ADC, DAC AND SENSOR INTERFACING

ANALOG-TO-DIGITAL CONVERTER (ADC) INTERFACING:

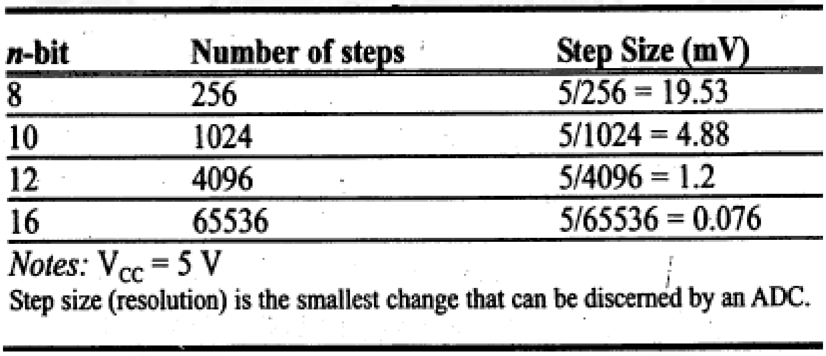
➢ ADCs (Analog-to-Digital Converters) are among the most widely used devices for data acquisition.

➢ A physical quantity, like temperature, pressure, humidity, and velocity, etc., is converted to electrical (voltage, current) signals using a device called a transducer or sensor

➢ We need an Analog-to-Digital Converter to translate the analog signals to digital numbers, so microcontroller can read and process them.

➢ An ADC has n-bit resolution where n can be 8, 10, 12, 16 or even 24 bits.

➢ The higher-resolution ADC provides a smaller step size, where step size is the smallest change that can be discerned by an ADC. This is shown in Table 5.5.1.



n-bit	Number of steps	Step Size (mV)
8	256	5/256 = 19.53
10	1024	5/1024 = 4.88
12	4096	5/4096 = 1.2
16	65536	5/65536 = 0.076

Notes: V_{CC} = 5 V
Step size (resolution) is the smallest change that can be discerned by an ADC.

Table 5.5.1 Resolution Vs Step Size for ADC

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay]

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 parallel ADC, we have 8 of more pins dedicated to bringing out the binary data, but in serial ADC we have only one pin for data out.

ADC804 CHIP:

ADC804 IC is an 8-bit parallel analog-to-digital converter.

➢ It works with +5 volts and has a resolution of 8bits.

➢ In ADC804 conversion time varies depending on the clocking signals applied to the CLK R and CLK IN pins, but it cannot be faster than 110µs.

➢ Figure 5.5.1 is the Pin out of ADC0804 in free running mode.

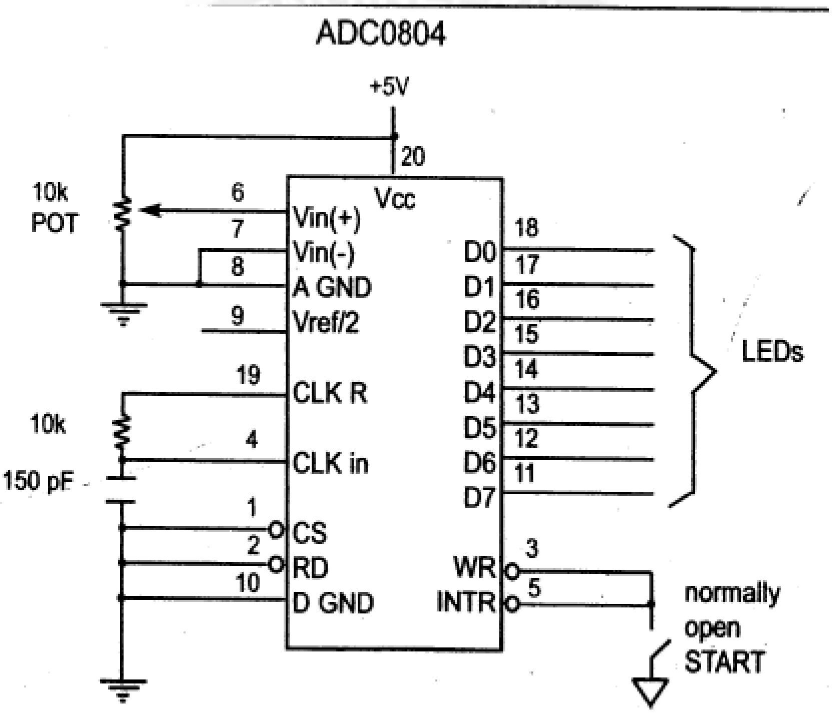


Figure 5.5.1 ADC0804 Chip (Testing ADC0804 in Free Running Mode)

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.375]

CLK IN is an input pin connected to an external clock source when an external clock is used for timing.

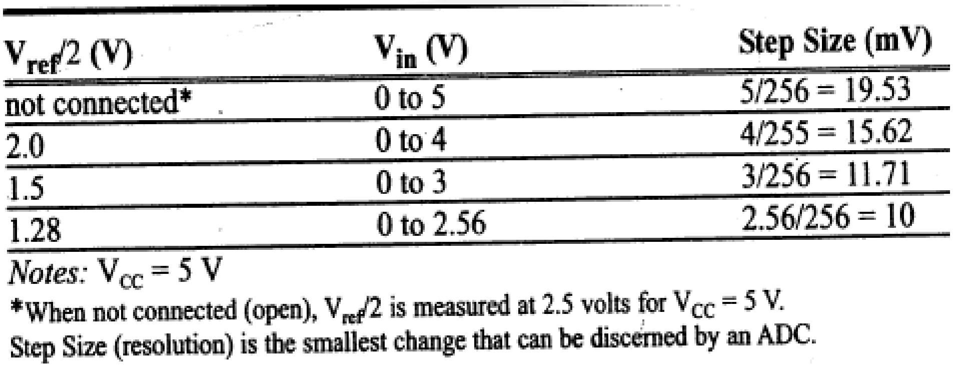
- However, the 0804 has an internal clock generator.
- To use the internal clock generator (also called self-clocking), CLK IN and CLK R pins are connected to a capacitor and a resistor and the clock frequency is determined by:

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

Vref/2: (Pin 9)

- It is used for the reference voltage.
- If this pin is open (not connected), the analog input voltage is in the range of 0 to 5 volts (the same as the Vcc pin).
- If the analog input range needs to be 0 to 4 volts, Vref/2 is connected to 2 volts.

Table 5.5.2 shows the Vin range for various Vref/2 inputs.



$V_{ref}/2$ (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

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.

□□

Table 5.5.2 Vref/2 Relation to Vin Range (ADC0804)

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay]

D0-D7 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 Dout = digital data output (in decimal), Vin = analog voltage, and Step size (resolution) is the smallest change, which is (2 * Vref/2)/256 for ADC 0804

Analog ground is connected to the ground of the analog Vin and digital ground is connected to the ground of the Vcc pin.

• The reason that to have ground pin is to isolate the analog Vin signal from transient voltages caused by digital switching of the output D0 – D7. This contributes to the accuracy of the digital data output.

- Differential analog inputs where Vin= Vin (+) – Vin (-).

• **Vin (-)** is connected to ground and **Vin(+)** is used as the analog input to be converted.

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 ADC0804 chip.

• IF "output enable" ,a high-to-low RD pulse is used to get the 8-bit converted data out of ADC804.

INTR is "end of conversion" .When the conversion is finished; it goes low to signal the CPU that the converted data is ready to be picked up.

□**WR** is an active low input

• It is "start conversion" When WR makes a low-to-high transition, ADC804 starts converting the analog input value of Vin to an 8-bit digital number.

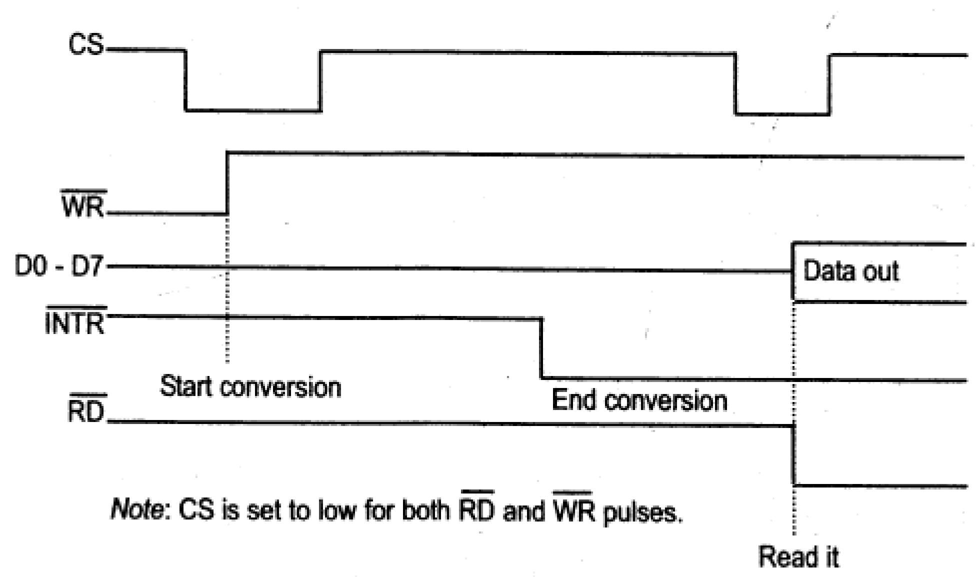
• When the data conversion is complete, the INTR pin is forced low by the ADC0804.

STEPS TO BE FOLLOWED FOR DATA CONVERSION:

The following steps must be followed for data conversion by the ADC804 chip:

- Make CS= 0 and send a L-to-H pulse to pin WR to start conversion.
- Monitor the INTR pin, if high keep polling but if low, conversion is complete, go to next step.
- Make CS= 0 and send a H-to-L pulse to pin RD to get the data out.

The Read and Write Timing for ADC0804 is shown in Figure 5.5.2



Note: CS is set to low for both $\overline{RD}$ and $\overline{WR}$ pulses.

Figure 5.5.2 Read and Write Timing for ADC0804

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.377]

8051 Connection to ADC0804 with Self-Clocking is shown in Figure 5.5.3.

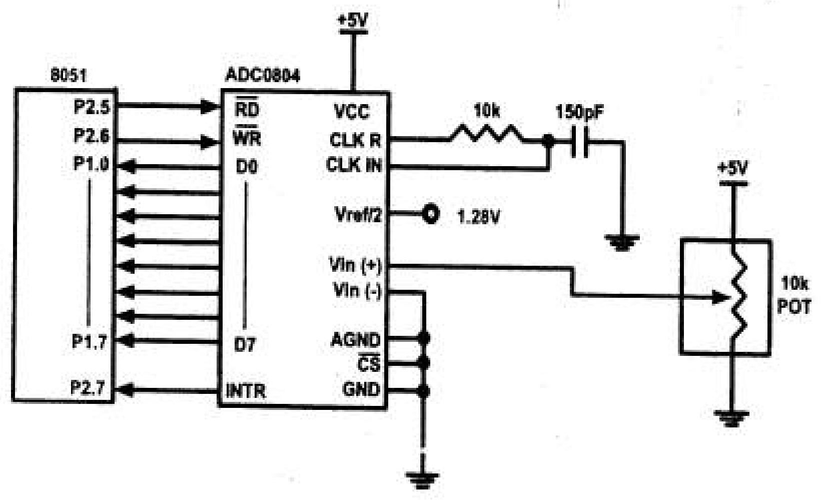


Figure 5.5.3 8051 Connection to ADC0804 with Self-Clocking

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.378]

Example:

Write a program to monitor the INTR pin and bring an analog input into register A. Then call a hex-to ACSII conversion and data display subroutines. Do this continuously.

;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 ADC0804

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
```

DIGITAL-TO-ANALOG (DAC) CONVERTER:

The two method of creating a DAC is binary weighted and R/2R ladder.

The Binary Weighted DAC, which contains one resistor or current source for each bit of the DAC connected to a summing point. These precise voltages or currents sum to the correct output value. This is one of the fastest conversion methods but suffers from poor accuracy because of the high precision required for each individual voltage or current. Such high-precision resistors and current-sources are expensive, so this type of converter is usually limited to 8-bit resolution or less.

The R-2R ladder DAC, which is a binary weighted DAC that uses a repeating cascaded structure of resistor values R and 2R. This improves the precision due to the relative ease of producing equal valued matched resistors (or current sources). However, wide converters perform slowly due to increasingly large RC-constants for each added R-2R link.

The first criterion for judging a DAC is its resolution, which is a function of the number of binary inputs. The common ones are 8, 10, and 12 bits. The number of data bit inputs decides the resolution of the DAC since the number of analog output levels is equal to 2n, where n is the number of data bit inputs. Therefore, an 8-input DAC such as the DAC0808 provides 256 discrete voltage (or current) levels of output. Similarly, the 12-bit DAC provides 4096 discrete voltage levels. There also 16-bit DACs, but they are more expensive.

DAC 0808:

➢ The digital inputs are converter to current (Iout), and by connecting a resistor to the Iout pin, we can convert the result to voltage.

➢ The total current provided by the Iout pin is a function of the binary numbers at the D0-D7 inputs of the DAC0808 and the reference current (Iref), and is as follows

$$I_{out} = I_{ref} \left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \frac{D3}{32} + \frac{D2}{64} + \frac{D1}{128} + \frac{D0}{256} \right)$$

Usually reference current is 2mA.

➢ Ideally, we connect the output pin to a resistor, convert this current to voltage, and monitor the output on the scope.

➢ But this can cause inaccuracy; hence an opamp is used to convert the output current to voltage.

➢ The 8051 connection to DAC0808is as shown in the below Figure 5.5.4.

➢ Now assuming that Iref = 2mA, if all the inputs to the DAC are high, the maximum output current is 1.99mA.

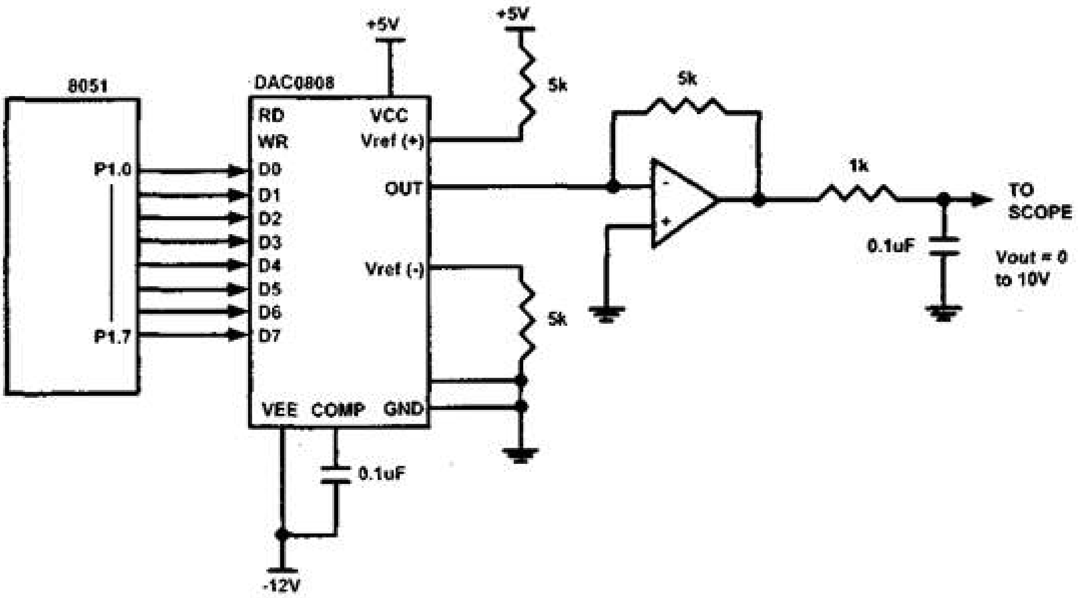


Figure 5.5.4 8051 Connection to DAC808

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.399]

Example 1:

Assuming that R=5K and Iref=2mA, calculate Vout for the following binary inputs:

(a) 10011001B

(b) 11001000B

Solution:

(a) Iout = 2mA(153/256) = 1.195mA and Vout = 1.195mA * 5K =5.975V

(b) Iout = 2mA(200/256) = 1.562mA and Vout = 1.562mA * 5K =7.8125V

CONVERTING IOUT TO VOLTAGE IN DAC0808:

➢ Ideally we connect the output pin lout, to a resistor, convert this current to voltage, and monitor the output on the scope.

➢ In real life, however, this can cause inaccuracy since the input resistance of the load where it is connected will also affect the output voltage.

➢ For this reason, the lref current output is isolated by connecting it to an op-amp such as the 741 with Rf = 5K ohms for the feedback resistor.

➢ Assuming that R= 5K ohms, by changing the binary input, the output voltage changes as shown in Example 2.

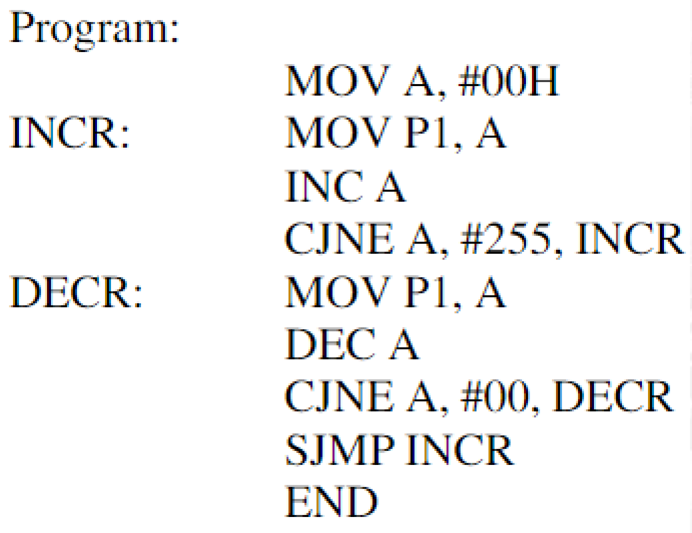
Example 2:

In order to generate a stair-step ramp, set up the circuit in Figure 5.5.4 and connect the output to an oscilloscope. Then write a program to send data to the DAC to generate a stair-step ramp.

```
CLR A
AGAIN: MOV P1,A             ; SEND DATA TO DAC
INC A                       ; COUNT FROM 0 TO FFH
ACALL DELAY                 ; LET DAC RECOVER
SJMP AGAIN
```

Example 3:

Write an ALP to generate a triangular waveform.



```
Program:
              MOV A, #00H
INCR:         MOV P1, A
              INC A
              CJNE A, #255, INCR
DECR:         MOV P1, A
              DEC A
              CJNE A, #00, DECR
              SJMP INCR
              END
```

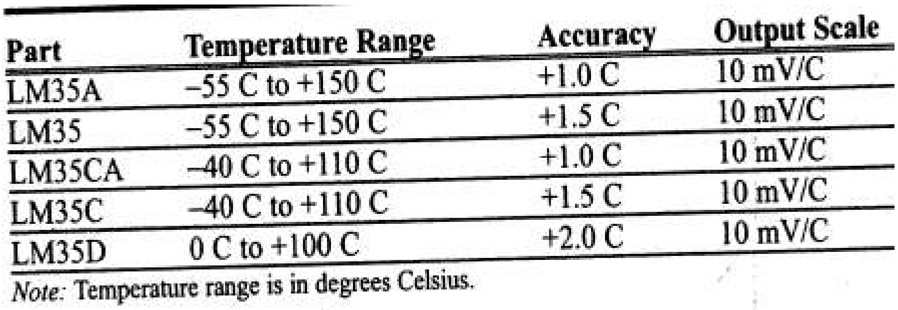
SENSOR INTERFACING:

LM35 TEMPERATURE SENSORS:

➢ 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 internally calibrated.

➢ It outputs 10mV for each degree of centigrade temperature.



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

Note: Temperature range is in degrees Celsius.

Table 5.5.3 LM35 Temperature Sensor Series Selection Guide

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay]

- The sensors of the LM34 series are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Fahrenheit temperature.
- It outputs 10mV for each degree Fahrenheit temperature.

SIGNAL CONDITIONING AND INTERFACING THE LM35 TO THE 8051

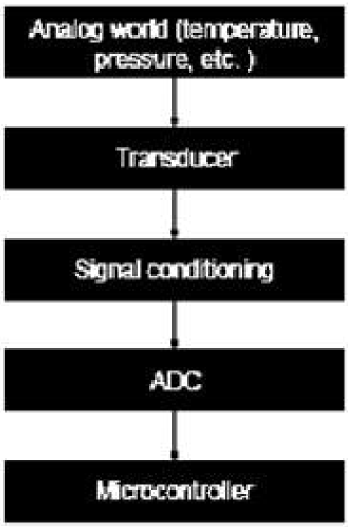


Figure 5.5.5 Getting Data from Analog World

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay]

- Signal conditioning is widely used 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.

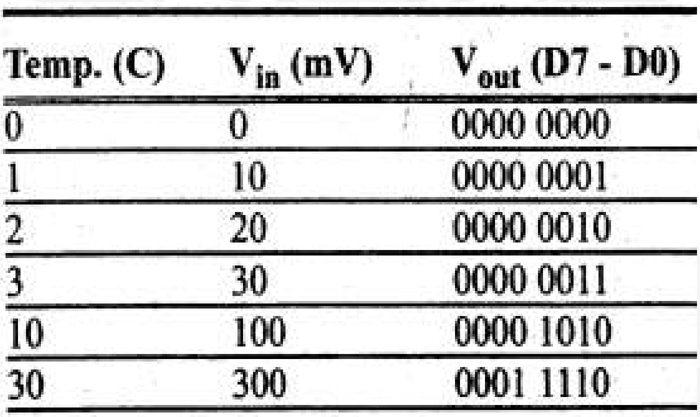
➢ The change of resistance must be translated into voltages in order to be of any use to an ADC.

➢ Look at the case of connecting an LM35 to an ADC0848.

➢ Since the ADC0848 has 8-bit resolution with a maximum of 256 (28) steps and the LM35 (or LM34) produces l0 mV for every degree of temperature change, we can condition Vin of the ADC0848 to produce a Vout, of 2560 mV (2.56 V) for full-scale output.

➢ Therefore, in order to produce the full-scale Vout of 2.56 V for the ADC0848, we need to set Vref = 2.56.

➢ This makes Vout, of the ADC0848 correspond directly to the temperature as monitored by the LM35. Refer the Table 5.5.4



Temp. (C)	V_{in} (mV)	V_{out} (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

Table 5.5.4 Temperature vs. Vout for ADC0848

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay]

Figure 5.5.5 shows the connection of a temperature sensor to the ADC0848.

➢ The LM336-2.5 zener diode to fix the voltage across the 10K pot at 2.5V.

➢ The use of the LM336-2.5 should overcome any fluctuations in the power supply.

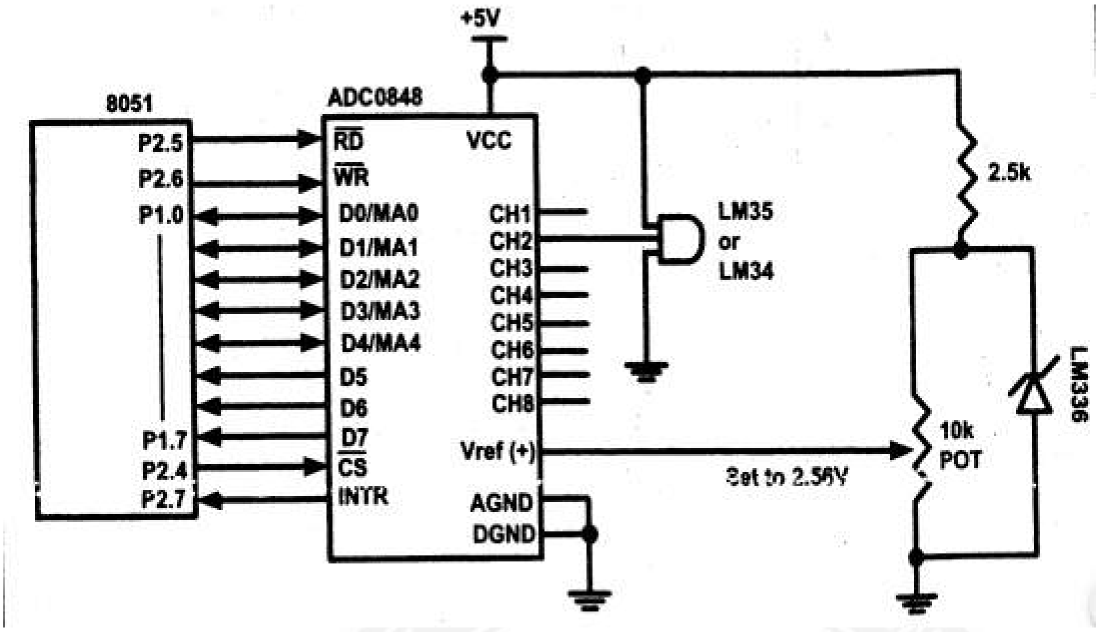


Figure 5.5.6 8051 Connection to ADC0848 and Temperature sensor

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.405]

Program:

```
RD BIT P2.5                    ;RD
WR BIT P2.6                    ;WR
INTR BIT P2.7                  ; END OF CONVERSION

MYDATA EQU P1 ; P1.0-P1.7 = D0-D7 OF THE ADC0848
MOV P1,#0FFH                   ;make P1 = input
SETB INTR
BACK: CLR WR                   ;WR = 0
SETB WR                        ;WR = 1 L-to-H to start conversion
HERE: JB INTR, HERE            ;wait for end of conversion
CLR RD                         ;conversion finished, enable RD
MOV A,MYDATA                    ;read the data
ACALL CONVERSION               ;hex-to-ASCII conversion
ACALL DATA_DISPLAY             ;display the data
SETB RD                        ;make RD=1 for next round
SJMP BACK
```

CONVERSION:

```
MOV B,#10
DIV AB
MOV R7,B
MOV B,#10
DIV AB
MOV R6,B
MOV R5,A
RET
DATA_DISPLAY:
MOV P0,R7
ACALL DELAY
MOV P0,R6
ACALL DELAY
MOV P0,R5
ACALL DELAY
RET
```