CHAPTER 3

SOFTWARE DEVELOPMENT FOR PC BASED INSTRUMENT FOR
THE MEASUREMENTS OF BASIC ELECTRICAL PARAMETERS
CHAPTER 3
SOFTWARE DEVELOPMENT FOR PC BASED INSTRUMENTS
FOR THE MEASUREMENT OF BASIC ELECTRICAL PARAMETERS

3.1 INTRODUCTION:

With the advent of PC, it has now become quite easy to add intelligence and computer power to any instrument, machine, process or system. The levels of intelligence and computational power that can be added to a PC based system depend on the memory capacity, software support and the interfacing arrangements actually provided. The total activities of any PC based on-line real-time system can be divided into two separate systems, viz. (i) the activities taking place in the PC world and (ii) that in the real world. Interchange between these two worlds of activities is made possible by employing suitable interfacing arrangements.

The designer of a PC based system will have to solve two problems: (1) Interfacing problems associated with interfacing hardware and (2) the software problem associated with the application requirements. It is also very essential to develop user-friendly software for easy application. The development of interactive software simplifies the operator's job very much and also enables error-free operation of the system.

3.2 SYSTEM SOFTWARE DIVISION:
The system software is developed in VISUAL BASIC and divided into six parts viz.,

i. program for displaying the menu and selecting option
ii. program for voltage measurement
iii. program for current measurement
iv. program for resistance measurement
v. program for frequency measurement and
vi. Program for phase measurement and displaying the parameters measured.

Fig 3.1 shows the generalized flowchart for the computer based measurement system. It describes briefly the working of the system. The program first initializes all peripheral devices like 8255, 8254 and A/D converter. The software enters into the menu program, then it display menu as shown below. The appropriate function can be carried out by exercising the following options.

Select the option by entering the appropriate number
1. Measurement of voltage
2. Measurement of current
3. Measurement of resistance
4. Measurement of frequency
5. Measurement of phase

The program after displaying the menu, poles the key board continuously until any one of the above mentioned keys is pressed. On selecting anyone of the options by pressing the appropriate key, the computer enters into the corresponding program and executes the corresponding process.

When key '1' is pressed, it enters into the process for voltage measurement. In this process, it measures the voltage of ac/dc signal applied at the input. After reading finally it displays the measured value. When key '2' is pressed it enters into the current measurement process, there it measures the current through parallel port and displays the same on the monitor. When key '3' is pressed, it enters into the process of resistance measurement, there it
measures unknown resistance and displays the same on the monitor. When key '4' is pressed, it enters into the process of frequency measurement, there it measures signal frequency and displays the same on the monitor. When key '5' is pressed, it enters into the process of phase measurement, there it measures phase between the signals and displays the same on the monitor. When key '6' is pressed it comes out of the program. The remaining flowcharts describe in detail the working of all subroutines used in the software.

3.3 VOLTAGE MEASUREMENT:
The system software for the measurement of voltage first initializes programmable peripheral interface (PPI) 8255. The detail of the software is based on the flow chart shown in figure 3.2. The PPI 8255 is initialized in mode-0 (simple I/O) and port-A as input, port-B as input, port-C (Upper) as output and port-C (Lower) as input port. After initialization of PPI 8255, the program issues SOC signal, checks EOC signal, enables output latches and reads digital data through port-A and port-B. This two’s complimented binary data is employed to compute the analog voltage. The detailed listing of the program is presented in section 3.9.

3.4 CURRENT MEASUREMENT:
It is exactly same as that of voltage measurement scheme with a change of multiplication factor. The current is gradually controlled from zero to maximum of 1 A and the corresponding digital signal is measured through the PC. Thus for every possible digital signal, the analog current flowing through the actual electrical circuit or system is determined to obtain the calibration curve of the analog current against the digital input to the PC. These operations are presented in form of flow chart shown in figure 3.3. The software is so developed that the PC, after initialization, receives the instantaneous signal, multiplies the result with a proper multiplication factor, and displays the measured analog current in
decimal form on the display screen. The detailed program listing is presented in section 3.10.

3.5 RESISTANCE MEASUREMENT:
A constant current of 100μA is allowed to flow through unknown resistor. The voltage across the unknown resistor is applied to an A/D converter and the corresponding digital signal is measured through the PC. Thus for every possible digital signal, the analog voltage across the actual electrical circuit or system is determined to obtain the calibration curve of the analog current against the digital input to the PC. These operations are presented in form of flow chart shown in figure 3.4. The software is so developed that the PC, after initialization, receives the instantaneous signal, multiplies the result with a proper multiplication factor, and displays the measured unknown resistor in decimal form on the display screen. The detailed program listing is presented in section 3.11.

3.6 FREQUENCY MEASUREMENT:
The fig.3.5. Shows the flow chart for frequency measurement. The program first initiates all peripheral devices such as programmable peripheral interface (PPI) 8255 and programmable interval timer (PIT) 8254. The PPI 8255 is initialized in mode-0 (simple I/O) and port-A as output port. The PIT 8254 is programmed in mode-0 (Interrupt on terminal count). It is programmed in this mode to count number of clock pulses in one second. After initialization of all peripheral devices counter0 is loaded with data FFFF H. The PPI 8255 enables the gate for 1 sec. counter0 count is decremented corresponding to the signal applied at clock0. Finally, software estimates frequency by subtracting count from FFFF H. The detailed program listing is presented in section 3.12.
3.7 PHASE MEASUREMENT:

The fig 3.6. Shows the flow chart for the phase meter. The first step in software initializes PPI and PIT and loads the counter with FFFF H. The second step selects the phase difference signal by sending HIGH to the corresponding control input of the multiplexer. After reading count proportional with ‘t’ in the counter, software selects time period signal through multiplexer, then it reads the count proportional to the 'tt' in the counter. The final step in the software estimates the actual count by subtracting the final counts for 't' and 'tt' from initial count. The process will be repeated continuously. The detailed program listing is presented in section 3.13.
FIG 3.1 FLOW CHART OF PC BASED VIRTUAL MULTIMETER

Start

Initialize all peripherals

Select the option
1. Measurement of voltage
2. Measurement of current
3. Measurement of resistance
4. Measurement of frequency
5. Measurement of phase
6. Exit

1
Voltage V

2
Current C

3
Resistance R

4
Frequency F

5
Phase P

6
Exit
3.8 VISUAL BASIC PROGRAM FOR VIRTUAL MULTIMETER MEASUREMENTS

Public Sub delay (howlong As Date)
    temptime = DateAdd("s", howlong, Now)
    While temptime > Now
        DoEvents
        Wend
    End Sub

Private Sub Form_Load ()

    'Declaration of variables
    Dim pa, pb, pc, cr, c0, c1, ccr, eoc, num, cnst, option As Integer
    Dim Isb As Byte
    Dim msb As Byte
    Dim ain As Single
    Dim w As Byte
    Dim x As Byte
    Dim y As Byte
    Dim z As Long

    'Initialization of 8255 & 8254 port addresses
    pa = &HB800
    pb = &HB801
pc = &HB802
cr = &HB803
c0 = &HB808
c1 = &HB809
ccr = &HB80B

'Initialization of constants
cnst = &H100
w = &H0

'Control word for 8255
out cr, &H93

Control word for 8254
out ccr, &H30
End Sub

Private Sub Text1_LostFocus()
If text1.text = val(1) then
MsgBox(text1.text)
MsgBox(voltage())
Elseif text1.text = val(2) then
MsgBox(text1.text)
MsgBox(Current())
Elseif text1.text = val(3) then
MsgBox(text1.text)
MsgBox(Resistance())
End Sub
Elseif text1.text=val(4) then
Msgbox(text1.text)
Msgbox( Frequency())
Elseif text1.text=val(5) then
Msgbox(text1.text)
Msgbox( Phase())
Elseif text1.text=val(6) then
End
Endif
Loop Until (w<>0)

End Sub
FIG 3.2 FLOW CHART OF PC BASED VOLTAGE MEASUREMENT

1. Issue SOC signal on PC4 bit
2. Read EOC signal on PC0 bit
3. Enable output latches by sending HIGH on PC7 bit
4. Read two's compliment binary output and convert it to decimal
5. Display results
6. RTN
3.9 VISUAL BASIC PROGRAM FOR MEASUREMENT OF VOLTAGE

Private Function Voltage ()
'Issuing SOC
Out pc, &H10
Out pc, &H0

'Read EOC until it becomes HIGH
Do
  eoc = inp (pc)
Loop Until ((eoc And &H1) = &H0)

'Updating output registers
Out pc, &H0
Out pc, &H10

'Read digital data
lsb = inp (pa)
msb = inp (pb)

'Conversion of binary two's compliment binary output to decimal
If (msb = &H80 And lsb = &H0) Then
  num = &H8000
ElseIf ((msb And &H80) = &H80) Then
  msb = Not msb
  lsb = Not lsb
  num = ((msb * &H100) Or lsb) + &H1
Else
  num = msb * &H100 + lsb
End If
Num = (Not num) + 1
Else
num = (msb * &H100) Or Isb
End If

'Calculation of decimal equivalent number to applied voltage
ain = num * 0.000305 * 2 * 1.414

'Display of measured voltage
Text2. Text = Val(ain) & "V"

End Function
FIG 3.3 FLOW CHART OF PC BASED CURRENT MEASUREMENT

1

Issue SOC signal on PC4 bit

Read EOC signal on PC0 bit

EOC = 1

Enable output latches by sending HIGH on PC7 bit

Read two's compliment binary output and convert it to its equivalent current measured

Display results

RTN
Private Function current ()
Dim r as integer
R=1
'Issuing SOC
Out pc, &H10
Out pc, &H0

'Read EOC until it becomes HIGH
Do
  eoc = inp(pc)
Loop Until ((eoc And &H1) = &H0)
'Updating output registers
Out pc, &H0
Out pc, &H10

'Read digital data
lsb = inp(pa)
msb = inp(pb)

'Conversion of binary two's compliment binary output to decimal
If (msb = &H80 And lsb = &H0) Then
  num = &H8000
ElseIf ((msb And &H80) = &H80) Then
  msb = Not msb
  lsb = Not lsb

num = ((msb * &H100) Or lsb) + &H1
num = (Not num) + 1
Else
num = (msb * &H100) or lsb
End If

'Calculation of decimal equivalent number to applied current
ain = (num * 0.000305 * 2 * 1.414)/(2*r)

'Display of measured current
Text2. Text = Val(ain) & "A"

End Function
FIG S.4 FLOW CHART OF PC BASED RESISTANCE MEASUREMENT

- Issue SOC signal on PC4 bit
- Read EOC signal on PC0 bit
- EOC = 1
- Enable output latches by sending HIGH on PC7 bit
- Read two's compliment binary output and convert it to its equivalent resistance measured
- Display results
- RTN
3.11 VISUAL BASIC PROGRAM FOR MEASUREMENT OF RESISTANCE

Private Function Resistance ()
Dim I as integer
I = 0.0001
'Issuing SOC
Out pc, &H10
Out pc, &H0
'Read EOC until it becomes HIGH
Do
eoc = inp (pc)
Loop Until ((eoc And &H1) = &H0)
'Updating output registers
Out pc, &H0
Out pc, &H10

'Read digital data
lsb = inp (pa)
msb = inp (pb)

'Conversion of binary two's compliment binary output to decimal
If (msb = &H80 And lsb = &H0) Then
num = &H8000
ElseIf ((msb And &H80) = &H80) Then
msb = Not msb
Isb = Not Isb
num = ((msb * &H100) Or Isb) + &H1
num = (Not num) + 1
Else
num = (msb * &H100) Or Isb
End If

'Calculation of decimal equivalent number to unknown resistance
ain = (num * 0.000305 )/(i*2)

'Display of measured resistance
Text2. Text = Val(ain) & "ohms"
End function
Load counter with FFFFH
Start measurement by sending HIGH to PA0
Read counter and subtract its count from FFFFH
Display frequency
RTN

FIG 3.3 FLOW CHART OF PC BASED FREQUENCY MEASUREMENT
3.12 VISUAL BASIC PROGRAM FOR MEASUREMENT OF FREQUENCY

Private function frequency()
  'Control word for 8254
  out ccr, &H30
  'Initialization of counter0
  out c0, &HFF
  out c0, &HFF
  'Sending HIGH to GATE for one second
  out pa, &H1
delay 1
  out pa, &H0
  'Reading counter
  x = inp(c0)
y = inp(c0)
  'Calculation of frequency
  z = ((y * cnst) + x)
z = 65535 - z
  'Displaying frequency
  Text2.Text = Val(z) & "Hz"
End Function
FIG 3.6 FLOW CHART OF PC BASED PHASE MEASUREMENT

- P
- Load counter with FFFFH
- Select 'T' input of multiplexer
- Start measurement by sending HIGH to PA0
- Read output of sync circuit through PC0
- Is PC0 HIGH?
- Subtract count in the counter from FFFFH
- Repeat above five steps for signal 'T'
- Display results
- RET
3.13 VISUAL BASIC PROGRAM FOR MEASUREMENT OF PHASE

Public Function count() As Long
    out pb, &H0 + a
    out pa, &H0 + a
    delay 5
    out pa, &H1 + a

    out ccr, &H70
    out c1, &HFF
    out c1, &HFF

    Do
        x = inp(pc)
        y = x And &H1
        loopwhile (y = 0)
    Loop

    Do
        x = inp(pc)
        y = x And &H1
        loopwhile (y <> 0)
    Loop

    out pa, &H0 + a
    x1 = inp(c1)
    y1 = inp(c1)
    z1 = (y1 * 256) + x1
    z1 = 65535 - z1
    out pb, a + &H0
End Function
Private function phase()
out ccr, &HB6

c=1
Do
out cr, &H89

a = &H2
t = countQ

a = &H4
tt = count()

p = (t / tt) * 360
text2.Text = Val(p)

loopwhile (c = 1)

End Function
VIRTUAL MULTIMETER

1. VOLTAGE MEASUREMENT
2. CURRENT MEASUREMENT
3. RESISTANCE MEASUREMENT
4. FREQUENCY MEASUREMENT
5. PHASE MEASUREMENT
6. EXIT

ENTER CHOICE

MEASURED VALUE IS 3.08 VOLTS

FIG 3.7 VISUAL BASIC FORM FOR VIRTUAL MULTIMETER