

Appendix A

Projection Specification

**University of Southern Queensland
Faculty of Engineering and Surveying**

**ENG 4111/2 Research Project
Project Specification**

For: Faculty of Engineering and Surveying

Topic: A monitoring, analysis and reporting system for health data captured during exercise

Supervisor: Dr Selvan Pather

Background: Over the years many efforts have been contributed to aid the aged population, but improvements in this area seem not to benefit much. This group of people has to raise awareness of the health of oneself. The project objective is to develop a device to monitor health indicators.

The Cardiovascular system in our body is a good indicator of the health. The cardiovascular system consists of the heart, the blood and the blood vessels. The measurement of the heart rate and blood pressure will be indicators of well being of health.

Program: Issue: 1 (28 March 2005)

1. Define the objective and tasks of the project.
2. State the Requirement, Restriction and Resources of Project.
3. Research the critical parameters, measuring device type and current devices in the market.
4. Investigate the interface of each device. (The storage system of device and the interface to computer)
5. Create ideas several possible systems and selection of the most suitable system
6. Develop and test the interface and software between the device and the computer

As time permits:

7. Evaluate the Design and minor improvement to the system
8. Incorporate other devices to the system

Specification drafted by
Andy Law Boon Lee
D10349257

Dr Selvan Pather (Supervisor)

Appendix B

Recorded Readings

B1 1st Reading

1 st reading								
	Each 2 Duty							
pins	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	2	3	1	1	1	0	2	2
3	2	2	1	3	1	1	2	0
4	2	1	1	3	1	2	2	0
5	2	2	0	1	1	1	3	2
6	2	3	1	1	1	0	2	2
7	2	2	1	3	1	1	2	0
8	2	1	1	3	1	2	2	0
9	2	2	0	1	1	1	3	2
10	3	3	0	0	0	0	3	3
10	0	2	0	2	3	1	3	1
11	3	3	0	2	0	0	3	1
11	1	3	0	0	2	0	3	3
12	3	3	1	1	0	0	2	2
12	3	3	2	0	0	0	1	3
13	1	1	1	2	2	2	2	1
13	3	3	2	0	0	0	1	3
14	3	3	0	0	0	0	3	3
14	3	3	1	0	0	2	3	3
15	3	3	0	0	0	0	3	3
15	1	3	2	0	2	0	1	3
16	2	2	2	1	1	1	1	2
17	2	3	1	1	1	0	2	2
18	3	3	1	0	0	0	2	3
19	3	3	0	2	0	0	3	1
20	0	2	0	2	3	1	3	1
21	3	3	0	0	0	0	3	3
22	3	3	3	1	0	0	0	2
23	2	3	1	1	1	0	2	2
24	3	3	1	0	0	0	0	2
25	3	3	0	0	0	0	3	3
26	1	1	1	1	2	2	2	2
27	2	2	1	1	1	1	2	2
28	3	3	1	2	0	0	2	1
29	3	3	1	1	0	0	2	2
30	3	3	1	0	0	0	2	3
31	1	3	1	3	2	0	2	0
32	3	3	1	2	0	0	2	1
33	2	2	1	1	1	1	2	2
34	2	2	2	1	1	1	1	2

1st reading from the input of the LCD

B2 2nd Reading

2 nd reading									
Each 2 Duty									
Pins	1	2	3	4	5	6	7	8	
1	0	0	0	0	0	0	0	0	0
2	2	3	1	1	1	0	2	2	2
3	1	3	1	1	2	0	2	2	2
4	2	2	0	2	1	1	3	1	1
5	2	2	0	1	1	1	3	2	2
6	2	3	1	1	1		2	2	2
7	1	2	0	2	2	1	3	2	2
8	1	3	1	2	2	1	3	2	2
9	2	2	0	1	1	1	3	2	2
10	3	3	0	0	0	0	3	3	3
10	0	2	0	2	3	1	3	1	1
11	3	3	1	3	0	0	2	0	0
12	3	3	1	2	0	0	2	1	1
12	3	3	2	0	0	0	1	3	3
13	2	2	2	1	1	1	1	2	2
14	3	3	0	0	0	0	3	3	3
14	3	3	1	0	0	0	2	3	3
15	3	3	0	0	0	0	3	3	3
15	1	3	2	0	2	0	1	3	3
16	1	1	1	1	2	2	2	2	2
17	2	3	1	1	1	0	2	2	2
18	3	3	1	0	0	0	2	3	3
19	0	2	0	3	3	1	3	0	0
20	1	3	1	0	2	1	2	3	3
21	1	3	0	0	2	0	3	3	3
22	3	3	1	2	0	0	2	1	1
23	3	3	1	1	0	0	2	2	2
24	3	3	1	0	0	0	2	3	3
25	3	3	0	0	0	0	3	3	3
26	2	2	2	1	1	1	1	2	2
27	2	2	2	1	1	1	1	2	2
28	3	3	1	2	0	0	2	1	1
29	3	3	1	1	0	0	2	2	2
30	3	3	1	2	0	0	2	1	1
31	3	3	1	1	0	0	2	2	2
32	3	3	1	2	0	0	2	1	1
33	2	2	1	1	1	1	2	2	2
34	2	2	1	1	1	1	2	2	2

2nd reading from the input of the LCD

B3 3rd Reading

3 rd reading									
Each 2 Duty									
Pins	1	2	3	4	5	6	7	8	
1	0	0	0	0	0	0	0	0	0
2	2	3	1	1	1	0	2	2	
3	1	2		2	2	1	3	1	
4	2	1	1	3	1	2	2	0	
5	2	2	0	1	1	1	3	2	
6	2	3	1	1	1		2	2	
7	2	2	1	3	1	1	2	0	
8	2	2	0	2	1	1	3	1	
9	2	2	0	1	1	1	3	2	
10	1	3	0	0	2	0	3	3	
10	3	3	1	0	0	0	2	3	
11	3	3	2	0	0	0	1	3	
11	3	3	0	0	0	0	3	3	
12	3	3	1	1	0	0	2	2	
12	1	3	2	2	2	0	1	1	
13	3	3	2	0	0	0	1	3	
13	1	1	2	2	2	2	1	1	
14	3	3	1	0	0	0	2	3	
14	3	3	0	2	0	0	3	1	
15	3	3	0	0	0	0	3	3	
15	1	3	0	0	2	0	3	3	
16	1	1	2	2	2	2	1	1	
17	2	3	1	1	1	0	2	2	
18	3	3	1	2	0	0	2	1	
19	3	3	1	1	0	0	2	2	
20	1	3	0	0	2	0	3	3	
21	3	3	2	0	0	0	1	3	
22	2	3	1	3	1	0	2		
23	1	3	0	0	2	0	3	3	
24	3	3	0	0	0	0	3	3	
25	2	3	1	1	1	0	2	2	
26	1	1	1	2	2	2	2	1	
27	2	2	2	1	1	1	1	2	
28	0	2	0	2	3	1	3	1	
29	3	3	0	0	0	0	3	3	
30	3	3	1	0	0	0	2	3	
31	2	3	1	1	1	0	2	2	
32	0	2	1	3	3	1	2	0	
33	2	2	2	1	1	1	1	2	
34	2	2	2	1	1	1	1	2	

3rd reading from the input of the LCD

B4 4th Reading

4 th reading									
Each 2 Duty									
Pins	1	2	3	4	5	6	7	8	
1	0	0	0	0	0	0	0	0	0
2	2	3	1	1	1		2	2	
3	2	2	1	3	1	1	2	0	
4	0	2	1	1	3	1	2	2	
5	2	2	0	1	1	1	3	2	
6	2	3	1	1	1	0	2	2	
7	2	2	1	3	1	1	3	0	
8	2	2	0	2	1	1	3	1	
9	2	2	0	1	1	1	3	2	
10	2	2	2	1	1	1	1	2	
11	2	2	2	1	1	1	1	2	
12	2	2	2	1	1	1	1	2	
13	2	2	2	1	1	1	1	2	
14	1	1	1	1	2	2	2	2	
15	2	2	2	1	1	1	1	2	
16	2	2	2	1	1	1	1	2	
17	2	3	1	1	1	0	2	2	
18	3	3	1	0	0	0	2	3	
19	0	2	0	3	3	1	3	0	
20	1	3	1	0	2	0	2	3	
21	0	2	0	3	3	1	3	0	
22	2	3	1	3	1	0	2	0	
23	3	3	1	3	0	0	2	0	
24	3	3	1	0	0	0	2	3	
25	3	3	0	0	0	0	3	3	
26	3	3	0	0	0	0	3	3	
27	2	2	2	1	1	1	1	2	
28	3	3	1	0	0	0	0	2	
29	3	3	0	2	0	0	3	1	
30	3	3	1	2	0	0	2	1	
31	3	3	1	1	0	0	2	2	
32	1	2	0	0	2	1	3	3	
33	2	3	1	1	1	0	2	2	
34	2	2	2	1	1	1	1	2	

4th reading from the input of the LCD

B5 The display on the LCD of the readings

1 st reading	
SYS	134
DIA	87
PULSE	60
Date	8/15
Time	3:20
PM/AM	PM
Memory Symbol	Yes
Average Symbol	No
Movement Symbol	No
Battery Low Symbol	No
Irregular Heart Beat Symbol	No
Deflation Symbol	No

2 nd reading	
SYS	144
DIA	84
PULSE	59
Date	8/15
Time	3:19
PM/AM	PM
Memory Symbol	Yes
Average Symbol	No
Movement Symbol	No
Battery Low Symbol	No
Irregular Heart Beat Symbol	No
Deflation Symbol	No

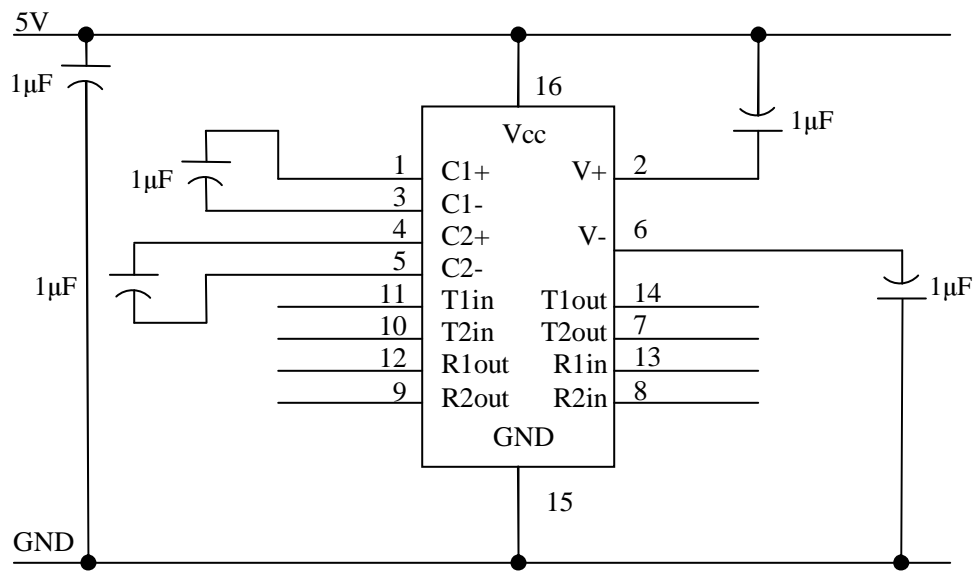
3 rd reading	
SYS	176
DIA	75
PULSE	124
Date	8/8
Time	5:22
PM/AM	PM
Memory Symbol	Yes
Average Symbol	No
Movement Symbol	No
Battery Low Symbol	No
Irregular Heart Beat Symbol	Yes
Deflation Symbol	No

4 th reading	
SYS	140
DIA	85
PULSE	59
Date	
Time	
PM/AM	
Memory Symbol	Yes
Average Symbol	Yes
Movement Symbol	No
Battery Low Symbol	No
Irregular Heart Beat Symbol	No
Deflation Symbol	No

Display on LCD of the 4 readings

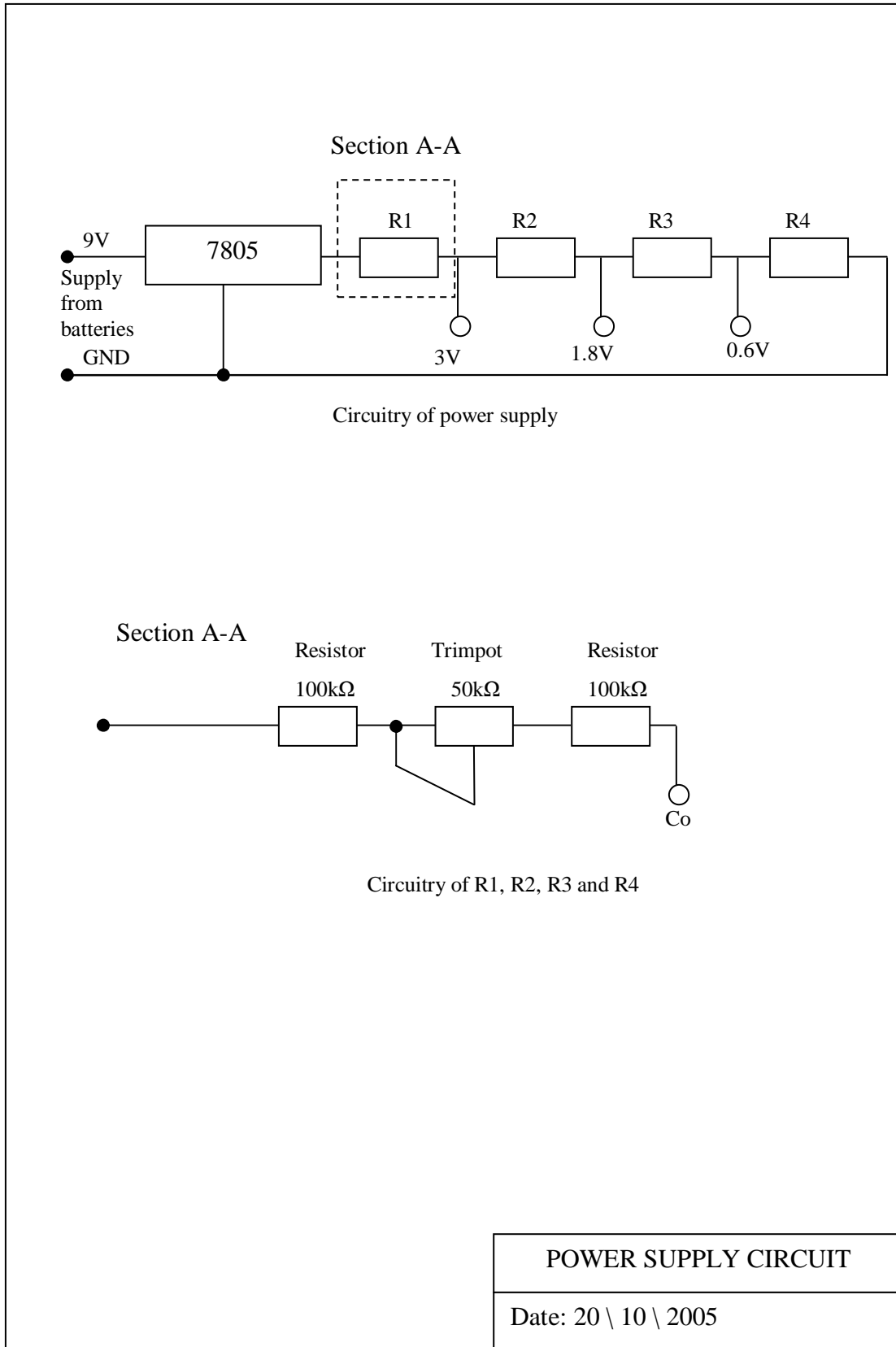
Appendix C

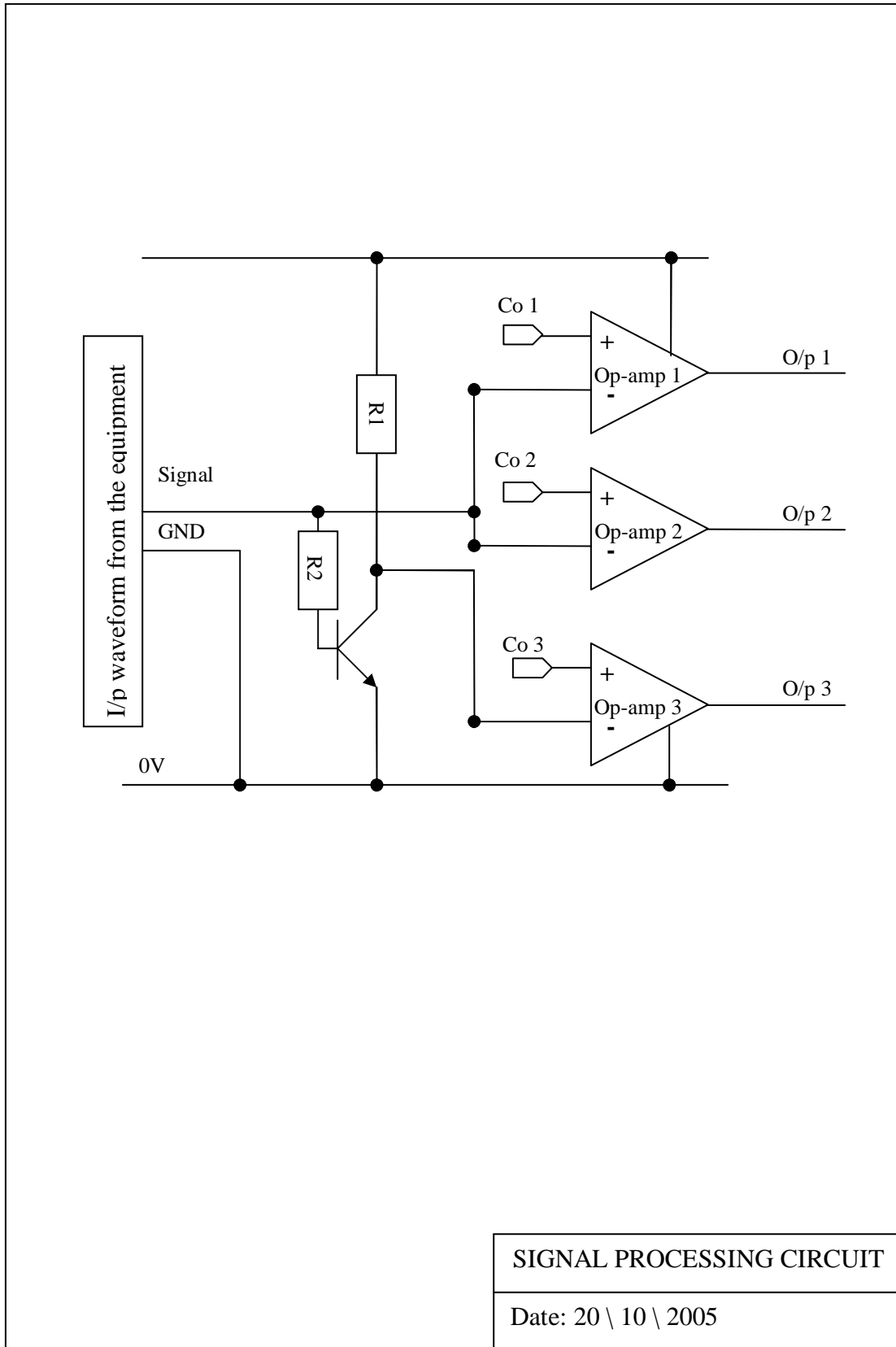
Circuit Diagram

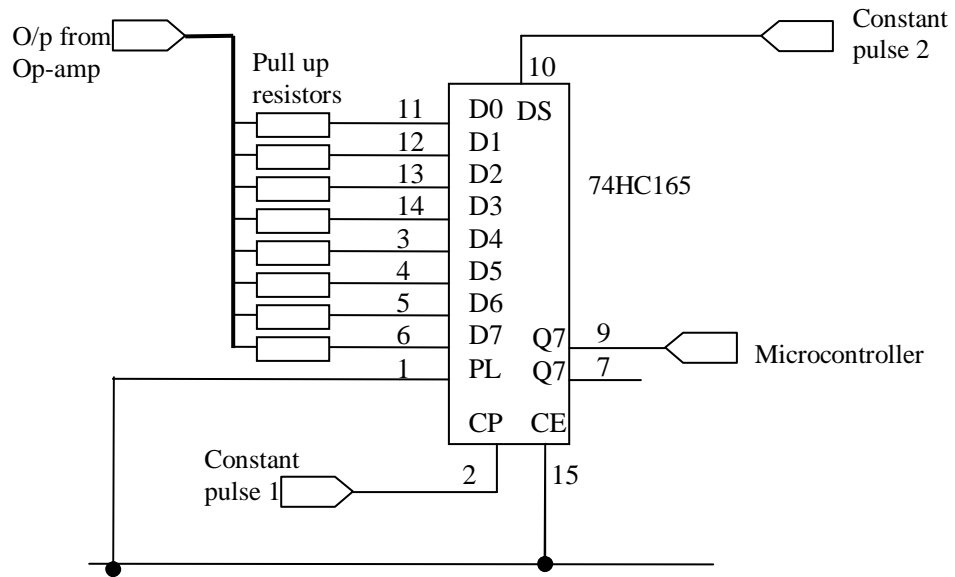


RS-232 CIRCUIT

Date: 20 \ 10 \ 2005

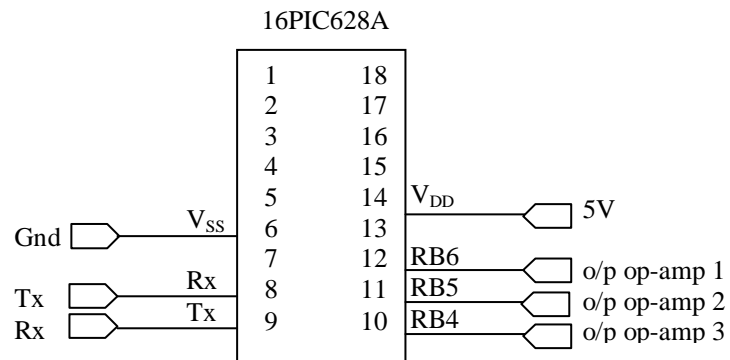






74HC165 CIRCUIT

Date: 20 \ 10 \ 2005



MICROCONTROLLER CIRCUIT

Date: 20 \ 10 \ 2005

Appendix D

MPLAB Code

```

    title "PIC16F628 counting program"
    list p=16f628,f=inhx32
    #include <p16f628.inc>
; This "header file" contains all
; the PIC16F628 special function
; register names and addresses.

```

D.1 Define the variable

```

count equ 0x20
temp equ 0x21
MAX equ 0x22
dcount equ 0x23

```

```

    org 00h                ;reset vector
    goto Start
    org 04h                ;interrupt vector
    goto Interrupt
    org 1Ch

```

D.2 Initialize the program

Start

```

    BCF STATUS, RP1
    BCF STATUS, RP0        ;Select Bank0
    movlw 0x00             ;clear W register
    movwf PORTA           ;clear PORTA
    movwf PORTB           ;clear PORTB
    MOVLW 0x07            ;Turn comparators off and
    MOVWF CMCON           ;enable pins for I/O

    BCF STATUS, RP1
    BSF STATUS, RP0        ; Select Bank1
    movlw 0x00             ;clear W register
    movwf TRISA           ;config PORTA as outputs
    movlw 0xFF
    movwf TRISB           ;config PORTB as inputs
    movlw 0x88
    movwf INTCON          ;turn on PORTB change interrupt

    movlw 0x24            ; BRGH = 1
    movwf TXSTA           ; turn on USART TX
    movlw 25              ; BCF STATUS, RP1
    movwf SPBRG           ; x with INT RC 4MHz + BRGH = 1

    BCF STATUS, RP1

```



```

BCF STATUS, RP0      ; Select Bank0
movlw 0x80            ;
movwf RCSTA          ; turn on USART + RX

```

D.3 Initialize the variable

Init

```

BCF STATUS, RP1
BCF STATUS, RP0      ;Select Bank0
clrf count           ;clr count
clrf temp            ;clr temp
clrf MAX

```

IncCount

```

movlw 0x18           ;00011000 3 x 8 = 24 set
movwf MAX            ;
movwf count          ;

```

D.4 Counter

IntCount1

```

movfw MAX
subwf count
decfsz count,0      ;decrement MAX by 1 if =0 nextset if =1 getdata
goto getdata
movfw MAX
movwf count
goto IntCount1
goto waitnext

movfw PORTB        ;
goto IntCount1    ;loop

```

D.5 Data processing

getdata

```

movfw PORTB
movwf temp          ;buffer
rlf temp,0         ;right shift 1 bit
movfw temp          ;move to working register
andlw 0xe0         ;11100000 to remove other 5 bit
movwf temp         ;move to store
decfsz MAX,1       ;decrement MAX by 1 if =0 nextset if =1 getdata
goto IntCount1
goto waitnext

```

```
waitnext
    movfw PORTB           ;read portb
    btfss PORTB,0        ;check for the next button(hardware)
    goto waitnext
    goto IncCount ;loop
```

D.6 Receive data

Interrupt

delay

```
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
```

```
movfw PORTB
movwf temp           ;buffer
rlf temp,0           ;right shift 1 bit
movfw temp           ;move to working register
andlw 0xe0           ;11100000 to remove other 5 bit
movwf temp           ;move to store
```

```
movfw temp           ; clear PORTB change
movwf TXREG          ; send byte on serial TX
bcf INTCON, RBIF     ; clear RBIF portb change flag
retfie               ; return from interrupt
```

```
end                  ;end of assembly
```

Appendix E

Visual Basic Code

E1 New User.vb

```
Public Class frmnewuser
    Inherits System.Windows.Forms.Form

    Structure structUser
        <VBFixedString(25)> Dim FirstName As String
        <VBFixedString(25)> Dim LastName As String
        <VBFixedString(6)> Dim Sex As String
        <VBFixedString(20)> Dim AgeGroup As String
        <VBFixedString(35)> Dim Address As String
        <VBFixedString(20)> Dim PostCode As String
        <VBFixedString(20)> Dim City As String
        <VBFixedString(20)> Dim Country As String
        <VBFixedString(20)> Dim HomePhone As String
        <VBFixedString(20)> Dim MobilePhone As String
    End Structure

    Structure userdata
        <VBFixedString(6)> Dim sys As String
        <VBFixedString(6)> Dim dia As String
        <VBFixedString(6)> Dim pulse As String
        <VBFixedString(10)> Dim datadate As String
        <VBFixedString(10)> Dim datatime As String
    End Structure

    Structure userdatal
        <VBFixedString(6)> Dim sys As String
        <VBFixedString(6)> Dim dia As String
        <VBFixedString(6)> Dim pulse As String
        <VBFixedString(10)> Dim datadate As String
        <VBFixedString(10)> Dim datatime As String
    End Structure

    Dim sys1(7) As String
    Dim sys2(7) As String
    Dim sys3(7) As String
    Dim dial(7) As String
    Dim dia2(7) As String
    Dim pull(7) As String
    Dim pul2(7) As String
    Dim pul3(7) As String

    Dim arr(2, 32) As String

    Private Sub radfemale_CheckedChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs)
        radmale.Checked = False
        radfemale.Checked = True
    End Sub

    Private Sub radmale_CheckedChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs)
        radmale.Checked = True
        radfemale.Checked = False
    End Sub
```

```
Private Sub btnnext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnnext1.Click
    TabControl1.SelectedTab = PartConfirm
    Me.TextBox1.Text = Me.txtfirstname.Text & " " &
Me.txtfamilyname.Text
    If radmale.Checked Then Me.TextBox2.Text = "Male"
    If radfemale.Checked Then Me.TextBox2.Text = "Female"
    If Me.cboagegroup.SelectedIndex = 0 Then
        TextBox8.Text = Me.cboagegroup.Text
    ElseIf Me.cboagegroup.SelectedIndex = 1 Then
        TextBox8.Text = Me.cboagegroup.Text
    ElseIf Me.cboagegroup.SelectedIndex = 2 Then
        TextBox8.Text = Me.cboagegroup.Text
    ElseIf Me.cboagegroup.SelectedIndex = 3 Then
        TextBox8.Text = Me.cboagegroup.Text
    ElseIf Me.cboagegroup.SelectedIndex = 4 Then
        TextBox8.Text = Me.cboagegroup.Text
    ElseIf Me.cboagegroup.SelectedIndex = 5 Then
        TextBox8.Text = Me.cboagegroup.Text
    ElseIf Me.cboagegroup.SelectedIndex = 6 Then
        TextBox8.Text = Me.cboagegroup.Text
    Else : TextBox8.Text = ""
    End If
    Me.TextBox3.Text = Me.txtaddress.Text
    Me.TextBox4.Text = Me.txtpost.Text
    Me.TextBox5.Text = Me.txtcity.Text
    Me.TextBox9.Text = Me.txtcountry.Text
    Me.TextBox6.Text = Me.txtthome.Text
    Me.TextBox7.Text = Me.txtthp.Text

End Sub

Private Sub btnexit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnexit.Click
    Me.Dispose()
End Sub

Private Sub btnprevious_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnprevious1.Click
    TabControl1.SelectedTab = PartEntry
End Sub

Private Sub btnnext2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnnext2.Click
    TabControl1.SelectedTab = Download
    Dim PRec As New structUser
    Dim strFileName As String
    strFileName = "currentuser.txt"
    FileOpen(1, strFileName, OpenMode.Output)
    PRec.FirstName = txtfirstname.Text
    PRec.LastName = txtfamilyname.Text
    PRec.Sex = TextBox2.Text
    PRec.AgeGroup = TextBox8.Text
    PRec.Address = TextBox3.Text
    PRec.PostCode = TextBox4.Text
    PRec.City = TextBox5.Text
```

```
    PRec.Country = TextBox9.Text
    PRec.HomePhone = TextBox6.Text
    PRec.MobilePhone = TextBox7.Text
    WriteLine(1, PRec.FirstName)
    WriteLine(1, PRec.LastName)
    WriteLine(1, PRec.Sex)
    WriteLine(1, PRec.AgeGroup)
    WriteLine(1, PRec.Address)
    WriteLine(1, PRec.PostCode)
    WriteLine(1, PRec.City)
    WriteLine(1, PRec.Country)
    WriteLine(1, PRec.HomePhone)
    WriteLine(1, PRec.MobilePhone)
    FileClose(1)
    loaddata()
End Sub

Private Sub btnload_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnload.Click
    Dim PRec As New structUser
    Dim strFileName As String
    strFileName = "currentuser.txt"
    FileOpen(1, strFileName, OpenMode.Input)
    PRec = New structUser
    Input(1, PRec.FirstName)
    Input(1, PRec.LastName)
    Input(1, PRec.Sex)
    Input(1, PRec.AgeGroup)
    Input(1, PRec.Address)
    Input(1, PRec.PostCode)
    Input(1, PRec.City)
    Input(1, PRec.Country)
    Input(1, PRec.HomePhone)
    Input(1, PRec.MobilePhone)
    FileClose(1)
    TextBox1.Text = PRec.FirstName & " " & PRec.LastName
    TextBox2.Text = PRec.Sex
    TextBox8.Text = PRec.AgeGroup
    TextBox3.Text = PRec.Address
    TextBox4.Text = PRec.PostCode
    TextBox5.Text = PRec.City
    TextBox9.Text = PRec.Country
    TextBox6.Text = PRec.HomePhone
    TextBox7.Text = PRec.MobilePhone
End Sub

Private Sub frmnewuser_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
    If System.IO.File.Exists("currentuser.txt") Then
        Dim PRec As New structUser
        FileOpen(1, "currentuser.txt", OpenMode.Input)
        Input(1, PRec.FirstName)
        Input(1, PRec.LastName)
        Input(1, PRec.Sex)
        Input(1, PRec.AgeGroup)
        Input(1, PRec.Address)
        Input(1, PRec.PostCode)
```

```
        Input(1, PRec.City)
        Input(1, PRec.Country)
        Input(1, PRec.HomePhone)
        Input(1, PRec.MobilePhone)
        FileClose(1)
        TextBox1.Text = PRec.FirstName & " " & PRec.LastName
        TextBox2.Text = PRec.Sex
        TextBox8.Text = PRec.AgeGroup
        TextBox3.Text = PRec.Address
        TextBox4.Text = PRec.PostCode
        TextBox5.Text = PRec.City
        TextBox9.Text = PRec.Country
        TextBox6.Text = PRec.HomePhone
        TextBox7.Text = PRec.MobilePhone
        TabControl1.SelectedTab = Download
        If System.IO.File.Exists("currentuserdata.txt") Then
            loadgraph()
            loaddata()
        End If
    Else
        TabControl1.SelectedTab = PartEntry
    End If

End Sub

Private Sub btnprevious2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprevious2.Click
    TabControl1.SelectedTab = DataDisplay
End Sub

Private Sub btnnext3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnnext3.Click
    TabControl1.SelectedTab = Graph
End Sub

Private Sub btnloadgra_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnloadgra1.Click
    Dim dtaFileName As String
    Dim i As String
    Dim length As Integer
    dtaFileName = "currentuserdata.txt"
    FileOpen(1, dtaFileName, OpenMode.Input)
    Do While Not EOF(1)
        Input(1, i)
        If i = "" Then
        Else
            length = length + 1
        End If
    Loop
    FileClose(1)
    length = length / 5
    Dim curdata As New userdata
    Dim dtacount As Integer
    Dim dtanumber As Integer
    dtanumber = length
    Dim arrData(dtanumber, 2) As Object
    curdata = New userdata
```

```

dtaFileName = "currentuserdata.txt"
FileOpen(1, dtaFileName, OpenMode.Input)
arrData(0, 1) = "Reading"
arrData(0, 2) = "SYS"
For dtacount = 1 To dtanumber
    Input(1, curdata.sys)
    Input(1, curdata.dia)
    Input(1, curdata.pulse)
    Input(1, curdata.datadate)
    Input(1, curdata.datatime)
    arrData(dtacount, 1) = "R" & dtacount
    arrData(dtacount, 2) = curdata.sys
Next
FileClose(1)
MSChart1.ChartData = arrData
With Me.MSChart1.Plot
    .SeriesCollection(1).DataPoints(-1).Brush.FillColor.Set(0,
0, 0) 'Black
End With

End Sub

Private Sub btnloadgra2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnloadgra2.Click
    Dim dtaFileName As String
    Dim i As String
    Dim length As Integer
    dtaFileName = "currentuserdata.txt"
    FileOpen(1, dtaFileName, OpenMode.Input)
    Do While Not EOF(1)
        Input(1, i)
        If i = "" Then
            Else
                length = length + 1
            End If
    Loop
    FileClose(1)
    length = length / 5
    Dim curdata As New userdata
    Dim dtacount As Integer
    Dim dtanumber As Integer
    dtanumber = length
    Dim arrData(dtanumber, 2) As Object
    curdata = New userdata
    dtaFileName = "currentuserdata.txt"
    FileOpen(1, dtaFileName, OpenMode.Input)
    arrData(0, 1) = "Reading"
    arrData(0, 2) = "DIA"
    For dtacount = 1 To dtanumber
        Input(1, curdata.sys)
        Input(1, curdata.dia)
        Input(1, curdata.pulse)
        Input(1, curdata.datadate)
        Input(1, curdata.datatime)
        arrData(dtacount, 1) = "R" & dtacount
        arrData(dtacount, 2) = curdata.dia
    Next

```



```

        FileClose(1)
        MSChart1.ChartData = arrData
        With Me.MSChart1.Plot
            .SeriesCollection(1).DataPoints(-1).Brush.FillColor.Set(0,
0, 255) 'Blue
        End With
    End Sub

```

```

    Private Sub btnloadgra3_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnloadgra3.Click
        Dim dtaFileName As String
        Dim i As String
        Dim length As Integer
        dtaFileName = "currentuserdata.txt"
        FileOpen(1, dtaFileName, OpenMode.Input)
        Do While Not EOF(1)
            Input(1, i)
            If i = "" Then
            Else
                length = length + 1
            End If
        Loop
        FileClose(1)
        length = length / 5
        Dim curdata As New userdata
        Dim dtacount As Integer
        Dim dtanumber As Integer
        dtanumber = length
        Dim arrData(dtanumber, 2) As Object
        curdata = New userdata
        dtaFileName = "currentuserdata.txt"
        FileOpen(1, dtaFileName, OpenMode.Input)
        arrData(0, 1) = "Reading"
        arrData(0, 2) = "PULSE"
        For dtacount = 1 To dtanumber
            Input(1, curdata.sys)
            Input(1, curdata.dia)
            Input(1, curdata.pulse)
            Input(1, curdata.datadate)
            Input(1, curdata.datatime)
            arrData(dtacount, 1) = "R" & dtacount
            arrData(dtacount, 2) = curdata.pulse
        Next
        FileClose(1)
        MSChart1.ChartData = arrData
        With Me.MSChart1.Plot
            .SeriesCollection(1).DataPoints(-1).Brush.FillColor.Set(0,
255, 0) 'Green
        End With
    End Sub

```

```

    Private Sub btnloadgra4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnloadgra4.Click
        loadgraph()
    End Sub
    Private Sub loadgraph()
        Dim dtaFileName As String

```

```

Dim i As String
Dim length As Integer
dtaFileName = "currentuserdata.txt"
FileOpen(1, dtaFileName, OpenMode.Input)
Do While Not EOF(1)
    Input(1, i)
    If i = "" Then
    Else
        length = length + 1
    End If
Loop
FileClose(1)
length = length / 5
Dim curdata As New userdata
Dim dtacount As Integer
Dim dtanumber As Integer
dtanumber = length
Dim arrData(dtanumber, 4) As Object
curdata = New userdata
dtaFileName = "currentuserdata.txt"
FileOpen(1, dtaFileName, OpenMode.Input)
arrData(0, 1) = "Reading"
arrData(0, 2) = "SYS"
arrData(0, 3) = "DIA"
arrData(0, 4) = "PULSE"
For dtacount = 1 To dtanumber
    Input(1, curdata.sys)
    Input(1, curdata.dia)
    Input(1, curdata.pulse)
    Input(1, curdata.datadate)
    Input(1, curdata.datatime)
    arrData(dtacount, 1) = "R" & dtacount
    arrData(dtacount, 2) = curdata.sys
    arrData(dtacount, 3) = curdata.dia
    arrData(dtacount, 4) = curdata.pulse
Next
FileClose(1)
MSChart1.ChartData = arrData

With MSChart1
    .chartType = MSChart20Lib.VtChChartType.VtChChartType2dLine
    .Legend.Location.LocationType =
MSChart20Lib.VtChLocationType.VtChLocationTypeRight
    .Legend.Location.Visible = True
End With
With Me.MSChart1.Plot
    .Axis(MSChart20Lib.VtChAxisId.VtChAxisIdX).AxisTitle.Text =
"Readings"
    .Axis(MSChart20Lib.VtChAxisId.VtChAxisIdY).AxisTitle.Text =
"mm/Hg"
    .SeriesCollection(1).DataPoints(-1).Brush.FillColor.Set(0,
0, 0) 'Black
    .SeriesCollection(2).DataPoints(-1).Brush.FillColor.Set(0,
0, 255) 'Blue
    .SeriesCollection(3).DataPoints(-1).Brush.FillColor.Set(0,
255, 0) 'Green
End With

```

```

End Sub
Private Sub loaddata()
    If ListView1.Items.Count = Nothing Then
        Dim dtaFileName As String
        Dim i As String
        Dim length As Integer
        dtaFileName = "currentuserdata.txt"
        FileOpen(1, dtaFileName, OpenMode.Input)
        Do While Not EOF(1)
            Input(1, i)
            If i = "" Then
            Else
                length = length + 1
            End If
        Loop
        FileClose(1)
        length = length / 5
        Dim curdata As New userdata
        Dim dtacount As Integer
        Dim dtanumber As Integer
        dtanumber = length
        curdata = New userdata
        dtaFileName = "currentuserdata.txt"
        FileOpen(1, dtaFileName, OpenMode.Input)
        For dtacount = 1 To dtanumber
            Input(1, curdata.sys)
            Input(1, curdata.dia)
            Input(1, curdata.pulse)
            Input(1, curdata.datadate)
            Input(1, curdata.datatime)
            ListView1.Items.Add(dtacount - 1)
            ListView1.Items(dtacount - 1).Text = dtacount
            ListView1.Items(dtacount - 1).SubItems.Add(curdata.sys)
            ListView1.Items(dtacount - 1).SubItems.Add(curdata.dia)
            ListView1.Items(dtacount -
1).SubItems.Add(curdata.pulse)
            ListView1.Items(dtacount -
1).SubItems.Add(curdata.datadate)
            ListView1.Items(dtacount -
1).SubItems.Add(curdata.datatime)
            cbonumberlist.Items.Add(dtacount)
        Next
        FileClose(1)
    Else
    End If
End Sub

Private Sub btnnext4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnnext4.Click
    TabControl1.SelectedTab = Print
End Sub

Private Sub btnloaddata_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnloaddata.Click
    loaddata()
End Sub

```

```

Private Sub btnprevious3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprevious3.Click
    TabControl1.SelectedTab = Graph
End Sub

```

```

Private Sub btnprevious4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprevious4.Click
    TabControl1.SelectedTab = PartConfirm
End Sub

```

```

Private Sub btnnext5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnnext5.Click
    TabControl1.SelectedTab = DataDisplay
End Sub

```

```

Private Sub btndeldata_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndeldata.Click

```

```

    Dim msg As String
    Dim title As String
    Dim style As MsgBoxStyle
    Dim response As MsgBoxResult
    Dim dtacount As Integer
    Dim i As Integer
    Dim j As Integer
    If cbonumberlist.SelectedItem = Nothing Then
        MsgBox("Please select a data to delete")
    Else
        msg = "Are you sure you want to remove these data?"
        style = MsgBoxStyle.DefaultButton2 Or _
            MsgBoxStyle.Critical Or MsgBoxStyle.YesNo
        title = "MsgBox Demonstration" ' Define title.
        response = MsgBox(msg, style, title)
        If response = MsgBoxResult.Yes Then
            dtacount = cbonumberlist.SelectedIndex
            ListView1.Items.RemoveAt(dtacount)
            j = ListView1.Items.Count
            For i = dtacount To j - 1
                ListView1.Items(i).Text = ListView1.Items(i).Text -

```

1

```

                Next
            Else
            End If
            updatedata()
        End If

```

```

End Sub

```

```

Private Sub btndeditdata_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

```

```

    'Dim dtacount As Integer
    'Dim i As Integer
    'Dim j As Integer

    'If cbonumberlist.SelectedItem = Nothing Then
    '    MsgBox("Please select a data to edit")
    'Else

```

```

    ' dtacount = cbonumberlist.SelectedIndex
    'ListView1.Items(dtacount).SubItems.IsReadOnly = False
    'j = ListView1.Items.Count
    'For i = dtacount To j - 1
    'ListView1.Items(i).Text = ListView1.Items(i).Text - 1
    'Next
    'updatedata()
    'End If
End Sub
Private Sub updatedata()

    Dim strFileName As String
    Dim i As Integer
    Dim j As Integer
    strFileName = "currentuserdata.txt"
    FileOpen(1, strFileName, OpenMode.Output)
    j = ListView1.Items.Count

    For i = 0 To j - 1
        WriteLine(1, ListView1.Items(i).SubItems(1).Text)
        WriteLine(1, ListView1.Items(i).SubItems(2).Text)
        WriteLine(1, ListView1.Items(i).SubItems(3).Text)
        WriteLine(1, ListView1.Items(i).SubItems(4).Text)
        WriteLine(1, ListView1.Items(i).SubItems(5).Text)
    Next
    FileClose(1)
End Sub
Private Sub btndownload_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btndownload.Click
    loaddataraw()
    sysfig()
    sysnum()
    diafig()
    dianum()
    pulfig()
    pulnum()
End Sub
Private Sub loaddataraw()
    Dim dtaFileName As String
    Dim temp As String
    Dim dtacount As Integer = 1
    Dim i As Integer = 0
    Dim j As Integer = 0
    Dim k As Integer = 0
    ListBoxtest.Items.Clear()
    ListBoxtest1.Items.Clear()
    ListBoxtest2.Items.Clear()
    dtaFileName = "currentuserdataraw.txt"
    FileOpen(1, dtaFileName, OpenMode.Input)
    For dtacount = 1 To 99
        Input(1, temp)
        Select Case (k)
            Case 0
                arr(i, j) = temp
            Case 1
                arr(i, j) = temp
            Case 2

```

```
        arr(i, j) = temp
    End Select
    j = j + 1
    If j = 33 Then
        i = i + 1
        j = 0
    ElseIf i = 4 Then
        Exit For
    End If
    k = k + 1
    If k = 3 Then
        k = 0
    End If
Next
FileClose(1)
End Sub

Private Sub sysfig()

    Dim dtacount As Integer
    For dtacount = 0 To 32
        ListBoxtest.Items.Add(arr(0, dtacount))
        ListBoxtest2.Items.Add(arr(2, dtacount))
    Next

    Dim upina1 As Integer
    Dim upina2 As Integer
    Dim upina3 As Integer
    Dim upina4 As Integer
    Dim upinb1 As Integer
    Dim upinb2 As Integer
    Dim upinb3 As Integer
    Dim upinb4 As Integer
    Dim upinc1 As Integer
    Dim upinc2 As Integer
    Dim upinc3 As Integer
    Dim upinc4 As Integer
    Dim upind1 As Integer
    Dim upind2 As Integer
    Dim upind3 As Integer
    Dim upind4 As Integer
    Dim upine1 As Integer
    Dim upine2 As Integer
    Dim upine3 As Integer
    Dim upine4 As Integer

    Dim lpina1 As Integer
    Dim lpina2 As Integer
    Dim lpina3 As Integer
    Dim lpina4 As Integer
    Dim lpinb1 As Integer
    Dim lpinb2 As Integer
    Dim lpinb3 As Integer
    Dim lpinb4 As Integer
    Dim lpinc1 As Integer
    Dim lpinc2 As Integer
```

```
Dim lpinc3 As Integer
Dim lpinc4 As Integer
Dim lpind1 As Integer
Dim lpind2 As Integer
Dim lpind3 As Integer
Dim lpind4 As Integer
Dim lpine1 As Integer
Dim lpine2 As Integer
Dim lpine3 As Integer
Dim lpine4 As Integer

Dim i As Integer = 3
Dim j As Integer = 4
Dim k As Integer = 5
Dim l As Integer = 6
Dim n As Integer = 7

upina1 = arr(0, i) / 10000000
upina2 = arr(0, i) / 1000000
upina3 = arr(0, i) / 100000
upina4 = arr(0, i) / 10000
lpina1 = arr(2, i) / 1000
lpina2 = arr(2, i) / 100
lpina3 = arr(2, i) / 10
lpina4 = arr(2, i)

upinb1 = arr(0, j) / 10000000
upinb2 = arr(0, j) / 1000000
upinb3 = arr(0, j) / 100000
upinb4 = arr(0, j) / 10000
lpinb1 = arr(2, j) / 1000
lpinb2 = arr(2, j) / 100
lpinb3 = arr(2, j) / 10
lpinb4 = arr(2, j)

upinc1 = arr(0, k) / 10000000
upinc2 = arr(0, k) / 1000000
upinc3 = arr(0, k) / 100000
upinc4 = arr(0, k) / 10000
lpinc1 = arr(2, k) / 1000
lpinc2 = arr(2, k) / 100
lpinc3 = arr(2, k) / 10
lpinc4 = arr(2, k)

upind1 = arr(0, l) / 10000000
upind2 = arr(0, l) / 1000000
upind3 = arr(0, l) / 100000
upind4 = arr(0, l) / 10000
lpind1 = arr(2, l) / 1000
lpind2 = arr(2, l) / 100
lpind3 = arr(2, l) / 10
lpind4 = arr(2, l)

upine1 = arr(0, n) / 10000000
upine2 = arr(0, n) / 1000000
upine3 = arr(0, n) / 100000
upine4 = arr(0, n) / 10000
```

```

lpine1 = arr(2, n) / 1000
lpine2 = arr(2, n) / 100
lpine3 = arr(2, n) / 10
lpine4 = arr(2, n)

'for pin 3 for third digit
If upina1 = 1 And lpina1 = 1 Then
    ListBoxtest1.Items.Add("1a")
    sys1(0) = 1
End If
If upina2 = 0 Or lpina2 = 0 Then
ElseIf upina2 = 1 Or upina2 = 11 And lpina2 = 1 Or lpina2 = 11
Then
    ListBoxtest1.Items.Add("1b")
    sys1(1) = 1
End If

If upina3 = 0 Or lpina3 = 0 Then
ElseIf upina3 = 1 Or upina3 = 11 Or upina3 = 111 Or upina3 =
101 And lpina3 = 1 Or lpina3 = 11 Or lpina3 = 111 Or lpina3 = 101 Then
    sys1(2) = 1
    ListBoxtest1.Items.Add("1c")
End If

If upina4 = 0 Or lpina4 = 0 Then
ElseIf upina4 = 1 Or upina4 = 11 Or upina4 = 111 Or upina4 =
101 Or upina4 = 1001 Or upina4 = 1011 Or upina4 = 1111 Or upina4 = 1101
And lpina4 = 1 Or lpina4 = 11 Or lpina4 = 111 Or lpina4 = 101 Or lpina4
= 1001 Or lpina4 = 1011 Or lpina4 = 1111 Or lpina4 = 1101 Then
    sys1(3) = 1
    ListBoxtest1.Items.Add("1d")
End If
'for pin 4 for third digit
If upinb1 = 1 And lpinb1 = 1 Then
    ListBoxtest1.Items.Add("1e")
    sys1(4) = 1
End If
If upinb2 = 0 Or lpinb2 = 0 Then
ElseIf upinb2 = 1 Or upinb2 = 11 And lpinb2 = 1 Or lpinb2 = 11
Then
    ListBoxtest1.Items.Add("1f")
    sys1(5) = 1
End If
If upinb3 = 0 Or lpinb3 = 0 Then
ElseIf upinb3 = 1 Or upinb3 = 11 Or upinb3 = 111 Or upinb3 =
101 And lpinb3 = 1 Or lpinb3 = 11 Or lpinb3 = 111 Or lpinb3 = 101 Then
    sys1(6) = 1
    ListBoxtest1.Items.Add("1g")
End If
If upinb4 = 0 Or lpinb4 = 0 Then
ElseIf upinb4 = 1 Or upinb4 = 11 Or upinb4 = 111 Or upinb4 =
101 Or upinb4 = 1001 Or upinb4 = 1011 Or upinb4 = 1111 Or upinb4 = 1101
And lpinb4 = 1 Or lpinb4 = 11 Or lpinb4 = 111 Or lpinb4 = 101 Or lpinb4
= 1001 Or lpinb4 = 1011 Or lpinb4 = 1111 Or lpinb4 = 1101 Then
    sys1(7) = 1
    ListBoxtest1.Items.Add("1h")
End If

```



```

'for pin 5 for third digit
If upinc1 = 1 And lpinc1 = 1 Then
    ListBoxtest1.Items.Add("2a")
    sys2(0) = 1
End If
If upinc2 = 0 Or lpinc2 = 0 Then
ElseIf upinc2 = 1 Or upinc2 = 11 And lpinc2 = 1 Or lpinc2 = 11
Then
    ListBoxtest1.Items.Add("2b")
    sys2(1) = 1
End If
If upinc3 = 0 Or lpinc3 = 0 Then
ElseIf upinc3 = 1 Or upinc3 = 11 Or upinc3 = 111 Or upinc3 =
101 And lpinc3 = 1 Or lpinc3 = 11 Or lpinc3 = 111 Or lpinc3 = 101 Then
    sys2(2) = 1
    ListBoxtest1.Items.Add("2c")
End If
If upinc4 = 0 Or lpinc4 = 0 Then
ElseIf upinc4 = 1 Or upinc4 = 11 Or upinc4 = 111 Or upinc4 =
101 Or upinc4 = 1001 Or upinc4 = 1011 Or upinc4 = 1111 Or upinc4 = 1101
And lpinc4 = 1 Or lpinc4 = 11 Or lpinc4 = 111 Or lpinc4 = 101 Or lpinc4
= 1001 Or lpinc4 = 1011 Or lpinc4 = 1111 Or lpinc4 = 1101 Then
    sys2(3) = 1
    ListBoxtest1.Items.Add("2d")
End If
'for pin 6 for third digit
If upind1 = 1 And lpind1 = 1 Then
    ListBoxtest1.Items.Add("2e")
    sys2(4) = 1
End If
If upind2 = 0 Or lpind2 = 0 Then
ElseIf upind2 = 1 Or upind2 = 11 And lpind2 = 1 Or lpind2 = 11
Then
    ListBoxtest1.Items.Add("2f")
    sys2(5) = 1
End If
If upind3 = 0 Or lpind3 = 0 Then
ElseIf upind3 = 1 Or upind3 = 11 Or upind3 = 111 Or upind3 =
101 And lpind3 = 1 Or lpind3 = 11 Or lpind3 = 111 Or lpind3 = 101 Then
    sys2(6) = 1
    ListBoxtest1.Items.Add("2g")
End If
If upind4 = 0 Or lpind4 = 0 Then
ElseIf upind4 = 1 Or upind4 = 11 Or upind4 = 111 Or upind4 =
101 Or upind4 = 1001 Or upind4 = 1011 Or upind4 = 1111 Or upind4 = 1101
And lpind4 = 1 Or lpind4 = 11 Or lpind4 = 111 Or lpind4 = 101 Or lpind4
= 1001 Or lpind4 = 1011 Or lpind4 = 1111 Or lpind4 = 1101 Then
    sys2(7) = 1
    ListBoxtest1.Items.Add("2h")
End If
'for pin 7 for third digit
If upine1 = 1 And lpine1 = 1 Then
    ListBoxtest1.Items.Add("3a")
    sys3(0) = 1
End If
If upine2 = 0 Or lpine2 = 0 Then

```

```

        ElseIf upine2 = 1 Or upine2 = 11 And lpine2 = 1 Or lpine2 = 11
Then
        ListBoxtest1.Items.Add("3b")
        sys3(1) = 1
    End If

    If upine3 = 0 Or lpine3 = 0 Then
    ElseIf upine3 = 1 Or upine3 = 11 Or upine3 = 111 Or upine3 =
101 And lpine3 = 1 Or lpine3 = 11 Or lpine3 = 111 Or lpine3 = 101 Then
        sys3(2) = 1
        ListBoxtest1.Items.Add("3c")
    End If

    If upine4 = 0 Or lpine4 = 0 Then
    ElseIf upine4 = 1 Or upine4 = 11 Or upine4 = 111 Or upine4 =
101 Or upine4 = 1001 Or upine4 = 1011 Or upine4 = 1111 Or upine4 = 1101
And lpine4 = 1 Or lpine4 = 11 Or lpine4 = 111 Or lpine4 = 101 Or lpine4
= 1001 Or lpine4 = 1011 Or lpine4 = 1111 Or lpine4 = 1101 Then
        sys3(3) = 1
        ListBoxtest1.Items.Add("3d")
    End If
End Sub

Private Sub sysnum()
    Dim currdata As New userdata1
    currdata.sys = 0
    'for pin 3 and 4 for third digit
    If sys1(0) = 1 And sys1(1) = 1 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 1 And sys1(5) = 0 And sys1(6) = 1 And sys1(7) = 1 Then
        '0
        currdata.sys = currdata.sys + 0
    End If
    If sys1(0) = 0 And sys1(1) = 1 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 0 And sys1(5) = 0 And sys1(6) = 0 And sys1(7) = 0 Then
        '1
        currdata.sys = currdata.sys + 1
    End If
    If sys1(0) = 1 And sys1(1) = 1 And sys1(2) = 0 And sys1(3) = 0
And sys1(4) = 0 And sys1(5) = 1 And sys1(6) = 1 And sys1(7) = 1 Then
        '2
        currdata.sys = currdata.sys + 2
    End If
    If sys1(0) = 1 And sys1(1) = 1 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 0 And sys1(5) = 1 And sys1(6) = 0 And sys1(7) = 1 Then
        '3
        currdata.sys = currdata.sys + 3
    End If
    If sys1(0) = 0 And sys1(1) = 1 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 1 And sys1(5) = 1 And sys1(6) = 0 And sys1(7) = 0 Then
        '4
        currdata.sys = currdata.sys + 4
    End If
    If sys1(0) = 1 And sys1(1) = 0 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 1 And sys1(5) = 1 And sys1(6) = 0 And sys1(7) = 1 Then
        '5
        currdata.sys = currdata.sys + 5
    End If

```

```
    If sys1(0) = 1 And sys1(1) = 0 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 1 And sys1(5) = 1 And sys1(6) = 1 And sys1(7) = 1 Then
    '6
    currdata.sys = currdata.sys + 6
End If
    If sys1(0) = 1 And sys1(1) = 1 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 0 And sys1(5) = 0 And sys1(6) = 0 And sys1(7) = 0 Then
    '7
    currdata.sys = currdata.sys + 7
End If
    If sys1(0) = 1 And sys1(1) = 1 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 1 And sys1(5) = 1 And sys1(6) = 1 And sys1(7) = 1 Then
    '8
    currdata.sys = currdata.sys + 8
End If
    If sys1(0) = 1 And sys1(1) = 1 And sys1(2) = 1 And sys1(3) = 0
And sys1(4) = 1 And sys1(5) = 1 And sys1(6) = 0 And sys1(7) = 1 Then
    '9
    currdata.sys = currdata.sys + 9
End If

'for pin 5 and 6 for second digit
    If sys2(0) = 1 And sys2(1) = 1 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 1 And sys2(5) = 0 And sys2(6) = 1 And sys2(7) = 1 Then
    '0
    currdata.sys = currdata.sys + 0
End If
    If sys2(0) = 0 And sys2(1) = 1 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 0 And sys2(5) = 0 And sys2(6) = 0 And sys2(7) = 0 Then
    '1
    currdata.sys = currdata.sys + 10
End If
    If sys2(0) = 1 And sys2(1) = 1 And sys2(2) = 0 And sys2(3) = 0
And sys2(4) = 0 And sys2(5) = 1 And sys2(6) = 1 And sys2(7) = 1 Then
    '2
    currdata.sys = currdata.sys + 20
End If
    If sys2(0) = 1 And sys2(1) = 1 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 0 And sys2(5) = 1 And sys2(6) = 0 And sys2(7) = 1 Then
    '3
    currdata.sys = currdata.sys + 30
End If
    If sys2(0) = 0 And sys2(1) = 1 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 1 And sys2(5) = 1 And sys2(6) = 0 And sys2(7) = 0 Then
    '4
    currdata.sys = currdata.sys + 40
End If
    If sys2(0) = 1 And sys2(1) = 0 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 1 And sys2(5) = 1 And sys2(6) = 0 And sys2(7) = 1 Then
    '5
    currdata.sys = currdata.sys + 50
End If
    If sys2(0) = 1 And sys2(1) = 0 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 1 And sys2(5) = 1 And sys2(6) = 1 And sys2(7) = 1 Then
    '6
    currdata.sys = currdata.sys + 60
End If
```

```
    If sys2(0) = 1 And sys2(1) = 1 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 0 And sys2(5) = 0 And sys2(6) = 0 And sys2(7) = 0 Then
    '7
    currdata.sys = currdata.sys + 70
End If
    If sys2(0) = 1 And sys2(1) = 1 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 1 And sys2(5) = 1 And sys2(6) = 1 And sys2(7) = 1 Then
    '8
    currdata.sys = currdata.sys + 80
End If
    If sys2(0) = 1 And sys2(1) = 1 And sys2(2) = 1 And sys2(3) = 0
And sys2(4) = 1 And sys2(5) = 1 And sys2(6) = 0 And sys2(7) = 1 Then
    '9
    currdata.sys = currdata.sys + 90
End If

'for pin 7 for first digit
If sys3(1) = 1 And sys3(2) = 1 Then
    '1
    currdata.sys = currdata.sys + 100
End If

ListBoxtest1.Items.Add(currdata.sys)
End Sub

Private Sub diafig()

    Dim upina1 As Integer
    Dim upina2 As Integer
    Dim upina3 As Integer
    Dim upina4 As Integer
    Dim upinb1 As Integer
    Dim upinb2 As Integer
    Dim upinb3 As Integer
    Dim upinb4 As Integer
    Dim upinc1 As Integer
    Dim upinc2 As Integer
    Dim upinc3 As Integer
    Dim upinc4 As Integer
    Dim upind1 As Integer
    Dim upind2 As Integer
    Dim upind3 As Integer
    Dim upind4 As Integer

    Dim lpina1 As Integer
    Dim lpina2 As Integer
    Dim lpina3 As Integer
    Dim lpina4 As Integer
    Dim lpinb1 As Integer
    Dim lpinb2 As Integer
    Dim lpinb3 As Integer
    Dim lpinb4 As Integer
    Dim lpinc1 As Integer
    Dim lpinc2 As Integer
    Dim lpinc3 As Integer
    Dim lpinc4 As Integer
    Dim lpind1 As Integer
```

```
Dim lpind2 As Integer
Dim lpind3 As Integer
Dim lpind4 As Integer

Dim i As Integer = 8
Dim j As Integer = 9
Dim k As Integer = 10
Dim l As Integer = 11

upina1 = arr(0, i) / 10000000
upina2 = arr(0, i) / 1000000
upina3 = arr(0, i) / 100000
upina4 = arr(0, i) / 10000
lpina1 = arr(2, i) / 1000
lpina2 = arr(2, i) / 100
lpina3 = arr(2, i) / 10
lpina4 = arr(2, i)

upinb1 = arr(0, j) / 10000000
upinb2 = arr(0, j) / 1000000
upinb3 = arr(0, j) / 100000
upinb4 = arr(0, j) / 10000
lpinb1 = arr(2, j) / 1000
lpinb2 = arr(2, j) / 100
lpinb3 = arr(2, j) / 10
lpinb4 = arr(2, j)

upinc1 = arr(0, k) / 10000000
upinc2 = arr(0, k) / 1000000
upinc3 = arr(0, k) / 100000
upinc4 = arr(0, k) / 10000
lpinc1 = arr(2, k) / 1000
lpinc2 = arr(2, k) / 100
lpinc3 = arr(2, k) / 10
lpinc4 = arr(2, k)

upind1 = arr(0, l) / 10000000
upind2 = arr(0, l) / 1000000
upind3 = arr(0, l) / 100000
upind4 = arr(0, l) / 10000
lpind1 = arr(2, l) / 1000
lpind2 = arr(2, l) / 100
lpind3 = arr(2, l) / 10
lpind4 = arr(2, l)

'for pin 8 for third digit
If upina1 = 1 And lpina1 = 1 Then
    ListBoxtest1.Items.Add("1a")
    dial(0) = 1
End If
If upina2 = 0 Or lpina2 = 0 Then
ElseIf upina2 = 1 Or upina2 = 11 And lpina2 = 1 Or lpina2 = 11
Then
    ListBoxtest1.Items.Add("1b")
    dial(1) = 1
End If
```

```

    If upina3 = 0 Or lpina3 = 0 Then
    ElseIf upina3 = 1 Or upina3 = 11 Or upina3 = 111 Or upina3 =
101 And lpina3 = 1 Or lpina3 = 11 Or lpina3 = 111 Or lpina3 = 101 Then
        dial(2) = 1
        ListBoxtest1.Items.Add("1c")
    End If

    If upina4 = 0 Or lpina4 = 0 Then
    ElseIf upina4 = 1 Or upina4 = 11 Or upina4 = 111 Or upina4 =
101 Or upina4 = 1001 Or upina4 = 1011 Or upina4 = 1111 Or upina4 = 1101
And lpina4 = 1 Or lpina4 = 11 Or lpina4 = 111 Or lpina4 = 101 Or lpina4
= 1001 Or lpina4 = 1011 Or lpina4 = 1111 Or lpina4 = 1101 Then
        dial(3) = 1
        ListBoxtest1.Items.Add("1d")
    End If
    'for pin 9 for third digit
    If upinb1 = 1 And lpinb1 = 1 Then
        ListBoxtest1.Items.Add("1e")
        dial(4) = 1
    End If
    If upinb2 = 0 Or lpinb2 = 0 Then
    ElseIf upinb2 = 1 Or upinb2 = 11 And lpinb2 = 1 Or lpinb2 = 11
Then
        ListBoxtest1.Items.Add("1f")
        dial(5) = 1
    End If
    If upinb3 = 0 Or lpinb3 = 0 Then
    ElseIf upinb3 = 1 Or upinb3 = 11 Or upinb3 = 111 Or upinb3 =
101 And lpinb3 = 1 Or lpinb3 = 11 Or lpinb3 = 111 Or lpinb3 = 101 Then
        dial(6) = 1
        ListBoxtest1.Items.Add("1g")
    End If
    If upinb4 = 0 Or lpinb4 = 0 Then
    ElseIf upinb4 = 1 Or upinb4 = 11 Or upinb4 = 111 Or upinb4 =
101 Or upinb4 = 1001 Or upinb4 = 1011 Or upinb4 = 1111 Or upinb4 = 1101
And lpinb4 = 1 Or lpinb4 = 11 Or lpinb4 = 111 Or lpinb4 = 101 Or lpinb4
= 1001 Or lpinb4 = 1011 Or lpinb4 = 1111 Or lpinb4 = 1101 Then
        dial(7) = 1
        ListBoxtest1.Items.Add("1h")
    End If
    'for pin 10 for third digit
    If upinc1 = 1 And lpincl = 1 Then
        ListBoxtest1.Items.Add("2a")
        dia2(0) = 1
    End If
    If upinc2 = 0 Or lpinc2 = 0 Then
    ElseIf upinc2 = 1 Or upinc2 = 11 And lpinc2 = 1 Or lpinc2 = 11
Then
        ListBoxtest1.Items.Add("2b")
        dia2(1) = 1
    End If
    If upinc3 = 0 Or lpinc3 = 0 Then
    ElseIf upinc3 = 1 Or upinc3 = 11 Or upinc3 = 111 Or upinc3 =
101 And lpinc3 = 1 Or lpinc3 = 11 Or lpinc3 = 111 Or lpinc3 = 101 Then
        dia2(2) = 1
        ListBoxtest1.Items.Add("2c")

```

```

        End If
        If upinc4 = 0 Or lpinc4 = 0 Then
        ElseIf upinc4 = 1 Or upinc4 = 11 Or upinc4 = 111 Or upinc4 =
101 Or upinc4 = 1001 Or upinc4 = 1011 Or upinc4 = 1111 Or upinc4 = 1101
And lpinc4 = 1 Or lpinc4 = 11 Or lpinc4 = 111 Or lpinc4 = 101 Or lpinc4
= 1001 Or lpinc4 = 1011 Or lpinc4 = 1111 Or lpinc4 = 1101 Then
            dia2(3) = 1
            ListBoxtest1.Items.Add("2d")
        End If
        'for pin 11 for third digit
        If upind1 = 1 And lpind1 = 1 Then
            ListBoxtest1.Items.Add("2e")
            dia2(4) = 1
        End If
        If upind2 = 0 Or lpind2 = 0 Then
        ElseIf upind2 = 1 Or upind2 = 11 And lpind2 = 1 Or lpind2 = 11
Then
            ListBoxtest1.Items.Add("2f")
            dia2(5) = 1
        End If
        If upind3 = 0 Or lpind3 = 0 Then
        ElseIf upind3 = 1 Or upind3 = 11 Or upind3 = 111 Or upind3 =
101 And lpind3 = 1 Or lpind3 = 11 Or lpind3 = 111 Or lpind3 = 101 Then
            dia2(6) = 1
            ListBoxtest1.Items.Add("2g")
        End If
        If upind4 = 0 Or lpind4 = 0 Then
        ElseIf upind4 = 1 Or upind4 = 11 Or upind4 = 111 Or upind4 =
101 Or upind4 = 1001 Or upind4 = 1011 Or upind4 = 1111 Or upind4 = 1101
And lpind4 = 1 Or lpind4 = 11 Or lpind4 = 111 Or lpind4 = 101 Or lpind4
= 1001 Or lpind4 = 1011 Or lpind4 = 1111 Or lpind4 = 1101 Then
            dia2(7) = 1
            ListBoxtest1.Items.Add("2h")
        End If

    End Sub
    Private Sub dianum()
        Dim currdata As New userdata1
        currdata.dia = 0
        'for pin 3 and 4 for third digit
        If dial(0) = 1 And dial(1) = 1 And dial(2) = 1 And dial(3) = 0
And dial(4) = 1 And dial(5) = 0 And dial(6) = 1 And dial(7) = 1 Then
            '0
            currdata.dia = currdata.dia + 0
        End If
        If dial(0) = 0 And dial(1) = 1 And dial(2) = 1 And dial(3) = 0
And dial(4) = 0 And dial(5) = 0 And dial(6) = 0 And dial(7) = 0 Then
            '1
            currdata.dia = currdata.dia + 1
        End If
        If dial(0) = 1 And dial(1) = 1 And dial(2) = 0 And dial(3) = 0
And dial(4) = 0 And dial(5) = 1 And dial(6) = 1 And dial(7) = 1 Then
            '2
            currdata.dia = currdata.dia + 2
        End If
        If dial(0) = 1 And dial(1) = 1 And dial(2) = 1 And dial(3) = 0
And dial(4) = 0 And dial(5) = 1 And dial(6) = 0 And dial(7) = 1 Then

```

```
        '3
        currdata.dia = currdata.dia + 3
    End If
    If dial(0) = 0 And dial(1) = 1 And dial(2) = 1 And dial(3) = 0
And dial(4) = 1 And dial(5) = 1 And dial(6) = 0 And dial(7) = 0 Then
        '4
        currdata.dia = currdata.dia + 4
    End If
    If dial(0) = 1 And dial(1) = 0 And dial(2) = 1 And dial(3) = 0
And dial(4) = 1 And dial(5) = 1 And dial(6) = 0 And dial(7) = 1 Then
        '5
        currdata.dia = currdata.dia + 5
    End If
    If dial(0) = 1 And dial(1) = 0 And dial(2) = 1 And dial(3) = 0
And dial(4) = 1 And dial(5) = 1 And dial(6) = 1 And dial(7) = 1 Then
        '6
        currdata.dia = currdata.dia + 6
    End If
    If dial(0) = 1 And dial(1) = 1 And dial(2) = 1 And dial(3) = 0
And dial(4) = 0 And dial(5) = 0 And dial(6) = 0 And dial(7) = 0 Then
        '7
        currdata.dia = currdata.dia + 7
    End If
    If dial(0) = 1 And dial(1) = 1 And dial(2) = 1 And dial(3) = 0
And dial(4) = 1 And dial(5) = 1 And dial(6) = 1 And dial(7) = 1 Then
        '8
        currdata.dia = currdata.dia + 8
    End If
    If dial(0) = 1 And dial(1) = 1 And dial(2) = 1 And dial(3) = 0
And dial(4) = 1 And dial(5) = 1 And dial(6) = 0 And dial(7) = 1 Then
        '9
        currdata.dia = currdata.dia + 9
    End If

    'for pin 5 and 6 for second digit
    If dia2(0) = 1 And dia2(1) = 1 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 1 And dia2(5) = 0 And dia2(6) = 1 And dia2(7) = 1 Then
        '0
        currdata.dia = currdata.dia + 0
    End If
    If dia2(0) = 0 And dia2(1) = 1 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 0 And dia2(5) = 0 And dia2(6) = 0 And dia2(7) = 0 Then
        '1
        currdata.dia = currdata.dia + 10
    End If
    If dia2(0) = 1 And dia2(1) = 1 And dia2(2) = 0 And dia2(3) = 0
And dia2(4) = 0 And dia2(5) = 1 And dia2(6) = 1 And dia2(7) = 1 Then
        '2
        currdata.dia = currdata.dia + 20
    End If
    If dia2(0) = 1 And dia2(1) = 1 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 0 And dia2(5) = 1 And dia2(6) = 0 And dia2(7) = 1 Then
        '3
        currdata.dia = currdata.dia + 30
    End If
    If dia2(0) = 0 And dia2(1) = 1 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 1 And dia2(5) = 1 And dia2(6) = 0 And dia2(7) = 0 Then
```



```
        '4
        currdata.dia = currdata.dia + 40
    End If
    If dia2(0) = 1 And dia2(1) = 0 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 1 And dia2(5) = 1 And dia2(6) = 0 And dia2(7) = 1 Then
        '5
        currdata.dia = currdata.dia + 50
    End If
    If dia2(0) = 1 And dia2(1) = 0 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 1 And dia2(5) = 1 And dia2(6) = 1 And dia2(7) = 1 Then
        '6
        currdata.dia = currdata.dia + 60
    End If
    If dia2(0) = 1 And dia2(1) = 1 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 0 And dia2(5) = 0 And dia2(6) = 0 And dia2(7) = 0 Then
        '7
        currdata.dia = currdata.dia + 70
    End If
    If dia2(0) = 1 And dia2(1) = 1 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 1 And dia2(5) = 1 And dia2(6) = 1 And dia2(7) = 1 Then
        '8
        currdata.dia = currdata.dia + 80
    End If
    If dia2(0) = 1 And dia2(1) = 1 And dia2(2) = 1 And dia2(3) = 0
And dia2(4) = 1 And dia2(5) = 1 And dia2(6) = 0 And dia2(7) = 1 Then
        '9
        currdata.dia = currdata.dia + 90
    End If

    ListBoxtest1.Items.Add(currdata.dia)
End Sub
Private Sub pulfig()

    Dim upina1 As Integer
    Dim upina2 As Integer
    Dim upina3 As Integer
    Dim upina4 As Integer
    Dim upinb1 As Integer
    Dim upinb2 As Integer
    Dim upinb3 As Integer
    Dim upinb4 As Integer
    Dim upinc1 As Integer
    Dim upinc2 As Integer
    Dim upinc3 As Integer
    Dim upinc4 As Integer
    Dim upind1 As Integer
    Dim upind2 As Integer
    Dim upind3 As Integer
    Dim upind4 As Integer
    Dim upine1 As Integer
    Dim upine2 As Integer
    Dim upine3 As Integer
    Dim upine4 As Integer

    Dim lpina1 As Integer
    Dim lpina2 As Integer
    Dim lpina3 As Integer
```

```
Dim lpina4 As Integer
Dim lpinb1 As Integer
Dim lpinb2 As Integer
Dim lpinb3 As Integer
Dim lpinb4 As Integer
Dim lpinc1 As Integer
Dim lpinc2 As Integer
Dim lpinc3 As Integer
Dim lpinc4 As Integer
Dim lpind1 As Integer
Dim lpind2 As Integer
Dim lpind3 As Integer
Dim lpind4 As Integer
Dim lpine1 As Integer
Dim lpine2 As Integer
Dim lpine3 As Integer
Dim lpine4 As Integer

Dim i As Integer = 12
Dim j As Integer = 13
Dim k As Integer = 14
Dim l As Integer = 15
Dim n As Integer = 16

upina1 = arr(0, i) / 10000000
upina2 = arr(0, i) / 1000000
upina3 = arr(0, i) / 100000
upina4 = arr(0, i) / 10000
lpina1 = arr(2, i) / 1000
lpina2 = arr(2, i) / 100
lpina3 = arr(2, i) / 10
lpina4 = arr(2, i)

upinb1 = arr(0, j) / 10000000
upinb2 = arr(0, j) / 1000000
upinb3 = arr(0, j) / 100000
upinb4 = arr(0, j) / 10000
lpinb1 = arr(2, j) / 1000
lpinb2 = arr(2, j) / 100
lpinb3 = arr(2, j) / 10
lpinb4 = arr(2, j)

upinc1 = arr(0, k) / 10000000
upinc2 = arr(0, k) / 1000000
upinc3 = arr(0, k) / 100000
upinc4 = arr(0, k) / 10000
lpinc1 = arr(2, k) / 1000
lpinc2 = arr(2, k) / 100
lpinc3 = arr(2, k) / 10
lpinc4 = arr(2, k)

upind1 = arr(0, l) / 10000000
upind2 = arr(0, l) / 1000000
upind3 = arr(0, l) / 100000
upind4 = arr(0, l) / 10000
lpind1 = arr(2, l) / 1000
lpind2 = arr(2, l) / 100
```

```

lpind3 = arr(2, 1) / 10
lpind4 = arr(2, 1)

upine1 = arr(0, n) / 10000000
upine2 = arr(0, n) / 1000000
upine3 = arr(0, n) / 100000
upine4 = arr(0, n) / 10000
lpine1 = arr(2, n) / 1000
lpine2 = arr(2, n) / 100
lpine3 = arr(2, n) / 10
lpine4 = arr(2, n)

'for pin 12 for third digit
If upina1 = 1 And lpina1 = 1 Then
    ListBoxtest1.Items.Add("1a")
    pull(0) = 1
End If
If upina2 = 0 Or lpina2 = 0 Then
ElseIf upina2 = 1 Or upina2 = 11 And lpina2 = 1 Or lpina2 = 11
Then
    ListBoxtest1.Items.Add("1b")
    pull(1) = 1
End If

If upina3 = 0 Or lpina3 = 0 Then
ElseIf upina3 = 1 Or upina3 = 11 Or upina3 = 111 Or upina3 =
101 And lpina3 = 1 Or lpina3 = 11 Or lpina3 = 111 Or lpina3 = 101 Then
    pull(2) = 1
    ListBoxtest1.Items.Add("1c")
End If

If upina4 = 0 Or lpina4 = 0 Then
ElseIf upina4 = 1 Or upina4 = 11 Or upina4 = 111 Or upina4 =
101 Or upina4 = 1001 Or upina4 = 1011 Or upina4 = 1111 Or upina4 = 1101
And lpina4 = 1 Or lpina4 = 11 Or lpina4 = 111 Or lpina4 = 101 Or lpina4 =
= 1001 Or lpina4 = 1011 Or lpina4 = 1111 Or lpina4 = 1101 Then
    pull(3) = 1
    ListBoxtest1.Items.Add("1d")
End If
'for pin 13 for third digit
If upinb1 = 1 And lpinb1 = 1 Then
    ListBoxtest1.Items.Add("1e")
    pull(4) = 1
End If
If upinb2 = 0 Or lpinb2 = 0 Then
ElseIf upinb2 = 1 Or upinb2 = 11 And lpinb2 = 1 Or lpinb2 = 11
Then
    ListBoxtest1.Items.Add("1f")
    pull(5) = 1
End If
If upinb3 = 0 Or lpinb3 = 0 Then
ElseIf upinb3 = 1 Or upinb3 = 11 Or upinb3 = 111 Or upinb3 =
101 And lpinb3 = 1 Or lpinb3 = 11 Or lpinb3 = 111 Or lpinb3 = 101 Then
    pull(6) = 1
    ListBoxtest1.Items.Add("1g")
End If
If upinb4 = 0 Or lpinb4 = 0 Then

```

```

        ElseIf upinb4 = 1 Or upinb4 = 11 Or upinb4 = 111 Or upinb4 =
101 Or upinb4 = 1001 Or upinb4 = 1011 Or upinb4 = 1111 Or upinb4 = 1101
And lpinb4 = 1 Or lpinb4 = 11 Or lpinb4 = 111 Or lpinb4 = 101 Or lpinb4
= 1001 Or lpinb4 = 1011 Or lpinb4 = 1111 Or lpinb4 = 1101 Then
            pul1(7) = 1
            ListBoxtest1.Items.Add("1h")
        End If
        'for pin 14 for third digit
        If upinc1 = 1 And lpincl = 1 Then
            ListBoxtest1.Items.Add("2a")
            pul2(0) = 1
        End If
        If upinc2 = 0 Or lpinc2 = 0 Then
        ElseIf upinc2 = 1 Or upinc2 = 11 And lpinc2 = 1 Or lpinc2 = 11
Then
            ListBoxtest1.Items.Add("2b")
            pul2(1) = 1
        End If
        If upinc3 = 0 Or lpinc3 = 0 Then
        ElseIf upinc3 = 1 Or upinc3 = 11 Or upinc3 = 111 Or upinc3 =
101 And lpinc3 = 1 Or lpinc3 = 11 Or lpinc3 = 111 Or lpinc3 = 101 Then
            pul2(2) = 1
            ListBoxtest1.Items.Add("2c")
        End If
        If upinc4 = 0 Or lpinc4 = 0 Then
        ElseIf upinc4 = 1 Or upinc4 = 11 Or upinc4 = 111 Or upinc4 =
101 Or upinc4 = 1001 Or upinc4 = 1011 Or upinc4 = 1111 Or upinc4 = 1101
And lpinc4 = 1 Or lpinc4 = 11 Or lpinc4 = 111 Or lpinc4 = 101 Or lpinc4
= 1001 Or lpinc4 = 1011 Or lpinc4 = 1111 Or lpinc4 = 1101 Then
            pul2(3) = 1
            ListBoxtest1.Items.Add("2d")
        End If
        'for pin 15 for third digit
        If upind1 = 1 And lpind1 = 1 Then
            ListBoxtest1.Items.Add("2e")
            pul2(4) = 1
        End If
        If upind2 = 0 Or lpind2 = 0 Then
        ElseIf upind2 = 1 Or upind2 = 11 And lpind2 = 1 Or lpind2 = 11
Then
            ListBoxtest1.Items.Add("2f")
            pul2(5) = 1
        End If
        If upind3 = 0 Or lpind3 = 0 Then
        ElseIf upind3 = 1 Or upind3 = 11 Or upind3 = 111 Or upind3 =
101 And lpind3 = 1 Or lpind3 = 11 Or lpind3 = 111 Or lpind3 = 101 Then
            pul2(6) = 1
            ListBoxtest1.Items.Add("2g")
        End If
        If upind4 = 0 Or lpind4 = 0 Then
        ElseIf upind4 = 1 Or upind4 = 11 Or upind4 = 111 Or upind4 =
101 Or upind4 = 1001 Or upind4 = 1011 Or upind4 = 1111 Or upind4 = 1101
And lpind4 = 1 Or lpind4 = 11 Or lpind4 = 111 Or lpind4 = 101 Or lpind4
= 1001 Or lpind4 = 1011 Or lpind4 = 1111 Or lpind4 = 1101 Then
            pul2(7) = 1
            ListBoxtest1.Items.Add("2h")
        End If

```

```

'for pin 16 for third digit
If upine1 = 1 And lpine1 = 1 Then
    ListBoxtest1.Items.Add("3a")
    pul3(0) = 1
End If
If upine2 = 0 Or lpine2 = 0 Then
ElseIf upine2 = 1 Or upine2 = 11 And lpine2 = 1 Or lpine2 = 11
Then
    ListBoxtest1.Items.Add("3b")
    pul3(1) = 1
End If

If upine3 = 0 Or lpine3 = 0 Then
ElseIf upine3 = 1 Or upine3 = 11 Or upine3 = 111 Or upine3 =
101 And lpine3 = 1 Or lpine3 = 11 Or lpine3 = 111 Or lpine3 = 101 Then
    pul3(2) = 1
    ListBoxtest1.Items.Add("3c")
End If

If upine4 = 0 Or lpine4 = 0 Then
ElseIf upine4 = 1 Or upine4 = 11 Or upine4 = 111 Or upine4 =
101 Or upine4 = 1001 Or upine4 = 1011 Or upine4 = 1111 Or upine4 = 1101
And lpine4 = 1 Or lpine4 = 11 Or lpine4 = 111 Or lpine4 = 101 Or lpine4
= 1001 Or lpine4 = 1011 Or lpine4 = 1111 Or lpine4 = 1101 Then
    pul3(3) = 1
    ListBoxtest1.Items.Add("3d")
End If
End Sub

Private Sub pulnum()
    Dim currdata As New userdata1
    currdata.pulse = 0
    'for pin 12 and 13 for third digit
    If pull(0) = 1 And pull(1) = 1 And pull(2) = 1 And pull(3) = 0
And pull(4) = 1 And pull(5) = 0 And pull(6) = 1 And pull(7) = 1 Then
        '0
        currdata.pulse = currdata.pulse + 0
    End If
    If pull(0) = 0 And pull(1) = 1 And pull(2) = 1 And pull(3) = 0
And pull(4) = 0 And pull(5) = 0 And pull(6) = 0 And pull(7) = 0 Then
        '1
        currdata.pulse = currdata.pulse + 1
    End If
    If pull(0) = 1 And pull(1) = 1 And pull(2) = 0 And pull(3) = 0
And pull(4) = 0 And pull(5) = 1 And pull(6) = 1 And pull(7) = 1 Then
        '2
        currdata.pulse = currdata.pulse + 2
    End If
    If pull(0) = 1 And pull(1) = 1 And pull(2) = 1 And pull(3) = 0
And pull(4) = 0 And pull(5) = 1 And pull(6) = 0 And pull(7) = 1 Then
        '3
        currdata.pulse = currdata.pulse + 3
    End If
    If pull(0) = 0 And pull(1) = 1 And pull(2) = 1 And pull(3) = 0
And pull(4) = 1 And pull(5) = 1 And pull(6) = 0 And pull(7) = 0 Then
        '4
        currdata.pulse = currdata.pulse + 4
    End If
End Sub

```

```
End If
If pull1(0) = 1 And pull1(1) = 0 And pull1(2) = 1 And pull1(3) = 0
And pull1(4) = 1 And pull1(5) = 1 And pull1(6) = 0 And pull1(7) = 1 Then
'5
currdata.pulse = currdata.pulse + 5
End If
If pull1(0) = 1 And pull1(1) = 0 And pull1(2) = 1 And pull1(3) = 0
And pull1(4) = 1 And pull1(5) = 1 And pull1(6) = 1 And pull1(7) = 1 Then
'6
currdata.pulse = currdata.pulse + 6
End If
If pull1(0) = 1 And pull1(1) = 1 And pull1(2) = 1 And pull1(3) = 0
And pull1(4) = 0 And pull1(5) = 0 And pull1(6) = 0 And pull1(7) = 0 Then
'7
currdata.pulse = currdata.pulse + 7
End If
If pull1(0) = 1 And pull1(1) = 1 And pull1(2) = 1 And pull1(3) = 0
And pull1(4) = 1 And pull1(5) = 1 And pull1(6) = 1 And pull1(7) = 1 Then
'8
currdata.pulse = currdata.pulse + 8
End If
If pull1(0) = 1 And pull1(1) = 1 And pull1(2) = 1 And pull1(3) = 0
And pull1(4) = 1 And pull1(5) = 1 And pull1(6) = 0 And pull1(7) = 1 Then
'9
currdata.pulse = currdata.pulse + 9
End If

'for pin 14 and 15 for second digit
If pul2(0) = 1 And pul2(1) = 1 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 1 And pul2(5) = 0 And pul2(6) = 1 And pul2(7) = 1 Then
'0
currdata.pulse = currdata.pulse + 0
End If
If pul2(0) = 0 And pul2(1) = 1 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 0 And pul2(5) = 0 And pul2(6) = 0 And pul2(7) = 0 Then
'1
currdata.pulse = currdata.pulse + 10
End If
If pul2(0) = 1 And pul2(1) = 1 And pul2(2) = 0 And pul2(3) = 0
And pul2(4) = 0 And pul2(5) = 1 And pul2(6) = 1 And pul2(7) = 1 Then
'2
currdata.pulse = currdata.pulse + 20
End If
If pul2(0) = 1 And pul2(1) = 1 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 0 And pul2(5) = 1 And pul2(6) = 0 And pul2(7) = 1 Then
'3
currdata.pulse = currdata.pulse + 30
End If
If pul2(0) = 0 And pul2(1) = 1 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 1 And pul2(5) = 1 And pul2(6) = 0 And pul2(7) = 0 Then
'4
currdata.pulse = currdata.pulse + 40
End If
If pul2(0) = 1 And pul2(1) = 0 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 1 And pul2(5) = 1 And pul2(6) = 0 And pul2(7) = 1 Then
'5
currdata.pulse = currdata.pulse + 50
```

```

        End If
        If pul2(0) = 1 And pul2(1) = 0 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 1 And pul2(5) = 1 And pul2(6) = 1 And pul2(7) = 1 Then
            '6
            currdata.pulse = currdata.pulse + 60
        End If
        If pul2(0) = 1 And pul2(1) = 1 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 0 And pul2(5) = 0 And pul2(6) = 0 And pul2(7) = 0 Then
            '7
            currdata.pulse = currdata.pulse + 70
        End If
        If pul2(0) = 1 And pul2(1) = 1 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 1 And pul2(5) = 1 And pul2(6) = 1 And pul2(7) = 1 Then
            '8
            currdata.pulse = currdata.pulse + 80
        End If
        If pul2(0) = 1 And pul2(1) = 1 And pul2(2) = 1 And pul2(3) = 0
And pul2(4) = 1 And pul2(5) = 1 And pul2(6) = 0 And pul2(7) = 1 Then
            '9
            currdata.pulse = currdata.pulse + 90
        End If

        'for pin 16 for first digit
        If pul3(1) = 1 And pul3(2) = 1 Then
            '1
            currdata.pulse = currdata.pulse + 100
        End If

        ListBoxtest1.Items.Add(currdata.pulse)
    End Sub
    ' initialise for com port

    Private Sub btnCheckForPorts_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCheckForPorts.Click
        ' Check for Availability of each of the 4 Comm Ports, and
        ' place a check in the list box items that have openable
ports.
        Dim i As Integer
        For i = 1 To 4
            'WriteMessage("Testing COM" + i.ToString())
            If IsPortAvailable(i) Then
                ' Check the box for available ports.
                Me.clstPorts.SetItemChecked(i - 1, True)
            Else
                ' Uncheck the box for unavailable ports.
                Me.clstPorts.SetItemChecked(i - 1, False)
            End If
        Next

    End Sub
    'Private Sub WriteMessage(ByVal message As String)
    ' Me.txtStatus.Text += message + vbCrLf
    'End Sub
    Private m_CommPort As New Rs232
    Private m_ModemPort As Integer = 0

    ' This function attempts to open the passed Comm Port. If it is

```

```

' available, it returns True, else it returns False. To determine
' availability a Try-Catch block is used.
Private Function IsPortAvailable(ByVal ComPort As Integer) As
Boolean
    Try
        m_CommPort.Open(ComPort, 9600, 8,
Rs232.DataParity.Parity_None, Rs232.DataStopBit.StopBit_1, 4096)
        ' If it makes it to here, then the Comm Port is available.
        m_CommPort.Close()
        Return True
    Catch
        ' If it gets here, then the attempt to open the Comm Port
was unsuccessful.
        Return False
    End Try
End Function
Private Sub btnSendUserCommand_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)

    ' Always wrap up working with Comm Ports in exception handlers.
    Try
        ' Enable the timer.
        tmrReadCommPort.Enabled = True
        ' Attempt to open the port.
        m_CommPort.Open(m_ModemPort, 115200, 8,
Rs232.DataParity.Parity_None, Rs232.DataStopBit.StopBit_1, 4096)

        ' Write an user specified Command to the Port.
        'm_CommPort.Write(Me.txtUserCommand.Text & Chr(13))
        ' Sleep long enough for the modem to respond and the timer
to fire.
        System.Threading.Thread.Sleep(200)
        Application.DoEvents()
        m_CommPort.Close()

    Catch ex As Exception
        ' Warn the user.
        MessageBox.Show("Unable to communicate with Modem")
    Finally
        ' Disable the timer.
        Me.tmrReadCommPort.Enabled = False
    End Try

End Sub

Private Sub btnprintsetup_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnprintsetup.Click
    PageSetupDialog1.ShowDialog()
End Sub

Private Sub btnprinthealth_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnprinthealth.Click
    'mintSeatsCtr = 0
    PrintPreviewDialog1.ShowDialog()
End Sub

```



```
Private Sub PrintPreviewDialog1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PrintPreviewDialog1.Load

    End Sub

Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object,
ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles
PrintDocument1.PrintPage
    Dim i As Integer
    Dim j As Integer
    j = ListView1.Items.Count
    For i = 0 To j
        e.Graphics.DrawString(ListView1.Items(i).Text, New
Font("Courier New", 12, FontStyle.Bold), Brushes.Black, 150, 50 + i *
17)
        e.Graphics.DrawString(ListView1.Items(i).SubItems(1).Text,
New Font("Courier New", 12, FontStyle.Bold), Brushes.Black, 150, 50 + i
* 17)
        e.Graphics.DrawString(ListView1.Items(i).SubItems(2).Text,
New Font("Courier New", 12, FontStyle.Bold), Brushes.Black, 150, 50 + i
* 17)
        e.Graphics.DrawString(ListView1.Items(i).SubItems(3).Text,
New Font("Courier New", 12, FontStyle.Bold), Brushes.Black, 150, 50 + i
* 17)
        e.Graphics.DrawString(ListView1.Items(i).SubItems(4).Text,
New Font("Courier New", 12, FontStyle.Bold), Brushes.Black, 150, 50 + i
* 17)
        e.Graphics.DrawString(ListView1.Items(i).SubItems(5).Text,
New Font("Courier New", 12, FontStyle.Bold), Brushes.Black, 150, 50 + i
* 17)
        'mintSeatsCtr += 1
        'If mintSeatsCtr = j Then Exit For
    Next
    If i < j Then
        e.HasMorePages = True
    Else
        e.HasMorePages = False
    End If

End Sub
End Class
```

E2 Project Module.vb

```
Module Project_Module
    Sub Main()
        Dim objfrmnewuser1 As New frmnewuser
        Dim objfrmstartpage As New frmstartpage
        objfrmstartpage.ShowDialog()
        objfrmnewuser1.ShowDialog()
    End Sub
End Module
```

E3 StartPage.vb

```
Public Class frmstartpage
    Inherits System.Windows.Forms.Form

    Private Sub Timer1_Elapsed(ByVal sender As System.Object, ByVal e As
System.Timers.ElapsedEventArgs) Handles Timer1.Elapsed
        Me.Close()

    End Sub
End Class
```

E4 Rs232.vb

```
Option Strict On

Imports System.Runtime.InteropServices
Imports System.Text
Imports System.Threading

' This class provides all the necessary support for communicating
' with the Comm Port (otherwise known as the Serial Port, or
' RS232 port).
Public Class Rs232
    ' Declare the necessary class variables, and their initial values.

    Private mhRS As Integer = -1 ' Handle to Com Port

    Private miPort As Integer = 1 ' Default is COM1
    Private miTimeout As Integer = 70 ' Timeout in ms
    Private miBaudRate As Integer = 9600
    Private meParity As DataParity = 0
    Private meStopBit As DataStopBit = 0
    Private miDataBit As Integer = 8
    Private miBufferSize As Integer = 512 ' Buffers size default to
512 bytes
    Private mabtRxBuf As Byte() ' Receive buffer
    Private meMode As Mode ' Class working mode
    Private mbWaitOnRead As Boolean
    Private mbWaitOnWrite As Boolean
    Private mbWriteErr As Boolean
```

```

Private muOverlapped As OVERLAPPED
Private muOverlappedW As OVERLAPPED
Private muOverlappedE As OVERLAPPED
Private mabtTmpTxBuf As Byte() ' Temporary buffer used by Async Tx
Private moThreadTx As Thread
Private moThreadRx As Thread
Private miTmpBytes2Read As Integer
Private meMask As EventMasks

#Region "Enums"

' This enumeration provides Data Parity values.
Public Enum DataParity
    Parity_None = 0
    Parity_Odd
    Parity_Even
    Parity_Mark
End Enum

' This enumeration provides Data Stop Bit values.
' It is set to begin with a one, so that the enumeration values
' match the actual values.
Public Enum DataStopBit
    StopBit_1 = 1
    StopBit_2
End Enum

' This enumeration contains values used to purge the various
buffers.
Private Enum PurgeBuffers
    RXAbort = &H2
    RXClear = &H8
    TxAbsort = &H1
    TxClear = &H4
End Enum

' This enumeration provides values for the lines sent to the Comm
Port
Private Enum Lines
    SetRts = 3
    ClearRts = 4
    SetDtr = 5
    ClearDtr = 6
    ResetDev = 7 ' Reset device if possible
    SetBreak = 8 ' Set the device break line.
    ClearBreak = 9 ' Clear the device break line.
End Enum

' This enumeration provides values for the Modem Status, since
' we'll be communicating primarily with a modem.
' Note that the Flags() attribute is set to allow for a bitwise
' combination of values.
<Flags(> Public Enum ModemStatusBits
    ClearToSendOn = &H10
    DataSetReadyOn = &H20
    RingIndicatorOn = &H40
    CarrierDetect = &H80
End Enum

```

```

' This enumeration provides values for the Working mode
Public Enum Mode
    NonOverlapped
    Overlapped
End Enum

' This enumeration provides values for the Comm Masks used.
' Note that the Flags() attribute is set to allow for a bitwise
' combination of values.
<Flags(> Public Enum EventMasks
    RxChar = &H1
    RXFlag = &H2
    TxBufferEmpty = &H4
    ClearToSend = &H8
    DataSetReady = &H10
    ReceiveLine = &H20
    Break = &H40
    StatusError = &H80
    Ring = &H100
End Enum
#End Region

#Region "Structures"
' This is the DCB structure used by the calls to the Windows API.
<StructLayout(LayoutKind.Sequential, Pack:=1)> Private Structure
DCB
    Public DCBlength As Integer
    Public BaudRate As Integer
    Public Bits1 As Integer
    Public wReserved As Int16
    Public XonLim As Int16
    Public XoffLim As Int16
    Public ByteSize As Byte
    Public Parity As Byte
    Public StopBits As Byte
    Public XonChar As Byte
    Public XoffChar As Byte
    Public ErrorChar As Byte
    Public EofChar As Byte
    Public EvtChar As Byte
    Public wReserved2 As Int16
End Structure

' This is the CommTimeOuts structure used by the calls to the
Windows API.
<StructLayout(LayoutKind.Sequential, Pack:=1)> Private Structure
COMMTIMEOUTS
    Public ReadIntervalTimeout As Integer
    Public ReadTotalTimeoutMultiplier As Integer
    Public ReadTotalTimeoutConstant As Integer
    Public WriteTotalTimeoutMultiplier As Integer
    Public WriteTotalTimeoutConstant As Integer
End Structure

' This is the CommConfig structure used by the calls to the Windows
API.

```

```

    <StructLayout(LayoutKind.Sequential, Pack:=1)> Private Structure
COMMCONFIG
    Public dwSize As Integer
    Public wVersion As Int16
    Public wReserved As Int16
    Public dcbx As DCB
    Public dwProviderSubType As Integer
    Public dwProviderOffset As Integer
    Public dwProviderSize As Integer
    Public wcProviderData As Byte
End Structure

' This is the OverLapped structure used by the calls to the Windows
API.
    <StructLayout(LayoutKind.Sequential, Pack:=1)> Public Structure
OVERLAPPED
    Public Internal As Integer
    Public InternalHigh As Integer
    Public Offset As Integer
    Public OffsetHigh As Integer
    Public hEvent As Integer
End Structure
#End Region

#Region "Exceptions"

' This class defines a customized channel exception. This exception
is
'   raised when a NACK is raised.
Public Class CIOChannelException : Inherits ApplicationException
    Sub New(ByVal Message As String)
        MyBase.New(Message)
    End Sub
    Sub New(ByVal Message As String, ByVal InnerException As
Exception)
        MyBase.New(Message, InnerException)
    End Sub
End Class

' This class defines a customized timeout exception.
Public Class IOTimeoutException : Inherits CIOChannelException
    Sub New(ByVal Message As String)
        MyBase.New(Message)
    End Sub
    Sub New(ByVal Message As String, ByVal InnerException As
Exception)
        MyBase.New(Message, InnerException)
    End Sub
End Class

#End Region

#Region "Events"
' These events allow the program using this class to react to Comm
Port
'   events.

```

```
Public Event DataReceived(ByVal Source As Rs232, ByVal DataBuffer()  
As Byte)  
Public Event TxCompleted(ByVal Source As Rs232)  
Public Event CommEvent(ByVal Source As Rs232, ByVal Mask As  
EventMasks)  
#End Region  
  
#Region "Constants"  
' These constants are used to make the code clearer.  
Private Const PURGE_RXABORT As Integer = &H2  
Private Const PURGE_RXCLEAR As Integer = &H8  
Private Const PURGE_TXABORT As Integer = &H1  
Private Const PURGE_TXCLEAR As Integer = &H4  
Private Const GENERIC_READ As Integer = &H80000000  
Private Const GENERIC_WRITE As Integer = &H40000000  
Private Const OPEN_EXISTING As Integer = 3  
Private Const INVALID_HANDLE_VALUE As Integer = -1  
Private Const IO_BUFFER_SIZE As Integer = 1024  
Private Const FILE_FLAG_OVERLAPPED As Integer = &H40000000  
Private Const ERROR_IO_PENDING As Integer = 997  
Private Const WAIT_OBJECT_0 As Integer = 0  
Private Const ERROR_IO_INCOMPLETE As Integer = 996  
Private Const WAIT_TIMEOUT As Integer = &H102&  
Private Const INFINITE As Integer = &HFFFFFFFF  
  
#End Region  
  
#Region "Properties"  
  
' This property gets or sets the BaudRate  
Public Property BaudRate() As Integer  
Get  
Return miBaudRate  
End Get  
Set(ByVal Value As Integer)  
miBaudRate = Value  
End Set  
End Property  
  
' This property gets or sets the BufferSize  
Public Property BufferSize() As Integer  
Get  
Return miBufferSize  
End Get  
Set(ByVal Value As Integer)  
miBufferSize = Value  
End Set  
End Property  
  
' This property gets or sets the DataBit.  
Public Property DataBit() As Integer  
Get  
Return miDataBit  
End Get  
Set(ByVal Value As Integer)  
miDataBit = Value
```

```

    End Set
End Property

' This write-only property sets or resets the DTR line.
Public WriteOnly Property Dtr() As Boolean
    Set(ByVal Value As Boolean)
        If Not mhRS = -1 Then
            If Value Then
                EscapeCommFunction(mhRS, Lines.SetDtr)
            Else
                EscapeCommFunction(mhRS, Lines.ClearDtr)
            End If
        End If
    End Set
End Property

' This read-only property returns an array of bytes that represents
' the input coming into the Comm Port.
Overridable ReadOnly Property InputStream() As Byte()
    Get
        Return mabtRxBuf
    End Get
End Property

' This read-only property returns a string that represents
' the data coming into to the Comm Port.
Overridable ReadOnly Property InputStreamString() As String
    Get
        Dim oEncoder As New System.Text.ASCIIEncoding
        Return oEncoder.GetString(Me.InputStream)
    End Get
End Property

' This property returns the open status of the Comm Port.
ReadOnly Property IsOpen() As Boolean
    Get
        Return CBool(mhRS <> -1)
    End Get
End Property

' This read-only property returns the status of the modem.
Public ReadOnly Property ModemStatus() As ModemStatusBits
    Get
        If mhRS = -1 Then
            Throw New ApplicationException("Please initialize and
open " + _
                "port before using this method")
        Else
            ' Retrieve modem status
            Dim lpModemStatus As Integer
            If Not GetCommModemStatus(mhRS, lpModemStatus) Then
                Throw New ApplicationException("Unable to get modem
status")
            Else
                Return CType(lpModemStatus, ModemStatusBits)
            End If
        End If
    End Get
End Property

```

```
        End Get
    End Property

    ' This property gets or sets the Parity
    Public Property Parity() As DataParity
        Get
            Return meParity
        End Get
        Set(ByVal Value As DataParity)
            meParity = Value
        End Set
    End Property

    ' This property gets or sets the Port
    Public Property Port() As Integer
        Get
            Return miPort
        End Get
        Set(ByVal Value As Integer)
            miPort = Value
        End Set
    End Property

    ' This write-only property sets or resets the RTS line.
    Public WriteOnly Property Rts() As Boolean
        Set(ByVal Value As Boolean)
            If Not mhRS = -1 Then
                If Value Then
                    EscapeCommFunction(mhRS, Lines.SetRts)
                Else
                    EscapeCommFunction(mhRS, Lines.ClearRts)
                End If
            End If
        End Set
    End Property

    ' This property gets or sets the StopBit
    Public Property StopBit() As DataStopBit
        Get
            Return meStopBit
        End Get
        Set(ByVal Value As DataStopBit)
            meStopBit = Value
        End Set
    End Property

    ' This property gets or sets the Timeout
    Public Overridable Property Timeout() As Integer
        Get
            Return miTimeout
        End Get
        Set(ByVal Value As Integer)
            miTimeout = CInt(IIf(Value = 0, 500, Value))
            ' If Port is open updates it on the fly
            pSetTimeout()
        End Set
    End Property
```



```

' This property gets or sets the working mode to overlapped
' or non-overlapped.
Public Property WorkingMode() As Mode
    Get
        Return meMode
    End Get
    Set(ByVal Value As Mode)
        meMode = Value
    End Set
End Property

#End Region

#Region "Win32API"
' The following functions are the required Win32 functions needed
to
' make communication with the Comm Port possible.

<DllImport("kernel32.dll")> Private Shared Function BuildCommDCB( _
    ByVal lpDef As String, ByRef lpDCB As DCB) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function
ClearCommError( _
    ByVal hFile As Integer, ByVal lpErrors As Integer, _
    ByVal l As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function CloseHandle( _
    ByVal hObject As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function CreateEvent( _
    ByVal lpEventAttributes As Integer, ByVal bManualReset As
Integer, _
    ByVal bInitialState As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal lpName As String) As
Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function CreateFile( _
    <MarshalAs(UnmanagedType.LPStr)> ByVal lpFileName As String, _
    ByVal dwDesiredAccess As Integer, ByVal dwShareMode As Integer,
_
    ByVal lpSecurityAttributes As Integer, _
    ByVal dwCreationDisposition As Integer, _
    ByVal dwFlagsAndAttributes As Integer, _
    ByVal hTemplateFile As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function
EscapeCommFunction( _
    ByVal hFile As Integer, ByVal ifunc As Long) As Boolean
End Function

```

```

    <DllImport("kernel32.dll")> Private Shared Function
FormatMessage( _
    ByVal dwFlags As Integer, ByVal lpSource As Integer, _
    ByVal dwMessageId As Integer, ByVal dwLanguageId As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal lpBuffer As String, _
    ByVal nSize As Integer, ByVal Arguments As Integer) As Integer
End Function

Private Declare Function FormatMessage Lib "kernel32" Alias _
"FormatMessageA" (ByVal dwFlags As Integer, ByVal lpSource As
Integer, _
    ByVal dwMessageId As Integer, ByVal dwLanguageId As Integer, _
    ByVal lpBuffer As StringBuilder, ByVal nSize As Integer, _
    ByVal Arguments As Integer) As Integer

    <DllImport("kernel32.dll")> Public Shared Function
GetCommModemStatus( _
    ByVal hFile As Integer, ByRef lpModemStatus As Integer) As
Boolean
End Function

    <DllImport("kernel32.dll")> Private Shared Function GetCommState( _
    ByVal hCommDev As Integer, ByRef lpDCB As DCB) As Integer
End Function

    <DllImport("kernel32.dll")> Private Shared Function
GetCommTimeouts( _
    ByVal hFile As Integer, ByRef lpCommTimeouts As COMMTIMEOUTS)
As Integer
End Function

    <DllImport("kernel32.dll")> Private Shared Function GetLastError()
As Integer
End Function

    <DllImport("kernel32.dll")> Private Shared Function
GetOverlappedResult( _
    ByVal hFile As Integer, ByRef lpOverlapped As OVERLAPPED, _
    ByRef lpNumberOfBytesTransferred As Integer, _
    ByVal bWait As Integer) As Integer
End Function

    <DllImport("kernel32.dll")> Private Shared Function PurgeComm( _
    ByVal hFile As Integer, ByVal dwFlags As Integer) As Integer
End Function

    <DllImport("kernel32.dll")> Private Shared Function ReadFile( _
    ByVal hFile As Integer, ByVal Buffer As Byte(), _
    ByVal nNumberOfBytesToRead As Integer, _
    ByRef lpNumberOfBytesRead As Integer, _
    ByRef lpOverlapped As OVERLAPPED) As Integer
End Function

    <DllImport("kernel32.dll")> Private Shared Function
SetCommTimeouts( _
    ByVal hFile As Integer, ByRef lpCommTimeouts As COMMTIMEOUTS)
As Integer

```

```

End Function

<DllImport("kernel32.dll")> Private Shared Function SetCommState( _
    ByVal hCommDev As Integer, ByRef lpDCB As DCB) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function SetupComm( _
    ByVal hFile As Integer, ByVal dwInQueue As Integer, _
    ByVal dwOutQueue As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function SetCommMask( _
    ByVal hFile As Integer, ByVal lpEvtMask As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function
WaitCommEvent( _
    ByVal hFile As Integer, ByRef Mask As EventMasks, _
    ByRef lpOverlap As OVERLAPPED) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function
WaitForSingleObject( _
    ByVal hHandle As Integer, ByVal dwMilliseconds As Integer) As
Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function WriteFile( _
    ByVal hFile As Integer, ByVal Buffer As Byte(), _
    ByVal nNumberOfBytesToWrite As Integer, _
    ByRef lpNumberOfBytesWritten As Integer, _
    ByRef lpOverlapped As OVERLAPPED) As Integer
End Function

#End Region

#Region "Methods"

' This subroutine invokes a thread to perform an asynchronous read.
' This routine should not be called directly, but is used
' by the class.
Public Sub _R()
    Dim iRet As Integer = Read(miTmpBytes2Read)
End Sub

' This subroutine invokes a thread to perform an asynchronous write.
' This routine should not be called directly, but is used
' by the class.
Public Sub _W()
    Write(mabtTmpTxBuf)
End Sub

' This subroutine uses another thread to read from the Comm Port.
It
' raises RxCompleted when done. It reads an integer.
Public Overloads Sub AsyncRead(ByVal Bytes2Read As Integer)

```

```

        If meMode <> Mode.Overlapped Then Throw New
ApplicationException( _
    "Async Methods allowed only when WorkingMode=Overlapped")
    miTmpBytes2Read = Bytes2Read
    moThreadTx = New Thread(AddressOf _R)
    moThreadTx.Start()
End Sub

' This subroutine uses another thread to write to the Comm Port. It
' raises TxCompleted when done. It writes an array of bytes.
Public Overloads Sub AsyncWrite(ByVal Buffer() As Byte)
    If meMode <> Mode.Overlapped Then Throw New
ApplicationException( _
    "Async Methods allowed only when WorkingMode=Overlapped")
    If mbWaitOnWrite = True Then Throw New ApplicationException( _
    "Unable to send message because of pending transmission.")
    mabtTmpTxBuf = Buffer
    moThreadTx = New Thread(AddressOf _W)
    moThreadTx.Start()
End Sub

' This subroutine uses another thread to write to the Comm Port. It
' raises TxCompleted when done. It writes a string.
Public Overloads Sub AsyncWrite(ByVal Buffer As String)
    Dim oEncoder As New System.Text.ASCIIEncoding
    Dim aByte() As Byte = oEncoder.GetBytes(Buffer)
    Me.AsyncWrite(aByte)
End Sub

' This function takes the ModemStatusBits and returns a boolean
value
' signifying whether the Modem is active.
Public Function CheckLineStatus(ByVal Line As ModemStatusBits) As
Boolean
    Return Convert.ToBoolean(ModemStatus And Line)
End Function

' This subroutine clears the input buffer.
Public Sub ClearInputBuffer()
    If Not mhRS = -1 Then
        PurgeComm(mhRS, PURGE_RXCLEAR)
    End If
End Sub

' This subroutine closes the Comm Port.
Public Sub Close()
    If mhRS <> -1 Then
        CloseHandle(mhRS)
        mhRS = -1
    End If
End Sub

' This subroutine opens and initializes the Comm Port
Public Overloads Sub Open()
    ' Get Dcb block, Update with current data
    Dim uDcb As DCB, iRc As Integer
    ' Set working mode

```

```

    Dim iMode As Integer = Convert.ToInt32(IIf(meMode =
Mode.Overlapped, _
    FILE_FLAG_OVERLAPPED, 0))
' Initializes Com Port
If miPort > 0 Then
    Try
        ' Creates a COM Port stream handle
        mhRS = CreateFile("COM" & miPort.ToString, _
        GENERIC_READ Or GENERIC_WRITE, 0, 0, _
        OPEN_EXISTING, iMode, 0)
        If mhRS <> -1 Then
            ' Clear all communication errors
            Dim lpErrCode As Integer
            iRc = ClearCommError(mhRS, lpErrCode, 0&)
            ' Clears I/O buffers
            iRc = PurgeComm(mhRS, PurgeBuffers.RXClear Or _
            PurgeBuffers.TxClear)
            ' Gets COM Settings
            iRc = GetCommState(mhRS, uDcb)
            ' Updates COM Settings
            Dim sParity As String = "NOEM"
            sParity = sParity.Substring(meParity, 1)
            ' Set DCB State
            Dim sDCBState As String = String.Format( _
            "baud={0} parity={1} data={2} stop={3}", _
            miBaudRate, sParity, miDataBit, CInt(meStopBit))
            iRc = BuildCommDCB(sDCBState, uDcb)
            iRc = SetCommState(mhRS, uDcb)
            If iRc = 0 Then
                Dim sErrTxt As String =
pErr2Text(GetLastError())
                Throw New CIOChannelException( _
                "Unable to set COM state0" & sErrTxt)
            End If
            ' Setup Buffers (Rx,Tx)
            iRc = SetupComm(mhRS, miBufferSize, miBufferSize)
            ' Set Timeouts
            pSetTimeout()
        Else
            ' Raise Initialization problems
            Throw New CIOChannelException( _
            "Unable to open COM" & miPort.ToString)
        End If
    Catch Ex As Exception
        ' Generica error
        Throw New CIOChannelException(Ex.Message, Ex)
    End Try
Else
    ' Port not defined, cannot open
    Throw New ApplicationException("COM Port not defined, " + _
    "use Port property to set it before invoking InitPort")
End If
End Sub

' This subroutine opens and initializes the Comm Port (overloaded
' to support parameters).
Public Overloads Sub Open(ByVal Port As Integer, _

```

```

    ByVal BaudRate As Integer, ByVal DataBit As Integer, _
    ByVal Parity As DataParity, ByVal StopBit As DataStopBit, _
    ByVal BufferSize As Integer)

    Me.Port = Port
    Me.BaudRate = BaudRate
    Me.DataBit = DataBit
    Me.Parity = Parity
    Me.StopBit = StopBit
    Me.BufferSize = BufferSize
    Open()
End Sub

' This function translates an API error code to text.
Private Function pErr2Text(ByVal lCode As Integer) As String
    Dim sRtrnCode As New StringBuilder(256)
    Dim lRet As Integer

    lRet = FormatMessage(&H1000, 0, lCode, 0, sRtrnCode, 256, 0)
    If lRet > 0 Then
        Return sRtrnCode.ToString
    Else
        Return "Error not found."
    End If
End Function

' This subroutine handles overlapped reads.
Private Sub pHandleOverlappedRead(ByVal Bytes2Read As Integer)
    Dim iReadChars, iRc, iRes, iLastErr As Integer
    muOverlapped.hEvent = CreateEvent(Nothing, 1, 0, Nothing)
    If muOverlapped.hEvent = 0 Then
        ' Can't create event
        Throw New ApplicationException( _
            "Error creating event for overlapped read.")
    Else
        ' Ovellaped reading
        If mbWaitOnRead = False Then
            ReDim mabtRxBuf(Bytes2Read - 1)
            iRc = ReadFile(mhRS, mabtRxBuf, Bytes2Read, _
                iReadChars, muOverlapped)
            If iRc = 0 Then
                iLastErr = GetLastError()
                If iLastErr <> ERROR_IO_PENDING Then
                    Throw New ArgumentException("Overlapped Read
Error: " & _
                        pErr2Text(iLastErr))
                Else
                    ' Set Flag
                    mbWaitOnRead = True
                End If
            Else
                ' Read completed successfully
                RaiseEvent DataReceived(Me, mabtRxBuf)
            End If
        End If
    End If
End Sub

```

```

' Wait for operation to be completed
If mbWaitOnRead Then
    iRes = WaitForSingleObject(muOverlapped.hEvent, miTimeout)
    Select Case iRes
        Case WAIT_OBJECT_0
            ' Object signaled, operation completed
            If GetOverlappedResult(mhRS, muOverlapped, _
                iReadChars, 0) = 0 Then

                ' Operation error
                iLastError = GetLastError()
                If iLastError = ERROR_IO_INCOMPLETE Then
                    Throw New ApplicationException( _
                        "Read operation incomplete")
                Else
                    Throw New ApplicationException( _
                        "Read operation error " &
iLastError.ToString)
                End If
            Else
                ' Operation completed
                RaiseEvent DataReceived(Me, mabtRxBuf)
                mbWaitOnRead = False
            End If
        Case WAIT_TIMEOUT
            Throw New IOException("Timeout error")
        Case Else
            Throw New ApplicationException("Overlapped read
error")
    End Select
End If
End Sub

' This subroutine handles overlapped writes.
Private Function pHandleOverlappedWrite(ByVal Buffer() As Byte) As
Boolean
Dim iBytesWritten, iRc, iLastError, iRes As Integer, bErr As
Boolean
muOverlappedW.hEvent = CreateEvent(Nothing, 1, 0, Nothing)
If muOverlappedW.hEvent = 0 Then
    ' Can't create event
    Throw New ApplicationException( _
        "Error creating event for overlapped write.")
Else
    ' Overlapped write
    PurgeComm(mhRS, PURGE_RXCLEAR Or PURGE_TXCLEAR)
    mbWaitOnRead = True
    iRc = WriteFile(mhRS, Buffer, Buffer.Length, _
        iBytesWritten, muOverlappedW)
    If iRc = 0 Then
        iLastError = GetLastError()
        If iLastError <> ERROR_IO_PENDING Then
            Throw New ArgumentException("Overlapped Read Error:
" & _
                pErr2Text(iLastError))
        Else
            ' Write is pending

```

```

        iRes = WaitForSingleObject(muOverlappedW.hEvent,
INFINITE)
        Select Case iRes
            Case WAIT_OBJECT_0
                ' Object signaled, operation completed
                If GetOverlappedResult(mhRS, muOverlappedW,
-
                    iBytesWritten, 0) = 0 Then

                    bErr = True
                Else
                    ' Notifies Async tx completion, stops
thread
                    mbWaitOnRead = False
                    RaiseEvent TxCompleted(Me)
                End If
            End Select
        End If
    Else
        ' Wait operation completed immediatly
        bErr = False
    End If
End If
CloseHandle(muOverlappedW.hEvent)
Return bErr
End Function

' This subroutine sets the Comm Port timeouts.
Private Sub pSetTimeout()
    Dim uCtm As COMMTIMEOUTS
    ' Set ComTimeout
    If mhRS = -1 Then
        Exit Sub
    Else
        ' Changes setup on the fly
        With uCtm
            .ReadIntervalTimeout = 0
            .ReadTotalTimeoutMultiplier = 0
            .ReadTotalTimeoutConstant = miTimeout
            .WriteTotalTimeoutMultiplier = 10
            .WriteTotalTimeoutConstant = 100
        End With
        SetCommTimeouts(mhRS, uCtm)
    End If
End Sub

' This function returns an integer specifying the number of bytes
' read from the Comm Port. It accepts a parameter specifying the
number
' of desired bytes to read.
Public Function Read(ByVal Bytes2Read As Integer) As Integer
    Dim iReadChars, iRc As Integer

    ' If Bytes2Read not specified uses BufferSize
    If Bytes2Read = 0 Then Bytes2Read = miBufferSize
    If mhRS = -1 Then
        Throw New ApplicationException( _

```



```

        "Please initialize and open port before using this
method")
    Else
        ' Get bytes from port
        Try
            ' Purge buffers
            'PurgeComm(mhRS, PURGE_RXCLEAR Or PURGE_TXCLEAR)
            ' Creates an event for overlapped operations
            If meMode = Mode.Overlapped Then
                pHandleOverlappedRead(Bytes2Read)
            Else
                ' Non overlapped mode
                ReDim mabtRxBuf(Bytes2Read - 1)
                iRc = ReadFile(mhRS, mabtRxBuf, Bytes2Read,
iReadChars, Nothing)
                If iRc = 0 Then
                    ' Read Error
                    Throw New ApplicationException( _
                        "ReadFile error " & iRc.ToString)
                Else
                    ' Handles timeout or returns input chars
                    If iReadChars < Bytes2Read Then
                        Throw New IOException("Timeout
error")
                    Else
                        mbWaitOnRead = True
                        Return (iReadChars)
                    End If
                End If
            End If
        Catch Ex As Exception
            ' Others generic erroes
            Throw New ApplicationException("Read Error: " &
Ex.Message, Ex)
        End Try
    End If
End Function

' This subroutine writes the passed array of bytes to the
' Comm Port to be written.
Public Overloads Sub Write(ByVal Buffer As Byte())
    Dim iBytesWritten, iRc As Integer

    If mhRS = -1 Then
        Throw New ApplicationException( _
            "Please initialize and open port before using this
method")
    Else
        ' Transmit data to COM Port
        Try
            If meMode = Mode.Overlapped Then
                ' Overlapped write
                If pHandleOverlappedWrite(Buffer) Then
                    Throw New ApplicationException( _
                        "Error in overlapped write")
                End If
            Else

```

```
        ' Clears IO buffers
        PurgeComm(mhRS, PURGE_RXCLEAR Or PURGE_TXCLEAR)
        iRc = WriteFile(mhRS, Buffer, Buffer.Length, _
            iBytesWritten, Nothing)
        If iRc = 0 Then
            Throw New ApplicationException( _
                "Write Error - Bytes Written " & _
                iBytesWritten.ToString & " of " & _
                Buffer.Length.ToString)
        End If
    End If
    Catch Ex As Exception
        Throw
    End Try
End If
End Sub

' This subroutine writes the passed string to the
' Comm Port to be written.
Public Overloads Sub Write(ByVal Buffer As String)
    Dim oEncoder As New System.Text.ASCIIEncoding
    Dim aByte() As Byte = oEncoder.GetBytes(Buffer)
    Me.Write(aByte)
End Sub

#End Region

End Class
```