

# Measure resistance and capacitance without an A/D

AN449

Author: Don Sherman

This article describes how to measure values of a variable resistor or a variable capacitor using a microcontroller to measure the RC charge time to a constant threshold voltage. A variety of these "one-shot" circuits are described and the accuracy of the various methods is measured. Sample programs for the 87C51 family are presented at the end of the article.

The old "one-shot" monostable multivibrator circuit has been around since the discovery of the electron. Popular IC versions of this variable pulse width generator include the 74123 and the famous (Philips) Signetics NE555 timer. These circuits use the time to charge a capacitor to a predefined voltage as their time delay. If the threshold voltage is  $1-1/e$  ( $e$  is the natural log base), then the time delay in seconds is exactly the product of the resistance in ohms and the capacitance in Farads ( $t=RC$ ). This is a linear equation and it can be used to measure an unknown value of  $R$  if  $C$  and  $t$  are known. The purpose of this article is to describe ways to measure a RESISTANCE by using this technique with an 80C51 type of microcontroller.

Examining the voltage across the Capacitor versus time graph in Figure 1, it is obvious, I hope, that the time to reach any given voltage will be directly proportional to the value of  $R$ . In fact the equation for the delay time becomes:

$$V_C = V_{CC} (1 - e^{-t/RC})$$

From this equation, it follows that:

$$V_C/V_{CC} = 1 - e^{-t/RC}$$

and

$$1 - V_C/V_{CC} = e^{-t/RC}$$

Since we are solving for time,  $t$ :

$$\ln(1 - V_C/V_{CC}) = -t/RC$$

so that

$$t = RC \{-\ln(1 - V_C/V_{CC})\}$$

or

$$t = RC (K)$$

where

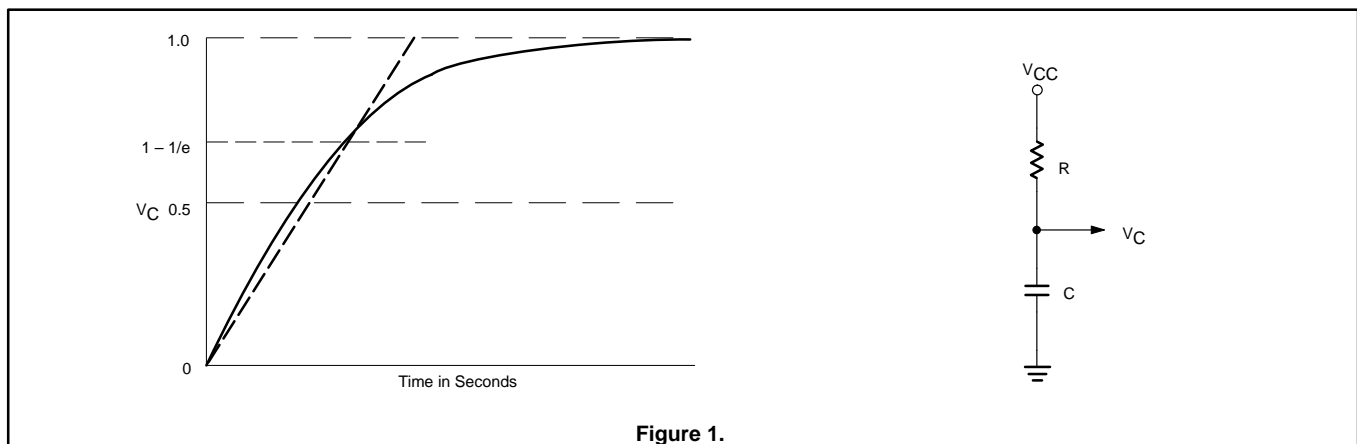
$$K = 1/\{-\ln(1 - V_C/V_{CC})\}.$$

Note that  $K$  becomes a constant if  $V_C/V_{CC}$  is held constant. This ratio of voltages is determined by the threshold of the comparator which is connected to monitor the voltage,  $V_C$ , across the capacitor. Thus, for a constant threshold voltage for  $V_C$  and a constant value for the capacitance,  $C$ , the equation for TIME reduces to:

$$t = R \times \text{"a constant"}$$

which is a simple linear equation.

We can see from the above equations that any point is valid as the threshold voltage as long as it remains the same for each set of readings. The threshold for the LM324 comparator, shown in Figure 2, is  $2/3 V_{CC}$ . If 74HCTxx type of digital logic is used, the threshold will be approximately 1.2 volts. If 74HCxx logic is used the threshold will be approximately 2.5 volts. However, if the threshold is different than  $1-1/e$ , and/or, if the capacitance is not known exactly, it will be necessary to calibrate the system with a known resistance.



# Measure resistance and capacitance without an A/D

AN449

In this article I have investigated different hardware techniques for generating a pulse with the microcontroller, and then the resultant characteristics are summarized. The matrix of variables consists of the following basic one-shot circuits shown in Figures 2 through 12 and various time measurement methods available in common microcontrollers. To be more specific, time was measured by dividing the microcontroller's oscillator frequency with an internal timer and gating a counter on and off by the one-shot. The other method of timing was to use the one-shot time to enable and then stop a software DJNZ counter loop.

In order to obtain some statistical comparison of the different circuits, I wrote an 80C51 assembly language program to generate data for a histogram so that standard deviation of the values could be calculated. The program first took 256 readings of the 16 bit counter/timer and added all of the results to obtain a 3 byte wide answer. The top two bytes were then used as the average of the 16 bit readings. This was the equivalent of dividing the answer by 256 (shifting the 3 byte sum right by 8 bits). In the next sequence a single reading was taken and the average was subtracted from it. The displacement of the answer, which is the deviation from the average, is used to increment a RAM location. RAM locations were allocated -16 and +15 locations around the "average" value's location. This process was repeated for 255 readings. Thus, if 241 readings are the same as the average and five readings are one less than the average and nine readings are one greater than the average, the result is shown in Table 1.

Values beyond the allocated deviation are summarized in the counters at the two ends of the allocated area. If a significant

number of readings appear in these overflow and underflow counters, I included a 16 bit shift routine to reduce the number of bits of the reading until a reasonable spread in values could be plotted. All of the assembly language programs which I used are available from the Philips BBS.

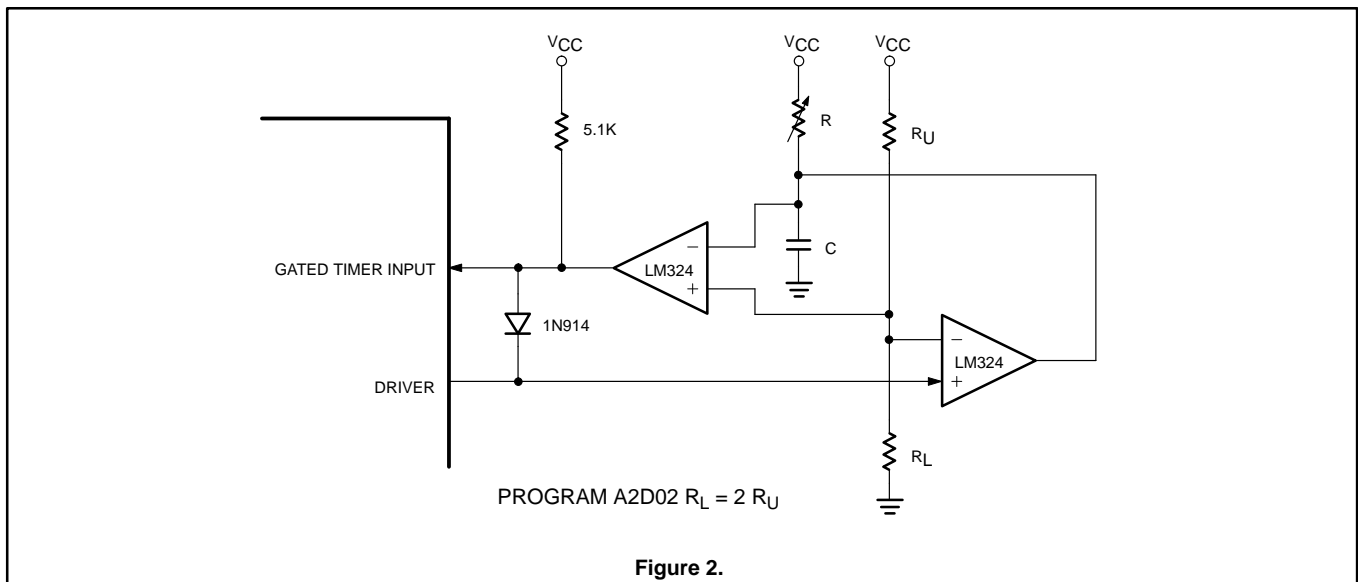
The first thing that I noticed when I started to evaluate the circuits was a strong shift in the center of the readings from the average. By using some freeze mist and an old hair drier, I found that the cheap old monolithic capacitor which I was using as my C, was quite temperature sensitive. So, RULE NUMBER 1 for successful and dependable R readings is to use a capacitor which does not vary with temperature. Polycarbonate, polypropylene, and mylar are good low temperature coefficient dielectrics. These usually come in some high voltage ratings and are also quite large in size compared to the monolithic capacitor. NPO versions of monolithics are available, but be sure to check the temp-co. Don't even think of using an electrolytic capacitor.

The first circuit which I evaluated is shown in Figure 2 and is an attempt to copy the NE555 interior. One half of a LM324 quad comparator was used to implement the circuitry which is external to the microcontroller. The statistical program and the A/D Assembly language program are at the end of this article.

RU and RL were chosen to be in the same ratio as the NE555 timer. This value gives a threshold voltage of  $2/3 V_{CC}$ , which is very close to the  $(1 - 1/e)$  value for a one-shot delay time of  $t = RC$ . The actual NE555 equation is  $t = 1.1 RC$ .

Table 1.

00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	05
F2	09	00	00	00	00	00	00	00	00	00	00	00	00	00	00



# Measure resistance and capacitance without an A/D

## AN449

In this circuit the microcontroller "Driver" outputs a logic 0 to the non-inverting input of the comparator. The open collector output of the comparator follows the input and shorts the timing capacitor, C, to ground. It is important to allow enough time to discharge the capacitor to ground, since any residual charge on the capacitor will cause errors in the readings. The "Gated Timer Input" will allow an internal timer to count signals divided down from the microprocessor clock. The divided down oscillator signals are gated into a counter while the "Gated Timer Input" is held high. The measurement cycle starts when the "Driver" transitions to a logic 1 and simultaneously releases the short to ground on C and, through the diode, allows the Gated Timer Input to be driven high by the other comparator which is monitoring the voltage across the timing capacitor. When the capacitor voltage exceeds the  $\frac{2}{3} V_{CC}$  reference voltage ( $R_L/(R_L+R_U)$ ) on the comparator, the Gated Timer Input is driven low by the comparator and the 16 bit timer is halted.

If only one resistor is to be measured, it is possible to eliminate the diode by using one of the two left over comparators in the LM324. The inverting side should be connected to the junction of RU and RL. The non-inverting input is connected to the "Driver" output. The Open Collector output of the comparator is connected to the "Gated Timer Input". This will hold the Gated Timer Input low while the capacitor is being discharged and will release the Gated Timer Input line to go high at the same time that the Capacitor is released.

Eleven resistor values were used to obtain a spread of timer values which were then plotted to determine the linearity of readings and to also check out the theory that this entire process is really linear. Amazingly enough, the plot was really linear! The spread in the 255 samples per resistor value was greater for larger values of resistance. Using a 0.47 mfd polycarbonate timing capacitor with a 91KΩ R, the average value was 0BEC0H and the deviation table is shown in Table 2.

From these deviation readings we can see that 99% (3 sigma) of the readings are within 5 counts of the average. Shifting the answer right 2 places (dividing by 4) will result in a 14 bit accuracy answer ± 1 bit. This is not too bad a performance for the price of half of a 35 cent IC and a capacitor.

The rest of the circuits which I have shown use a digital input in place of the precision comparator of the first example. As a result,

the accuracy of the resistance reading dropped to the 8 bit range. Readings were taken with the Gated Timer Input method and with a 1 byte wide software DJNZ loop. The accuracy between the two techniques was virtually the same, but the speed of the conversion was four times faster with the timer. The timer counts at 1/12 of the microprocessor oscillator frequency. Thus, a full scale reading of 0FFH would take  $255 \times (1/12\text{MHz}) \times 12 = 255$  microseconds. The software timer loop consists of a loop formed by JB (24 clock cycles) + DJNZ (24 clock cycles) for an oscillator frequency divided by 48 time for 1 count. The equation for the maximum integration time is then:  $255 \times (1/12\text{MHz}) \times 48 = 1.02$  milli-seconds for the software timer.

The first of the digital gate instead of a comparator circuits is shown in Figure 3. Figure 3a shows the implementation of a single channel resistance measuring circuit and Figure 3b shows one possible way to implement a two resistor measuring circuit. The A/D conversion subroutine for Figure 3a is called A2D82.

The timing capacitor is shorted to ground by the open drain digital output of a 74HC05 Hex inverter by making the microcontroller Driver output go to logic 1. Upon the release of the capacitor hold down signal, the other digital gate acts as a high input impedance comparator with a threshold of approximately one half of  $V_{CC}$  if it is CMOS or about 1.4 volts if it is HCT. The two resistor version in Figure 3b is run by alternately shorting out the timing capacitor on one side while starting the timer to measure the opposite side. When the wire NORed circuit sees a logic 1 from Vcap exceeding the inverter threshold, the active inverter pulls the Gated Timer Input low to stop the timer. There will be a slight offset error caused by the nonsimultaneity of releasing the capacitor and the start of the gated timer. This offset will be a constant and can be dealt with in the calibration process by subtracting a constant from the answer. Note that the wired NOR circuit is depending on the quasi-bi-directional I/O port pull up resistor inside of the 80C51 type of microcontroller. If the input to the microcontroller is really open drain, it will be necessary to add a pull-up resistor.

This circuit idea can be extended to a 3 channel circuit using the last two inverters in the 74HC05 IC package. The three channel circuit uses four port pins. Likewise, if two ICs are used, 6 channels can be implemented using seven port pins.

**Table 2.**

00	00	00	00	00	00	00	00	00	00	00	00	01	11	0D	2E
2C	37	1D	17	0E	0A	00	00	00	00	00	00	00	00	00	00

# Measure resistance and capacitance without an A/D

AN449

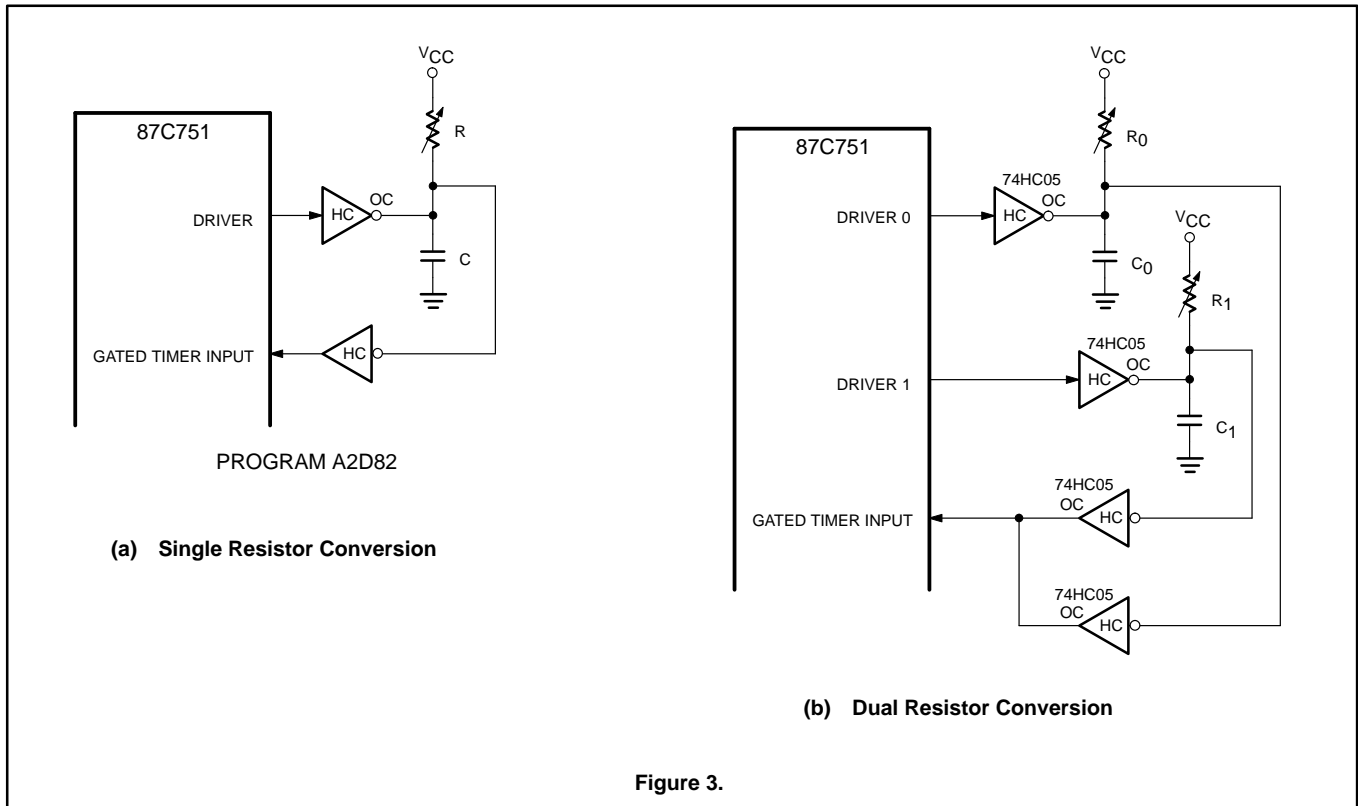


Figure 3.

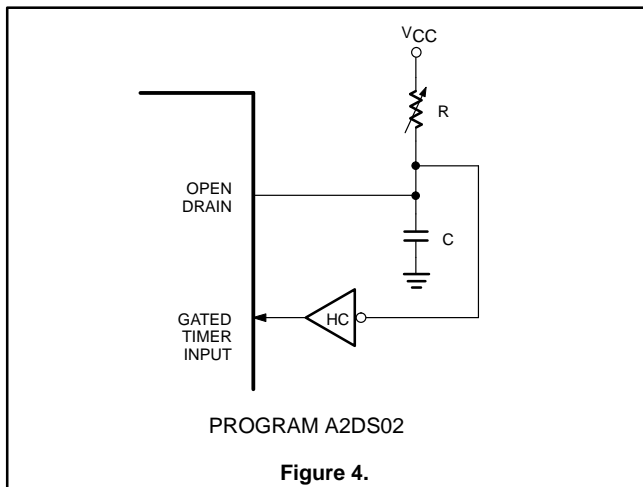


Figure 4.

The previous circuits will work with just about any type of microcontroller port I/O. Some microcontrollers can be programmed to have an open drain output configuration and some port outputs are only available as open drain. Examples of the latter type are port 0 of the 87C51 family and I2C and Vpp pins on some of the 87C51 family derivatives (i.e.. SDA, SCL, Vpp on the 87C751). With these

open drain outputs, the open drain buffer can be eliminated as shown in Figure 4.

Using this configuration, the open drain output is first set to 0 to discharge the capacitor to ground and then the gated timer is enabled and then the open drain output is programmed to 1 as shown in Program "A2DS02", which is on the Philips BBS. The slight offset error caused by the nonsimultaneity of releasing the capacitor and the start of the gated timer can be dealt with in the calibration process by subtracting a constant from the answer.

Figure 5 is an example of how multiple channels can also be created using the Figure 4 configuration. Since there are 8 open drain outputs on the 87C51 Port 0, it is possible to make an 8 channel resistance reader which uses all of port 0's 8 bits and an additional input for the timer for a total of 9 port pins.

In Figure 5, the timer runs from the release of the capacitor hold down to the detection of a low level on the Software Timer Input port pin. The heavy dark lines labeled "Time" indicate the timing periods for each of the four capacitor timing circuits. The detection of a logic high on the one capacitor which is integrating current causes the Nor gate to go low. The software timer halts when the logic low is detected and then the driver transistor is turned on to discharge the capacitor. As soon as the capacitor voltage goes below the threshold for logic one, the Nor gate returns to a high logic level as shown on the "Input" section of Figure 5.

# Measure resistance and capacitance without an A/D

AN449

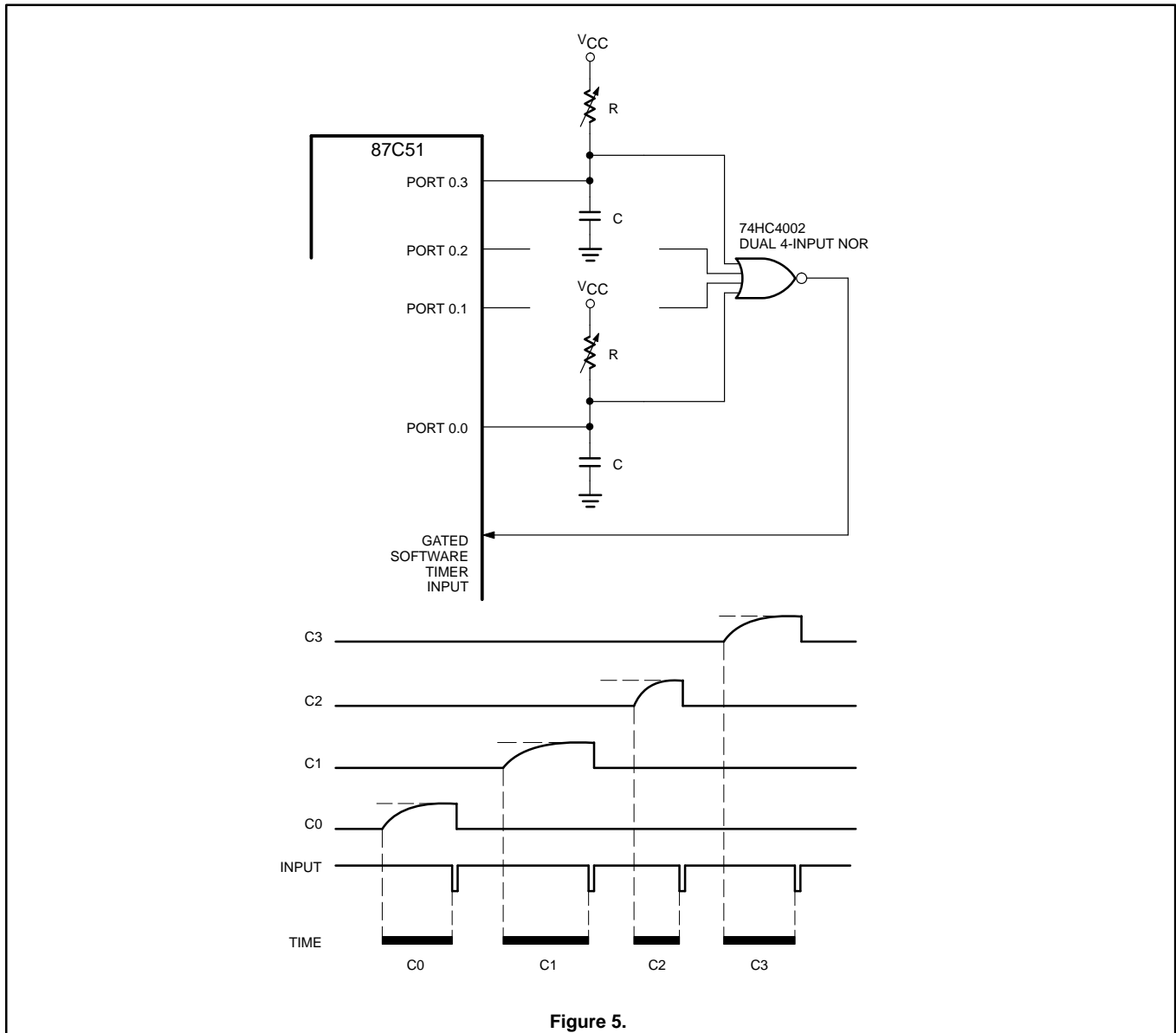


Figure 5.

In Figure 6 we have the simplest embodiment of the RC timer. This configuration will not allow the use of the Gated Timer, since the Gated Timer input pin is not open drain. The software counter technique is used, where a counter is decremented and the input is checked for a logic high (see Figure 7).

When the counter is stopped, the value is inverted to give a count-up appearing output. Ground bounce on the microcontroller caused by the clock distribution is probably the main source of the noise with this configuration. The result is an 8 bit answer, but it is only repeatable to within 5 bits of accuracy.

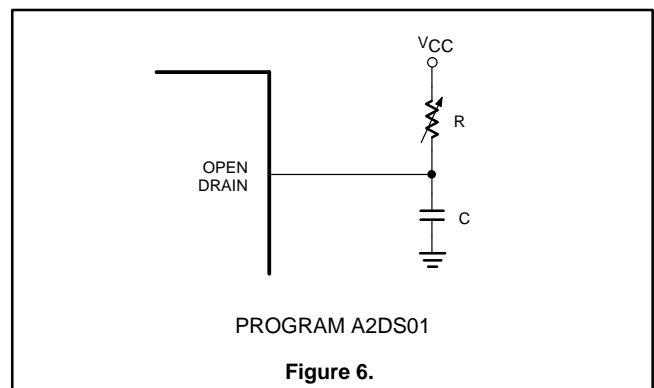


Figure 6.

# Measure resistance and capacitance without an A/D

AN449

```

ADLOOP: JB     SDA,OVER      ;CHECK BIT "SDA" AND JUMP OUT
                                ;OF LOOP WHEN IT = 1
          DJNZ  RESULT,ADLOOP ;DECREMENT "RESULT" AND
                                ;REPEAT. JUST DROP THROUGH IF
                                ;RESULT = 0, SINCE IS FULL
                                ;SCALE.
OVER:     XRL   RESULT,#0FFH ;DO 1'S COMPLEMENT TO GET
                                ;COUNT-UP TYPE RESULTS.
    
```

Figure 7.

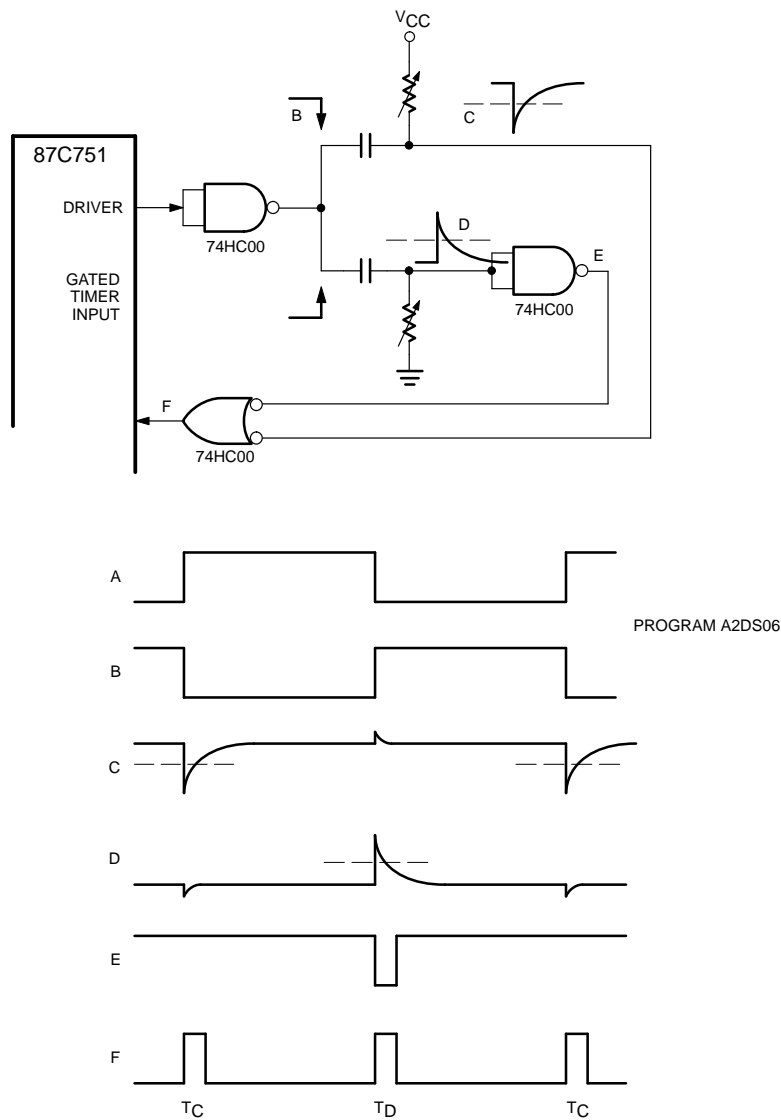


Figure 8.

# Measure resistance and capacitance without an A/D

AN449

Figure 8 operates when a square wave is output by the Driver and then buffered by the 74HC00 to give the waveform shown as B. The falling edge of B will generate the waveform shown in C. The output of the 2 input NAND gate, F, will produce a logic 1 while the input is below the HC threshold voltage of approximately 2.5 volts.

The rising edge of B will be differentiated by the CR circuit and produce the waveform shown in D. The decay back to ground will have the same type of exponential curve as described in the original Figure 1. The decaying waveform, D, is turned into a digital pulse by the inverter's output as the signal, E. Again the two input NAND gate output, F, will go high while the input from E is low.

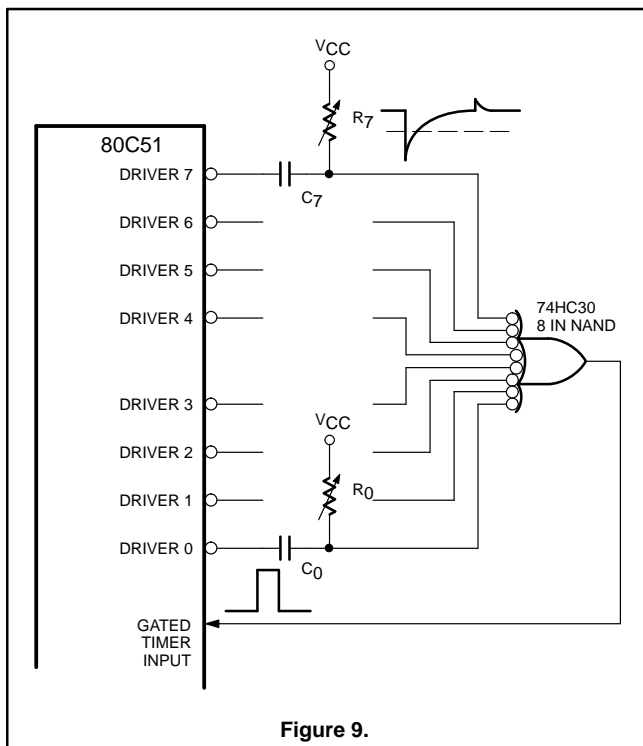


Figure 9.

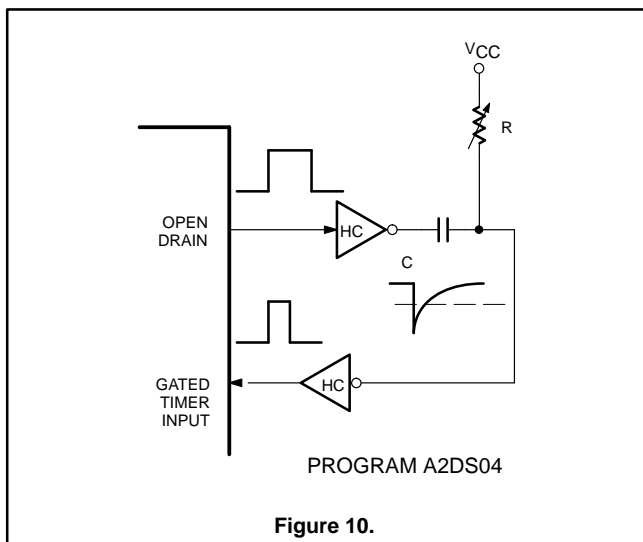


Figure 10.

The constant in the  $t = R \times \text{a constant}$  equation will be different for the C waveform than for the D waveform. However, the readings from C and from D will be linearly correct, but the C and D readings can not be compared and must be separately calibrated. The good news is that the 74HC00 is the most common and probably one of the cheapest digital ICs available.

The circuit in Figure 9 uses the strong pull-down drivers of the 80C51 to produce the waveform shown next to R7. Each of the successive drivers is taken low and held there until the 74HC30's output returns to zero, indicating that the input threshold has been exceeded. At this point, the Gated Timer has stopped counting and the results are saved in RAM and the associated driver is returned to a logic one to allow the capacitor to recover. The test circuit for this type of configuration is shown in Figure 10.

In Figure 11, a 10 output Johnson counter is first reset to Q0 by a long pulse from the driver, and then a sequence of "short" pulses activates the successive RCs. The time that the output of the Johnson counter is high is used to precharge the capacitor, and the "short" pulse must be long enough to allow for the voltage to settle to less than 1 LSB below  $V_{CC}$ . When the counter increments, the previously high output falls to zero and produces the waveform shown on the "C<sub>X</sub>" section. While the voltage on the capacitor is less than the threshold of the 74HC30, it will be perceived as a digital logic low. The NAND gate will produce a pulse as long as one of the inputs is "low". Again, the gated timer input will allow the counter to run as long as the 74HC30 output is high, and then save the 16 bit results in RAM and proceed to generating the next "short" pulse from the driver. Note that the  $R_R C_R$  time constant should be at least three times as long as the longest  $R_X C_X$  time to insure that the Reset threshold is not exceeded.

The circuit in Figure 12 uses a 74HC138 3 to 8 line decoder to produce the negative going edges to turn on the 74HC30 8 in NAND gate. This circuit will do conversions faster than the Figure 11 Johnson Counter, because the capacitor precharging occurs while the 7 other RCs are being read. The not selected outputs of the decoder are at logic one and will precharge the capacitors. When an output is selected by the three Address inputs, A0, A1, and A2, the selected output will go to a logic zero and start the timing sequence with the gated timer input. Note that a sufficient time must be allowed for the capacitor to precharge to less than 1 LSB,  $(V_{CC} - V_{CC}/255)$ , before the output is selected again.

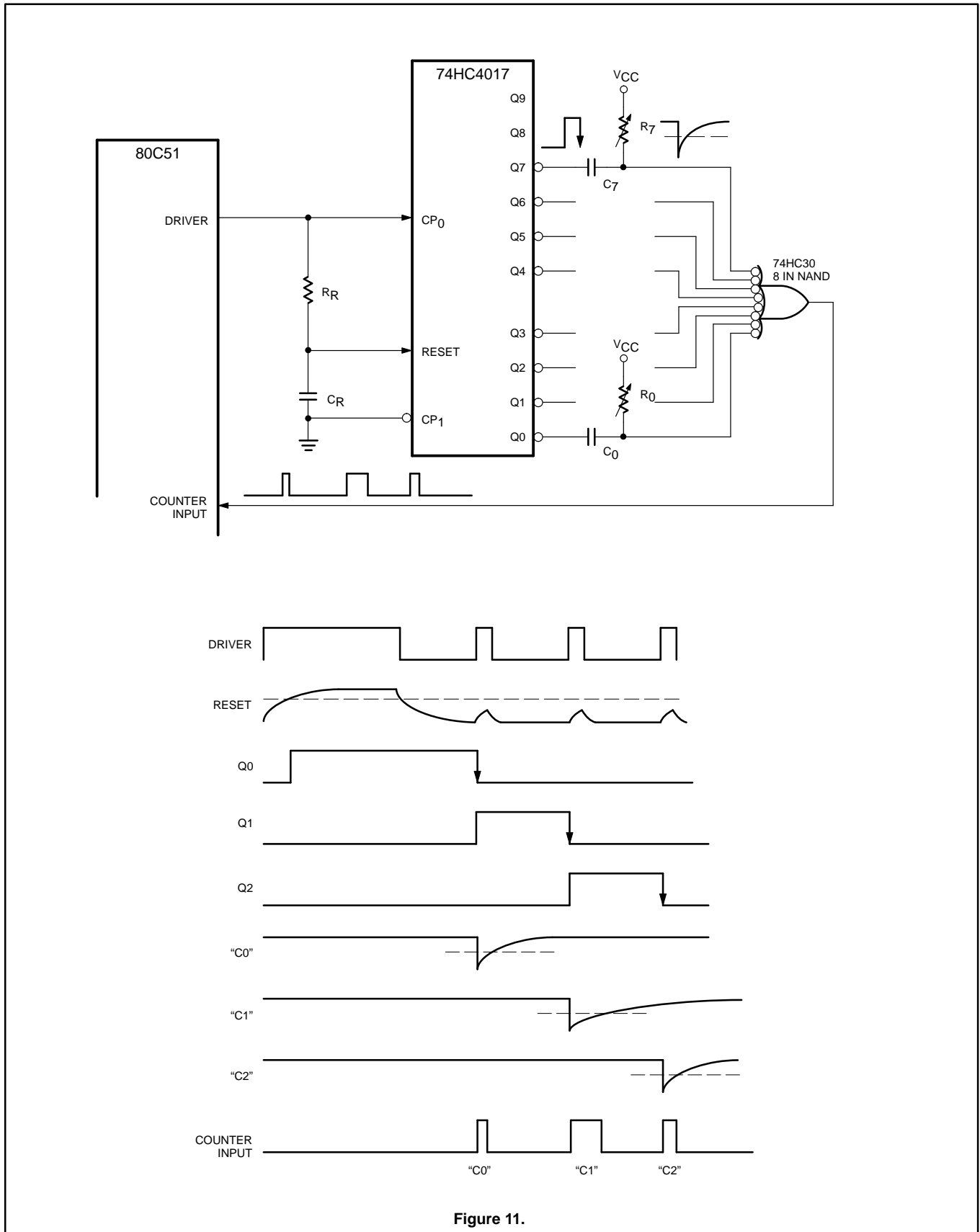
The circuit in Figure 13 is another way of producing a one-shot. The timer is run from the time that the Driver is set to a logic 0 and the time that the Gated Timer Input goes low.

## TEST RESULTS

The accuracy of the readings seemed to depend on the accuracy of the detection of the threshold voltage. The circuit in Figure 2 which used a comparator allowed readings to 3 sigma of  $14 \pm 1$  bit. The circuits which used digital logic gates to detect the threshold voltage, produced readings which were accurate to  $8 \pm 1/4$  bit. For these circuits the results were of the same accuracy whether a software or the internal hardware timer was used. Finally, using the microcontroller's Port pin as the digital threshold detection device, shown in Figure 6, could only produce  $8 \pm 3$  bit answers. As bad as this is, it can still provide useful inputs if truncated to 5 bits and used as a Joy Stick type input.

# Measure resistance and capacitance without an A/D

AN449





# Measure resistance and capacitance without an A/D

AN449

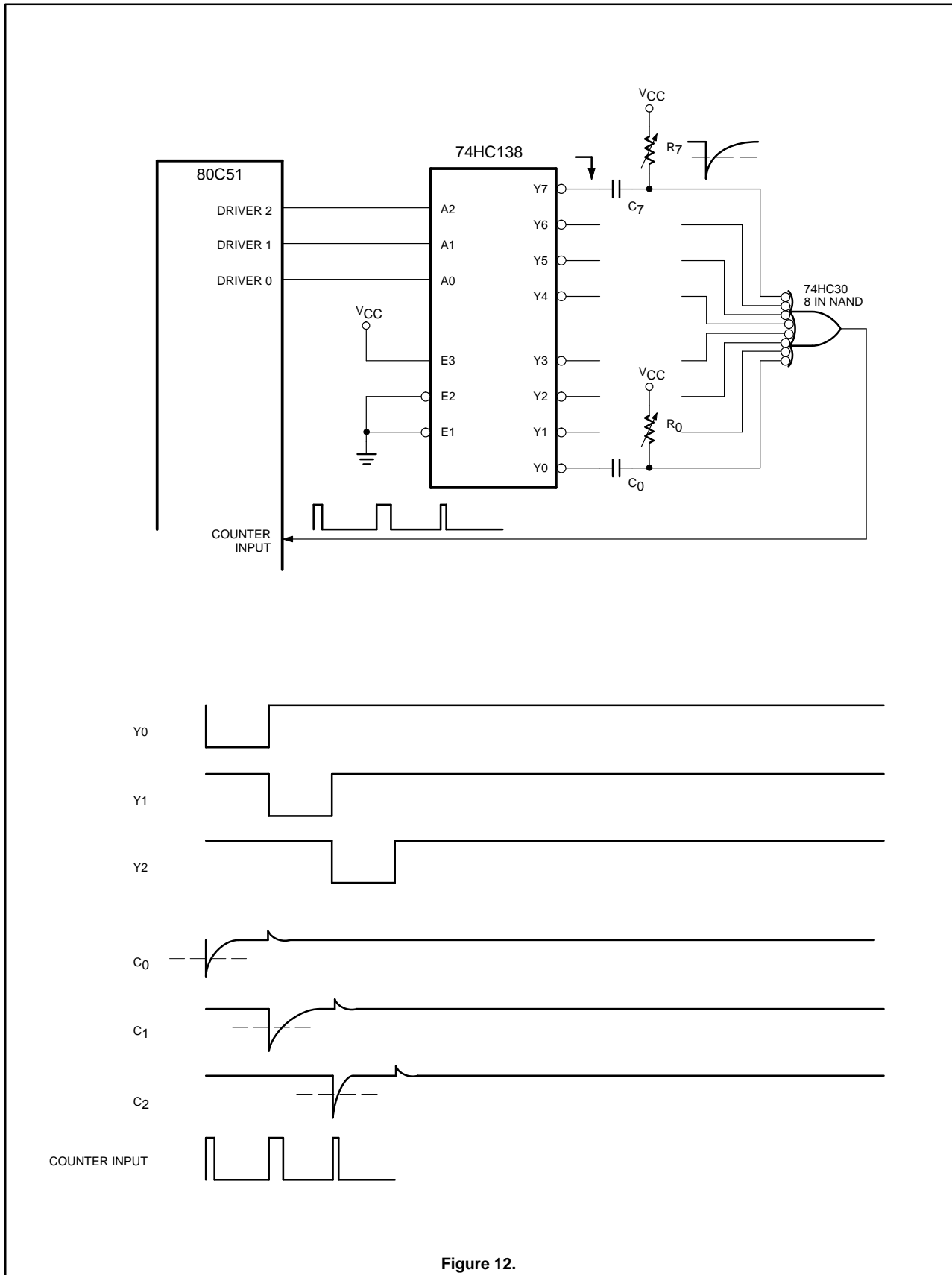
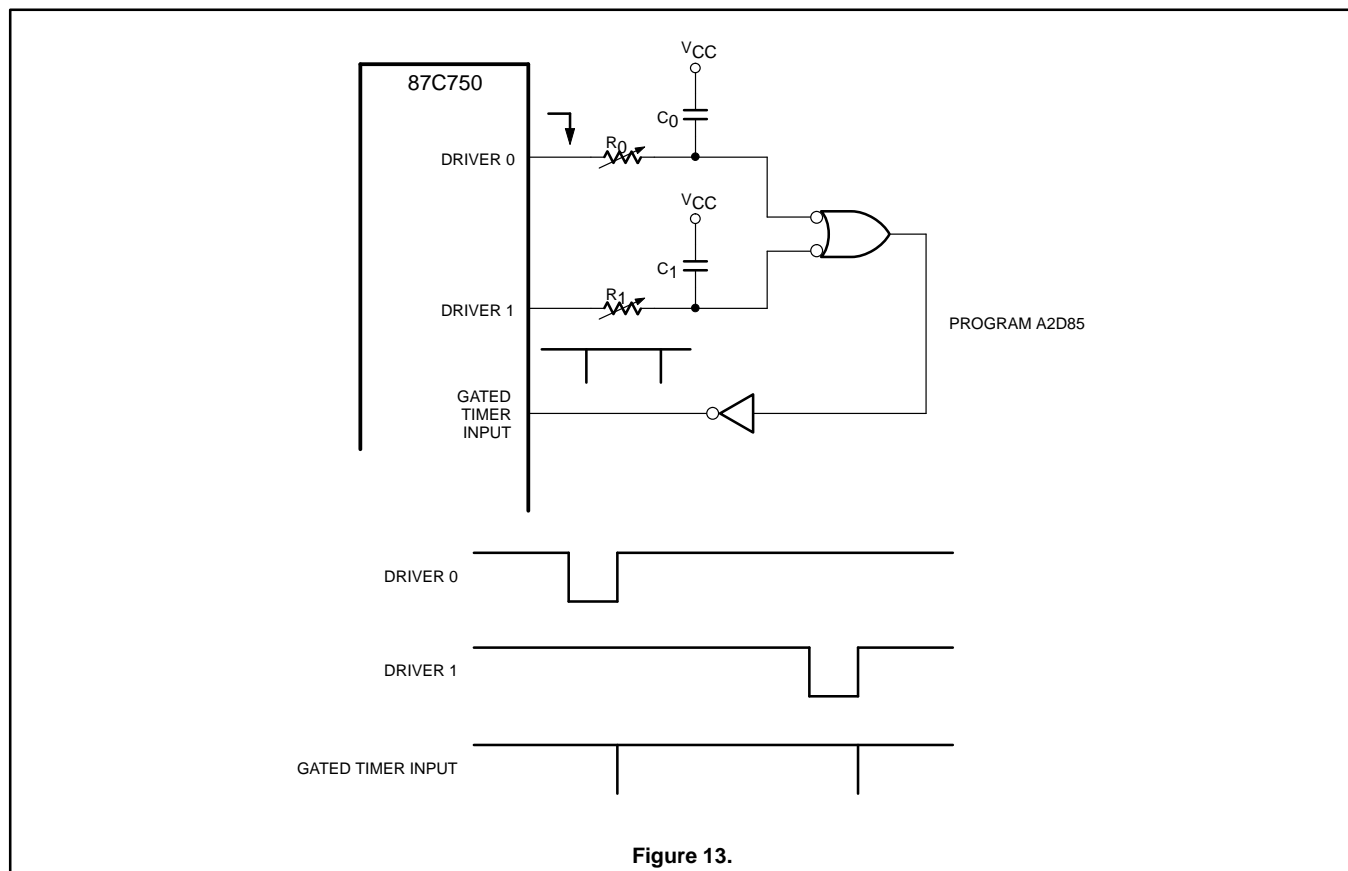


Figure 12.

# Measure resistance and capacitance without an A/D

AN449



# Measure resistance and capacitance without an A/D

AN449

## 80C51 ASSEMBLY LANGUAGE PROGRAMS

One of the programs has been listed below to show the section for the statistical data gathering. Note that the actual A to D code is quite short. The other programs which were used are available on the Philips Bulletin Board System, BBS. The BBS phone number and the data sheets for the 80C51 family products are available in the Philips IC20 "80C51-Based 8-Bit Microcontrollers" data book.

```

;***** A2D02 *****
$MOD752
;
;THIS PROGRAM IS TO TEST EXTERNAL COMPARATOR
;WITH INTERNAL TIMER FOR A/D CONVERSION.
;
LDELAY EQU    03FH    ;COUNTER FOR DELAY
HDELAY EQU    03EH    ;COUNTER FOR DELAY
DISPL EQU    03DH    ;LOW DISPLACEMENT
DISPH EQU    03CH    ;HIGH DISPLACEMENT
AVCNT EQU    03BH    ;AVERAGE COUNTER
SHFTCT EQU    03AH    ;16 BIT SHIFT CTR.
BOTTOM EQU    0FH    ;LO INTERMEDIATE ANS
LAVER EQU    0EH    ;MED INTERMEDIATE ANS
;LOW AVERAGE ANS
HAVER EQU    0DH    ;HI INTERMEDIATE AND
;HI AVERAGE ANS
A2DL EQU    3    ;TEMP TL
A2DH EQU    2    ;TEMP TH
LDEL EQU    0    ;DELAY OF 256 COUNTS
HDEL EQU    30    ;HIGH DELAY
SHIFTS EQU    3    ;NUMBER OF SHIFTS
;
    ORG    0000H
    AJMP   START
;
    ORG    0100H
;
START: MOV    R0,#3FH ;SET UP R0 FOR RAM CLR.
        SETB  P3.0    ;TURN OFF LED
        CLR   A        ;A = 0
ZERO:  MOV    @R0,A    ;RAM SET TO 0
        DJNZ  R0,ZERO
;
;   RAM IS NOW CLEARED
;
;
        MOV    IE,#0    ;DISABLE ALL INTERRUPTS
        MOV    AVCNT,#0 ;CLEAR AVERAGE COUNTER
;THIS PART DOES 256 DOUBLE PRECISION ADDS
;AND THEN ONLY THE TOP TWO BYTES ARE USED
;AS THE AVERAGE. THIS ACTS LIKE DIVIDING BY 256
;
GETAVR: ACALL  ATOD     ;DO AN A/D CONVERSION
        MOV    A,A2DL   ;LOW A/D TO A
        ADD   A,BOTTOM
        MOV    BOTTOM,A ;SAVE LOWEST IN BOTTOM
        MOV    A,A2DH   ;HIGH A/D TO A
        ADDC  A,LAVER
        MOV    LAVER,A
        CLR   A
        ADDC  A,HAVER
        MOV    HAVER,A ;HIGH AVERAGE
        DJNZ  AVCNT,GETAVR ;LOOP
;

```

## Measure resistance and capacitance without an A/D

AN449

```

;NOW DO THE DEVIATION PART
;
      MOV      AVCNT,#0FFH ;CLEAR AVCNT
DEVI: ACALL   ATOD      ;DO A/D CONVERSION
      MOV      A,A2DL
      CLR      C
      SUBB    A,LAVER
      MOV      DISPL,A
      MOV      A,A2DH
      SUBB    A,HAVER
      MOV      DISPH,A ;TH,TL - AVERAGE
                          ;TO DISP(H,L)
;
;DOUBLE PRECISION DIFFERENCE IN DISP(H,L)
;NOW ADD HEX 10 AND SEE IF DISP(H,L) IS NEG.
      ACALL   ADDTEN    ;ADD HEX 10 TO DISP
      MOV      A,DISPH ;CHECK IF NEG
      JNB     ACC.7,CKOVER ;MSB IS 0 SO POSITIVE
      MOV      DISPL,#0 ;NEG OVERFLOW,MAKE 0
      MOV      DISPH,#0
      SJMP    OK        ;DISP IS NOW IN RANGE
;
;NOW CHECK IF DISP(H,L) IS LARGER THAN 1F HEX
;
;      A=DISPH FROM ABOVE
;
CKOVER: JNZ     MAKEHI   ;UPPER BYTE GT 0
      MOV      A,DISPL
      ANL     A,#0E0H ;LOOK AT TOP 3 BITS
      JZ      OK        ;JUMP IF DISP IN RANGE
MAKEHI: MOV      DISPL,#1FH ;MAX ALLOWABLE DISP
OK:     ACALL   ADDTEN    ;NOW ADD TEN HEX
      MOV      R0,DISPL ;DISPLACEMENT IN R0
;
;R0 NOW HAS A DISPLACEMENT BETWEEN 10H AND 2FH
;
      INC     @R0      ;ADD 1 TO RAM SELECTED
      DJNZ    AVCNT,DEVI ;DO IT AGAIN, SAM
;
      CLR     P3.0     ;TURN ON LED
TONE:   MOV     HDELAY,#1H
SOUND:  DJNZ    LDELAY,$
      DJNZ    HDELAY,SOUND
      CPL     P3.5
      SJMP    TONE ;DONE, MAKE TONE
;*****THIS SUBROUTINE ADDS 10 HEX TO DISP
ADDTEN: MOV      A,#10H ;ADD HEX 10 TO MAKE +
      ADD     A,DISPL
      MOV     DISPL,A
      CLR     A
      ADDC    A,DISPH
      MOV     DISPH,A
      RET
;

```

## Measure resistance and capacitance without an A/D

AN449

```

;***** THIS IS THE A/D CONVERSION SUBROUTINE
;
ATOD:  SETB    P1.5    ;P1.5 IS INPUT
        SETB    P3.6    ;SHORT CAP TO GND
        MOV     TH,#0    ;16BIT TIMER =0
        MOV     TL,#0
        MOV     HDELAY,#HDEL ;SET UP HI DELAY TIME
SETTL:  MOV     LDELAY,#LDEL ;SET UP LO DELAY TIME
        DJNZ    LDELAY,$ ;LO DELAY LOOP
        DJNZ    HDELAY,SETTL ;UPPER DELAY LOOP
        MOV     TCON,#94H ;GATE, TR,IT0 ENABLE
        CLR     P3.6    ;START INTEGRATION
WAIT:   JNB     P1.5,GOOD ;WAIT FOR 0 ON P1.5
        JNB     TF,WAIT ;CHECK FOR OVERFLOW
        MOV     TCON,#0 ;OVERFLOW, STOP COUNTER
        MOV     TH,#0FFH ;MAKE TH = FULL SCALE
        MOV     TL,#0FFH ;MAKE TL = FULL SCALE
GOOD:   MOV     A2DL,TL ;SAVE A TO D LOW
        MOV     A2DH,TH ;SAVE A TO D HIGH
        SETB    P3.6    ;START DISCHARGING CAP
        MOV     SHFTCT,#SHIFTS ;SHIFT RIGHT
;
;THIS PART WILL DO A 16 BIT SHIFT RIGHT, SHFTCT TIMES.
;THIS IS DONE TO REDUCE THE PRECISION OF THE ANSWER
;BECAUSE THE SPREAD IN THE 255 READINGS IS TOO GREAT.
;
SHIFT:  MOV     A,SHFTCT
        JZ      ADDONE ;JUMP IF NO SHIFTS
        CLR     C      ;16 BIT SHIFT
        MOV     A,A2DH
        RRC     A
        MOV     A2DH,A
        MOV     A,A2DL
        RRC     A
        MOV     A2DL,A
        DJNZ    SHFTCT,SHIFT
ADDONE: RET
;
        END

```