LOW-COST LCD FREQUENCY METER

■ K.S. SANKAR

Frequency meters have always been expensive tools for the average hobbyists. Now, with microcontrollers and liquid-crystal displays (LCDs) having become very economical and popular, it is possible to build a compact and low-cost LCD-based frequency meter that can measure up to 250 kHz.

A sample photo of the LCD module is shown in **Fig. 1**. These modules are available in 14- and 16-pin configurations. The 16-pin module has a backlight option. Popular brands are Lampex and Hantronix. Note the pin numbers before soldering to the circuit.

In this project, the LCD module used is Lampex LM16200 with 16 alphanumeric characters and two lines with backlight option. Pin details of this module are given in Table I. A functional diagram of the module is shown in **Fig. 2**.

However, you may use any branded or unbranded 2-line, 16-character LCD module for this project. The 10k potentiometer, which controls the contrast of the LCD module, works best when its wiper contact is nearer to ground potential.

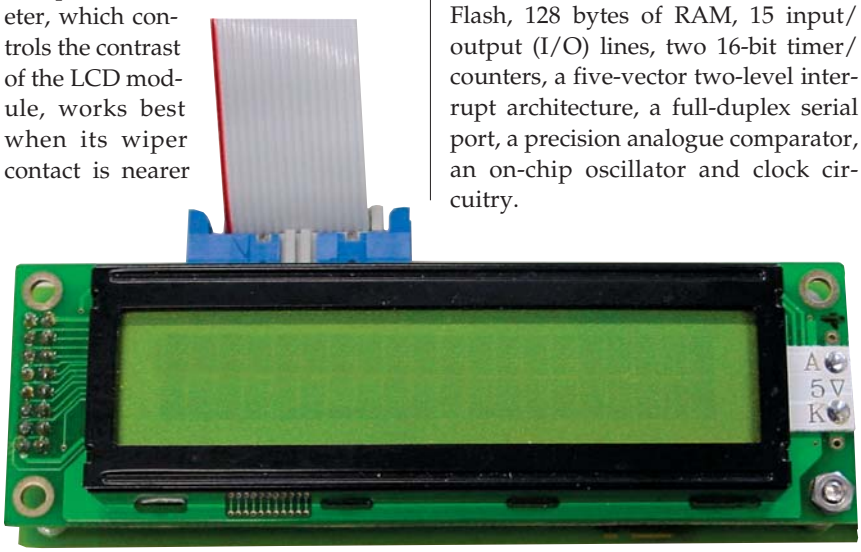

Fig. 1: 2×16 LCD module

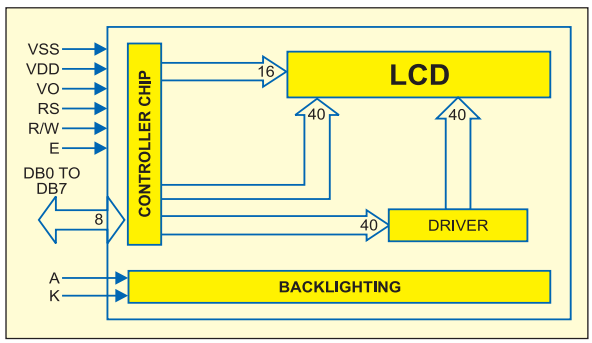

Fig. 2: Functonal diagram of LCD module

TABLE I
Pin Details of LM16200

Pin No.	Description	Pin no.	Description
1	Gnd	5	R/$\overline{W}$
2	Vcc	6	E
3	Vo	7 to 14	DB0 to DB7
4	RS	15 & 16	LED BL A & K

Circuit description

Fig. 3 shows the circuit of the frequency counter including the power supply. The microcontroller used is AT89C2051, which features 2 kB of Flash, 128 bytes of RAM, 15 input/output (I/O) lines, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full-duplex serial port, a precision analogue comparator, an on-chip oscillator and clock circuitry.

Port-1 is used to drive the LCD in 4-bit mode with 10-kilo-ohm pull-up resistors. The 24MHz crystal used gives a processing speed of 2 mega-instructions per second (MIPS).

Timer 0 is used as an external counter to count the input pulses. Transistor T1 amplifies the input signal, while non-inverting gate NI (1/6 CD4050) serves as a buffer for coupling the amplified pulses to input pin 8 (P2.4) of timer-0.

A software gate of one-second duration is used to count the number of pulses corresponding to the frequency of the input signal source. The count value is read and displayed on the 2-line, 16-character LCD module. The flow-chart of the frequency counting routine is shown in **Fig. 4.**

A conventional power supply circuit comprising a step-down trans-

PARTS LIST

Semiconductors:
IC1 - 7805, 5V regulator
IC2 - AT89C2051 microcontroller
IC3 - CD4050, hex buffer
T1 - BC547, npn transistor
D1-D4 - 1N4007, rectifier diode
D5 - 1N4148, switching diode
LED1 - 5mm LED

Resistors (all ¼-watt, ±5% carbon):
R1 - 1-kilo-ohm
R2, R4, R6-R9 - 10-kilo-ohm
R3 - 4.7-kilo-ohm
R5 - 150-ohm

Capacitors:
C1 - 1000µF, 25V electrolytic
C2, C3, C7 - 0.1µF ceramic disk
C4, C5 - 22pF ceramic disk
C6 - 10µF, 16V electrolytic

Miscellaneous:
X1 - 230V AC primary to 9V, 250mA secondary transformer
S1 - On/Off switch
S2 - Push-to-on switch
X_{TAL} - 24 MHz
- 16×2 LCD

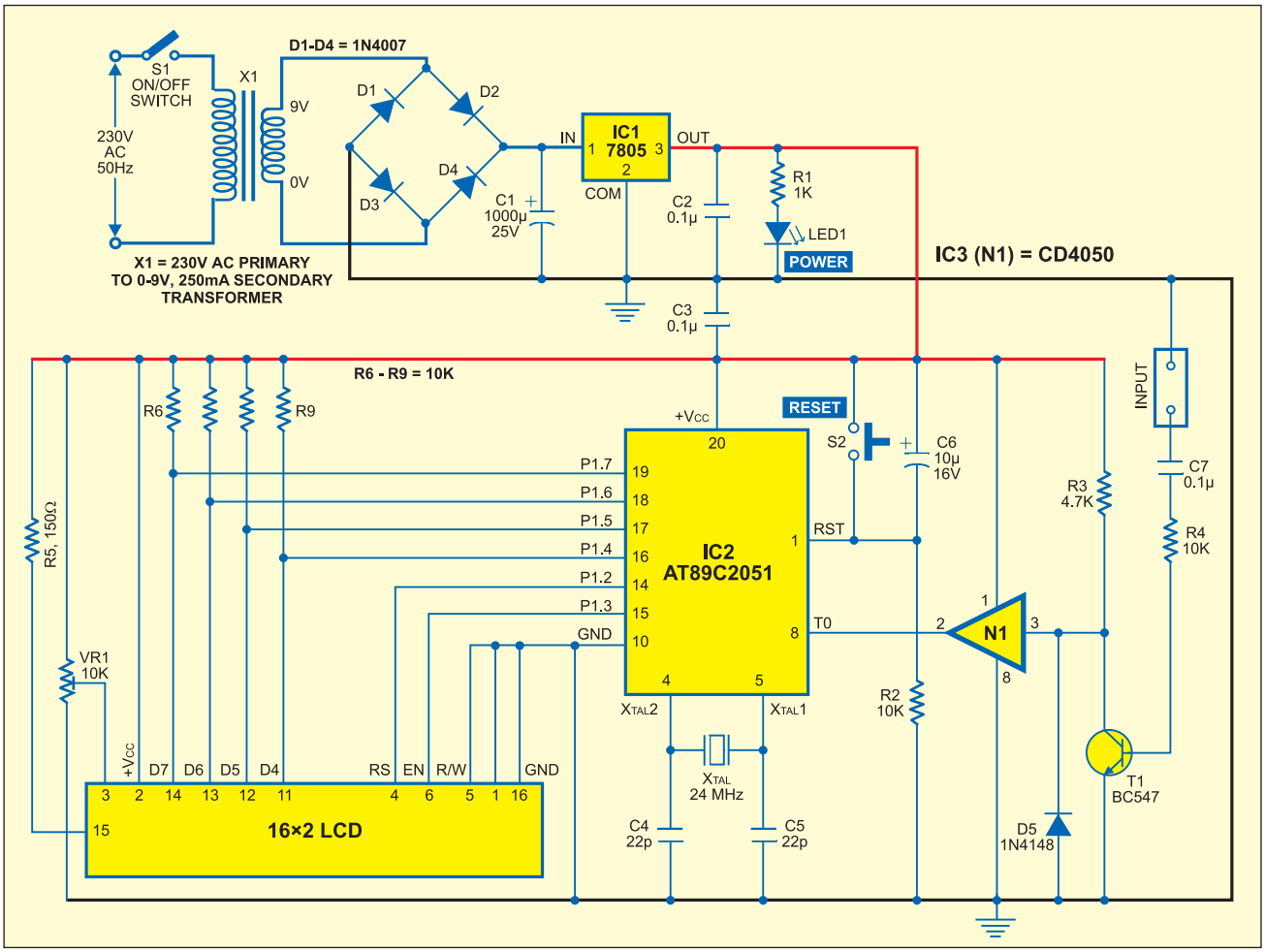

Fig. 3: Circuit diagram of frequency meter

TABLE II
LCD Connections Used for 4-Bit Data Mode

LCD display	Port	Pin
DB7	P1.7	14
DB6	P1.6	13
DB5	P1.5	12
DB4	P1.4	11
E	P1.3	6
RS	P1.2	4
RW	Ground	5
Vss	Ground	1
Vdd	+5 volt	2
Vo	0-5 volt	3

former followed by a bridge rectifier, smoothing capacitor and 5V regulator is used to power the circuit. Capacitor C2 (0.1µF) filters ripples in the output of the regulator and LED1 shows the supply status. To test the circuit, connect any pulse generator output to the probe and check the frequency displayed on the LCD screen.

The LCD module is used in the 4-bit data interface mode, wherein only data pins for DB4 through DB7 are used for data transfer. The configuration used is shown in **Table II**.

An actual-size, single-side PCB for the LCD frequency meter (**Fig. 3**) is shown in **Fig. 5** and its component layout in **Fig. 6.**

The software

The software is compiled using the demo version of BASCOM-8051, which can be downloaded from website 'www.mcselec.com.'

Syntax of some of the important instructions used in the program is shown in the box along with examples. The BASCOM compiler provides special instructions for use and display of data on the LCD module.

Syntax of Important Instructions Used in the Program with Examples

1. CONFIG LCDPIN. This instruction stores the pin usage in your program:
Syntax:
CONFIG LCDPIN = PIN,
Example:
CONFIG LCDPIN = PIN, DB4= P1.1, DB5=P1.2, DB6=P1.3, DB7=P1.4, E=P1.5, RS=P1.6
Note. LCD-module pin names are as used in **Table II**.
2. CONFIG LCD. This instruction is used for configuring the LCD display type:
Syntax:
CONFIG LCD = LCD type
LCD type can be one of the following:
40x4, 40x2, 16x1, 16x2, 16x4, 20x2 or 20x4 or 16x1a or 40x4a.
Note. Default 16x2 is assumed. The 16x1a LCD display is a special one. It is intended for the display that has the memory organised as two lines of eight characters. The 40x4a LCD display is also a special one. It has two ENABLE lines.
Example:
CONFIG LCD = 40x4
LCD "Hello" instruction is used for displaying 'Hello' on the LCD screen.
FOURTHLINE instruction selects line No. 4 of the screen for subsequent instruction.
LCD "4" displays '4' on the screen
END
3. CONFIG TIMER0, TIMER1. This instruction is used for configuring timer-0 or timer-1.
Syntax:
CONFIG TIMERx = COUNTER/TIMER,

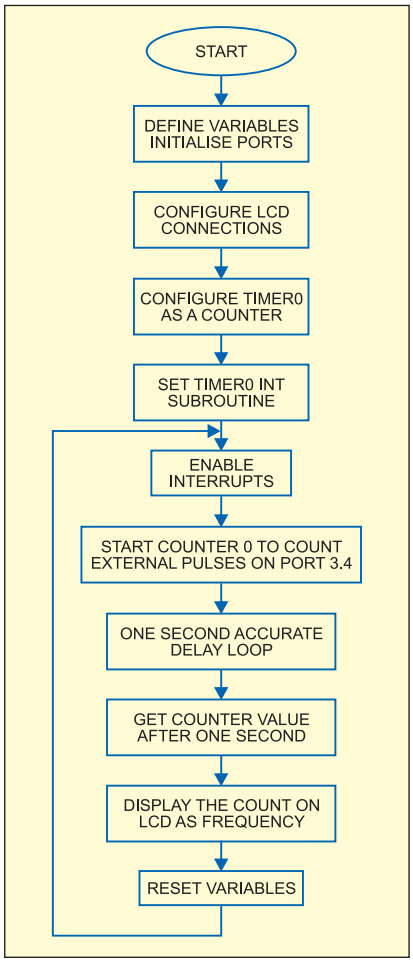

Fig. 4: Flowchart

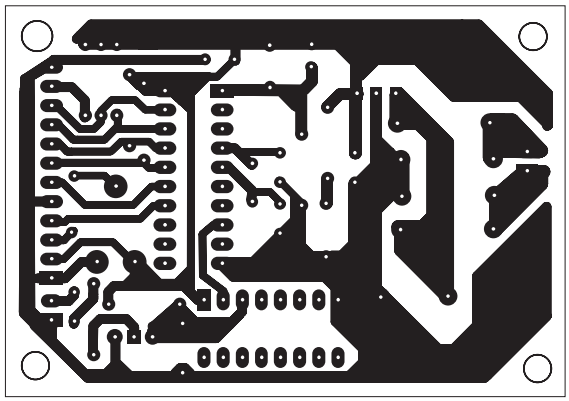

Fig. 5: Actual-size, single-side PCB layout for frequency meter

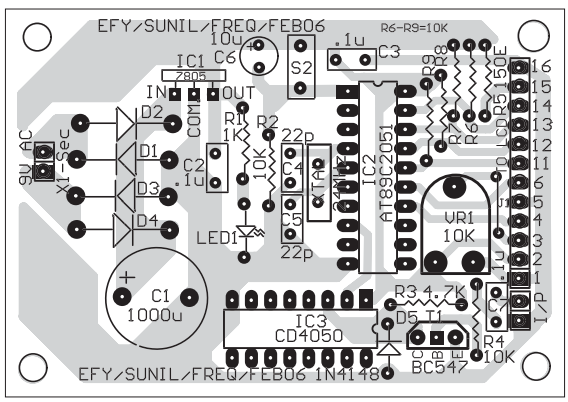

Fig. 6: Component layout for the PCB

```
GATE=INTERNAL/EXTERNAL, MODE=0/3
Remarks:
TIMERx     TIMER0 or TIMER1.COUNTER will con
figure TIMERx as a COUNTER and TIMER
willconfigure TIMERx as a
TIMER.A TIMER has
built-in clock input and a
COUNTER has external
clockinput.
GATE       —       INTERNAL or
EXTERNAL. Specify
EXTERNAL to enable gate
controlwith the INT
input.
MODE       —       Specify timer/
counter mode 0-3. See the
datasheet for more details.

Example:
CONFIG TIMER0=COUNTER,
MODE=1,
GATE=INTERNAL
COUNTER0 = 0 instruction resets
counter 0
START COUNTER0 enables the
counter to start counting
DELAY 'wait a while
PRINT COUNTER0 instruction
prints the counter0
count.
END
```

For use of BASCOM, you may refer to the article 'Real-Time Clock' published in Jan. 2005 issue of EFY.

The source code file EFY80FM24.BAS for this LCD frequency meter in BASCOM-51 is given at the end of this article. The same may be modified to meet your specific requirements. The relevant files are included in the EFY-CD.

EFY20FM24

```
'-----------------------------------------------------------------
' file: efy80fm24.BAS  3-12-05
' ok with word variable 45440
' Frequency Meter Program using AT89c2051 micro controller
' written using bascom-51
' from www.mcselec.com holland
' an embedded visual basic compiler for 8051 microcontrollers
' by K.S.Sankar Web: www.mostek.biz
'-----------------------------------------------------------------
' Connect the timer0 input P3.4 to a frequency generator
' freq meter
' 24  mhz xtal ok upto 300khz

' define crystal speed and include file
$regfile = "89C2051.dat"
$crystal = 24000000

' define variables used
Dim A As Byte
Dim C As Long , D As Long
Dim Count As Word
Dim Onceasec As Bit
Dim T0ic As Long
Dim Green As Byte
Dim Delayword As Word

' Initialize variables
Onceasec = 0
Count = 0
T0ic = 0
D = 0
Green = 0

' initialize ports
P1 = 0
P3 = 255

' configure lcd display
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = P1.4 , Db5 = P1.5 , Db6 = P1.6 , Db7 = P1.7 , E = P1.3 , Rs = P1.2
Cls
'clear the LCD display
Lcd "Frequency Meter"

' define timer0
Config Timer0 = Counter , Gate = Internal , Mode = 1
'Timer0 = counter : timer0 operates as a counter
'Gate = Internal  : no external gate control
' exte/internal makes no difference
'Mode = 1        : 16-bit counter
' set t0 internal interrupt

On Timer0 Timer_0_overflow_int
' interrupt will be generated on every 65536 count
Priority Set Timer0
Enable Interrupts
Enable Timer0

Counter0 = 0
'clear counter
Start Counter0
'enable the counter to count

Do
'set up a 1 sec accurate DO NOTHING loop
Enable Interrupts
'wait 1 as per BASCOM-51 is not accurate
For Delayword = 1 To 45440
Next Delayword

Disable Interrupts
C = Counter0
'get counter value
D = T0ic * 65536

Lowerline
C = C + D
T0ic = 0
Lcd "                "
Lowerline
' show the frequency
Lcd "f=" ; C ; " Hz"
Waitms 255
Waitms 255

C = 0

Counter0 = 0
Start Counter0
're-start it because it was stopped by accessing the COUNTER Loop

' timer0 int subroutine
Timer_0_overflow_int:
Rem timer0 overflow ( 65535 ) interrupt comes here
' increment the variable
Incr T0ic
Return
End

' end of program
' uses 1114 bytes of program memory
```