $V_{ref}(-)$ = Gnd and $V_{ref}(+)$ = 5 V, the step size is 5 V/256 = 19.53 mV. Therefore, to get a 10 mV step size we need to set $V_{ref}(+)$ = 2.56 V and $V_{ref}(-)$ = Gnd. From Figure 12-11, notice the ALE pin. We use A, B, and C addresses to select IN0 - IN7, and activate ALE to latch in the address. SC is for start conversion. EOC is for end-of-conversion, and OE is for output enable (READ). Next, we give the steps for programming this chip.

Steps to program the ADC808/809

The following are the steps to get data from analog input of ADC808/809 into the microcontroller.

1. Select an analog channel by providing bits to A, B, and C addresses according to Table 12-10.
2. Activate the ALE (address latch enable) pin. It needs an L-to-H pulse to latch in the address.
3. Activate SC (start conversion) by an H-to-L pulse to initiate conversion.
4. Monitor EOC (end of conversion) to see whether conversion is finished. H-to-L output indicates that the data is converted and is ready to be picked up.
5. Activate OE (output enable) to read data out of the ADC chip. An H-to-L pulse to the OE pin will bring digital data out of the chip.

Notice in the ADC808/809 that there is no self-clocking and the clock must be provided from an external source to the CLK pin. Although the speed of conversion depends on the frequency of the clock connected to CLK pin, it cannot be faster than 100 ms.

Review Questions

1. In the ADC804, the INTR signal is an _______ (input, output).
2. To begin conversion, send a(n) ________ pulse to pin _______.
3. Which pin of the ADC804 indicates end-of-conversion?
4. True or false. The transducer must be connected to signal conditioning circuitry before it is sent to the ADC.
5. The LM35 provides ______ mV for each degree of ______ (Fahrenheit, Celsius) temperature.
6. Both the ADC804 and ADC808 are _____-bit converters.
7. Indicate the direction (out, in) for each of the following pins of the ADC808/809.
   (a) A, B, C (b) SC (c) EOC

SUMMARY

This chapter showed how to interface real-world devices such as LCDs, ADC chips, and sensors, to the 8051. First, we described the operation modes of LCDs, then described how to program the LCD by sending data or commands to it via its interface to the 8051.

Next we explored ADC chips and temperature sensors. Getting data from the analog world to a digital device is called *signal conditioning*. It is an essential feature of data acquisition systems.

PROBLEMS

SECTION 12.1: INTERFACING AN LCD TO THE 8051

1. The LCD discussed in this section has _______ (4, 8) data pins.
2. Describe the function of pins E, R/W, and RS in the LCD.
3. What is the difference between the V_{CC} and V_{EE} pins on the LCD?
4. "Clear LCD" is a __________ (command code, data item) and its value is ___ hex.
5. What is the hex value of the command code for "display on, cursor on"?
6. Give the state of RS, E, and R/W when sending a command code to the LCD.
7. Give the state of RS, E, and R/W when sending data character 'Z' to the LCD.
8. Which of the following is needed on the E pin in order for a command code (or data) to be latched in by the LCD?
   (a) H-to-L pulse (b) L-to-H pulse
9. True or false. For the above to work, the value of the command code (data) must be already at the D0 - D7 pins.
10. In sending streams of characters to the LCD there are two methods: (1) checking the busy flag, or (2) putting some time delay in between sending each character without checking the busy flag. Explain the difference and the advantages and disadvantages of each method. Also explain how we monitor the busy flag.
11. For a 16x2 LCD, the location of the last character of line 1 is 8FH (its command code). Show how this value was calculated.
12. For a 16x2 LCD, the location of the first character of line 2 is C0H (its command code). Show how this value was calculated.
13. For a 20x2 LCD, the location of the last character of line 2 is 93H (its command code). Show how this value was calculated.
14. For a 20x2 LCD, the location of the third character of line 2 is C2H (its command code). Show how this value was calculated.
15. For a 40x2 LCD, the location of the last character of line 1 is A7H (its command code). Show how this value was calculated.
16. For a 40x2 LCD, the location of the last character of line 2 is E7H (its command code). Show how this value was calculated.
17. Show the value (in hex) for the command code for the 10th location, line 1 on a 20x2 LCD. Show how you got your value.
18. Show the value (in hex) for the command code for the 20th location, line 2 on a 40x2 LCD. Show how you got your value.
19. Rewrite the COMNWRT subroutine. Assume connections P1.4 = RS, P1.5 = R/W, P1.6 = E.
20. Repeat the Problem 19 for the data write subroutine. Send the string "Hello" to the LCD with checking the busy flag. Use instruction MOVC.

21. Give the status of CS and WR in order to start conversion for the ADC804.
22. Give the status of CS and WR in order to get data from the ADC804.
23. In the ADC804, what happens to the converted analog data? How do we know if the ADC is ready to provide us the data?
24. In the ADC804, what happens to the old data if we start conversion again before we pick up the last data?
25. In the ADC804, INTR is an _______ (input, output) signal. What is its function in the ADC804?
26. For an ADC804 chip, find the step size for each of the following $V_{ref}/2$ values.
    (a) $V_{ref}/2 = 1.28$ V (b) $V_{ref}/2 = 1$ V (c) $V_{ref}/2 = 1.9$ V
27. In the ADC804, what should be the $V_{ref}/2$ value for a step size of 20 mv?
28. In the ADC804, what should be the $V_{ref}/2$ value for a step size of 5 mv?
29. In the ADC804, what is the role of pins $V_{in}(+)$ and $V_{in}(-)$?
30. With a step size of 19.53 mV, what is the analog input voltage if all outputs are 1?
31. With $V_{ref}/2 = 0.64$ V, find the V_{in} for the following outputs.
    (a) D7 - D0 = 11111111 (b) D7 - D0 = 10011001 (c) D7 - D0 = 1101100
32. What does it mean when it is said that a given sensor has a linear output?
33. The LM34 sensor produces _______ mV for each degree of temperature.
34. What is signal conditioning?
35. What is the purpose of the LM336 Zener diode around the pot setting the $V_{ref}/2$ in Figure 12-10?
36. True or false. ADC804 is an 8-bit ADC.
37. Indicate the direction (in, out) for each of the following ADC808 pins.
    (a) SC (b) EOC (c) A, B, C
    (d) ALE (e) OE (f) IN0 - IN7
    (g) D0 - D7
38. Explain the role of the ALE pin in the ADC808 and show how to select channel 5 analog input.
39. In the ADC808, assume $V_{ref}(-)$ = Gnd. Give the $V_{ref}(+)$ voltage value if we want the following step sizes:
    (a) 20 mv (b) 5 mv (c) 10 mv
    (d) 15 mv (e) 2 mv (f) 25 mv
40. In the ADC808, assume $V_{ref}(-)$ = Gnd. Find the step size for the following values of $V_{ref}(+)$:
    (a) 1.28 V (b) 1 V (c) 0.64 V

ANSWERS TO REVIEW QUESTIONS

SECTION 12.1: INTERFACING AN LCD TO THE 8051

1. Input	2. Input	3. H-to-L
4. High	5. 80H and C0H	

SECTION 12.2: 8051 INTERFACING TO ADC, SENSORS

1. Output	2. L-to-H, WR	3. INTR
4. True	5. 10, both.	
6. 8	7. (a) all in (b) in (c) out	