

## EE1000 Project 4

### Digital Volt Meter

#### Overview

In this project, we make a device that can measure voltages in the range of 0 to 4 Volts with one digit of accuracy. The input is an analog voltage and the output is a single 7-segment digit that tells us what that input's voltage is to the nearest volt

#### Design

Recall that when we are faced with a design problem, we need to follow these design steps:

1. Understand the concept.
2. Understand or create an algorithm to do the job – We'll discuss this, below.
3. If the problem is too complicated, subdivide and repeat, otherwise, design it.
4. Test.

#### Approach (Algorithm)

One way to make (implement) this project is:

1. Convert the analog signal to digital.
2. Convert the digital data into binary data.
3. Decode the binary data for the 7-segment display.

This algorithm can be illustrated by a block diagram (See Figure 1).

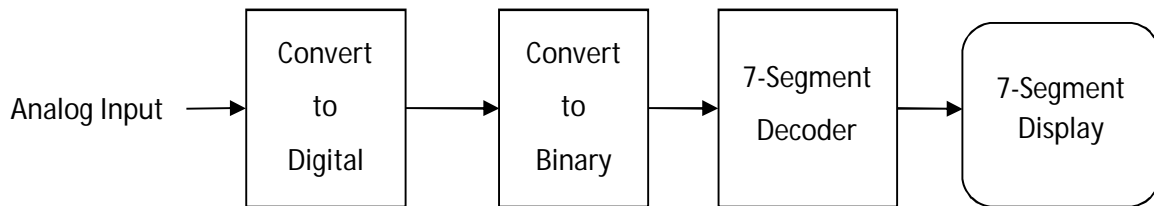


Figure 1 – Block Diagram

Recall that the arrows between the blocks represent information, not necessarily wires.

We will want to subdivide this design into 3 parts: (a) the digital conversion, (b) the binary conversion and (c) the 7-segment decoder and display.

#### Digital Conversion

The easiest way to convert an analog signal to digital is to use a comparator. A comparator takes two input voltages (labeled + and -) and outputs a high digital signal if the (+) input has a higher voltage than the (-) input, otherwise it outputs a low signal. We will need 4 comparators to compare the input to 0.5V, 1.5V, 2.5V and 3.5V as shown in Figure 2. The output of these comparators are digital signals we will call  $C_{0.5}$ ,  $C_{1.5}$ ,  $C_{2.5}$  and  $C_{3.5}$  respectively.

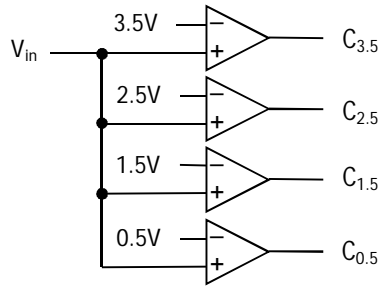


Figure 2 – Converting an Analog Signal to Digital.

For this project, we will be using the MCP6544 quad comparator (4 comparators per chip). A pin diagram is included at the end of this document. Based on the input voltage,  $V_{in}$ , the output signals will respond as shown in Figure 3.

Input Voltage	$C_{3.5}$	$C_{2.5}$	$C_{1.5}$	$C_{0.5}$
$0.0 < V_{in} < 0.5$	0	0	0	0
$0.5 < V_{in} < 1.5$	0	0	0	1
$1.5 < V_{in} < 2.5$	0	0	1	1
$2.5 < V_{in} < 3.5$	0	1	1	1
$3.5 < V_{in}$	1	1	1	1

Figure 3 – Comparator Outputs Based on Input Voltage

All that remains is to generate the fixed input voltages (0.5V through 3.5V). The easiest way to do this is by means of a *Voltage Divider*. A voltage divider is simply a bank of series resistors designed to produce one or more specific voltages. Since we have a regulated 5 Volt supply, we will use that for the Voltage Divider as shown in Figure 4.

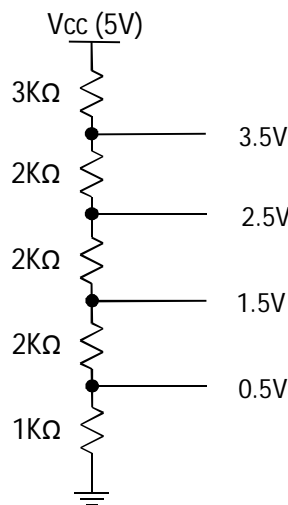


Figure 4 – Voltage Divider

The rule for the voltage divider is that the ratio of a resistor to the total resistance equals the ratio of the voltage across that resistor to the supply voltage. In other words,

$$\frac{V_n}{V_{total}} = \frac{R_n}{R_{total}}$$

In this case,  $V_{total} = 5V$  and  $R_{total} = 10K$ . If you work out the math, the voltage across the  $1K\Omega$  resistor is  $0.5V$ , the voltage across each  $2K\Omega$  resistor is  $1.0V$  and the voltage across the  $3K\Omega$  resistor is  $1.5V$ . These voltage drops give us the various fixed voltages we need for the comparators. Notice that the resistor values can be scaled by any factor (e.g. doubled or halved) without affecting the voltages that are produced.

### Conversion to Binary

Since the voltages we wish to display are in the range 0-4, it will be necessary for us to use a 3-bit signal to represent the voltage. We will call those bits  $B_2$ ,  $B_1$  and  $B_0$ , with  $B_0$  being least significant (i.e. the 1's bit). The binary signal needs to be generated based on the inputs  $C_{0.5}$ ,  $C_{1.5}$ ,  $C_{2.5}$  and  $C_{3.5}$  as shown in Figure 5.

$C_{3.5}$	$C_{2.5}$	$C_{1.5}$	$C_{0.5}$	$B_2$	$B_1$	$B_0$	Display
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	2
0	1	1	1	0	1	1	3
1	1	1	1	1	0	0	4

Figure 5 – Comparator Outputs Based on Input Voltage

The simplified logic equations for the 3-bit binary signals are:

$$B_0 = C_{2.5} \overline{C_{3.5}} + C_{0.5} \overline{C_{1.5}}$$

$$B_1 = C_{1.5} \overline{C_{3.5}}$$

$$B_2 = C_{3.5}$$

You will need to design a logic circuit (much as you did in Project 1) to implement these equations. Your inputs and outputs are active high. (If active low inputs are desired, you can exchange the + and – inputs in Figure 2.)

### 7-Segment Decoder and Display

Decoding binary data for 7-segment displays is much like decoding binary data for the dice pips in Project 1, and we can certainly decode each of the segments with NAND and NOR gates, but fortunately an integrated circuit (IC) is available that can decode all 7 segments at once. The part number for that IC is 74LS47. The 74LS47 takes a 4-bit binary number as input. (Since we have only 3 bits, we will need to connect the 4<sup>th</sup> input to ground.) It also has a ripple blanking input

(RBI) and a LED test (LT) inputs, both of which must be high (e.g. connected to Vcc) for the device to function properly. The 74LS47 has seven outputs labeled a-g, all active low, to drive the seven segments of the display. A typical circuit using the 74LS47 is shown in Figure 6. Pin diagrams for the 74LS74 and the display are available at the end of this document.

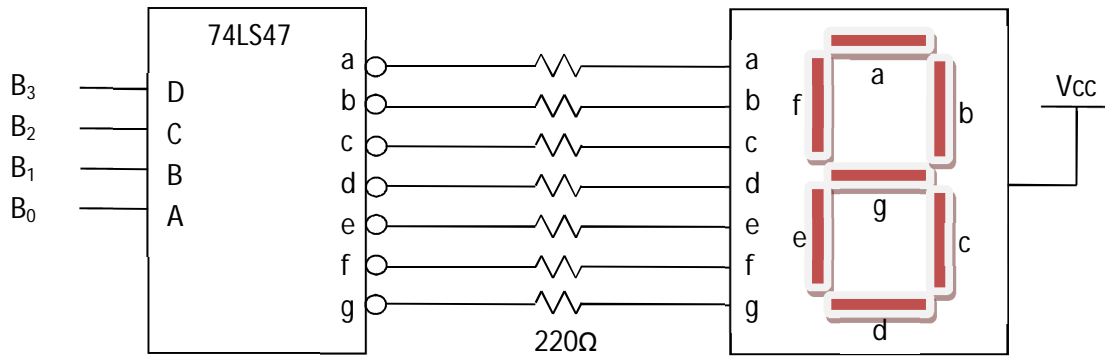


Figure 5 – 7-Segment Decoder and Display

### Testing

Use one channel of the power supply to provide power to your circuit through the 5V regulator. Connect another power supply to V<sub>in</sub>. Be very careful not to drive V<sub>in</sub> over 5 Volts or under 0 Volts, as this may damage the comparator.

Vary the voltage on V<sub>in</sub> from 0 to 4V and verify that the display shows the voltage (to the nearest volt). Note that minor errors can occur because the resistors you used to make the voltage divider have a tolerance of 5%.

## Pin Diagrams

