



```

msg = msg & vbCrLf & "Model: " & Model & ", SN: " & sn
reply = MsgBox(msg, vbExclamation + vbYesNo + vbDefaultButton2 + vbSystemModal,
"Activation")

Else
msg = "Activate the following module?" & vbCrLf & "Model: " & Model & ", SN: " & sn
reply = MsgBox(msg, vbQuestion + vbYesNo + vbSystemModal, "Activation")
End If

If reply = vbYes Then

temp = formatdatetime(Date, 2)
mm= qextract(temp,0,1,"/")
dd = qextract(temp, 1,2, "/")
yy = right(qextract(temp, 2,3, "/"),2)
cDebug.LogStatus "Module Data: MM " & MM & " dd:" & dd & " yy:" & yy
cConfig.DUTPrs(i).Obj1.CoefsToModule CInt(yy), CInt(mm), CInt(dd), CInt(rtv)

rtv = cConfig.DUTPrs(i).Obj1.ErrorCode

If rtv <> 0 Then
msg = "An error occurred while writing the coefficients to the
module." & vbCrLf & "Error: " & rtv & vbCrLf
msg = msg & cConfig.DUTPrs(i).Obj1.ErrorDescription

Else
msg = "A message titled ""Set Module Calibration Date"" may be behind COMPASS."
& vbCrLf
msg = msg & "Select this message from task bar when it appears."

End If

MsgBox msg, vbExclamation + vbSystemModal, "WARNING"

cCOMPASS.StatusDisplay " DUT " & i & "): Calibration update completed"
cDebug.LogStatus " DUT " & i & "): Calibration update completed"
Else
msg = "Recycling the power of the module will "
msg = msg & "remove the coefficients from the memory and reset the module."
MsgBox msg, vbSystemModal, "Activate Calibration"
End If

Set cConfig.DUTPrs(i).Obj1 = Nothing

```

```

        End If
    Next
End If

Exit Function

Case 300 'Setting a new target pressure
    cDebug.LogStatus "iP=" & iP & ",iC=" & iC & ",PCycles=" & cTest.TestPrsCycles
    If Not (iP = 1 And (iC = 2 And cTest.TestPrsCycles = 2)) Then
        'Only on 2nd cycle first point when the test has 2 cycles
        Exit Function
    End If

Case Else
    Exit Function

End Select

cDebug.LogStatus "*****"
cDebug.LogStatus "Zero/Span Cal Begin"
cDebug.LogStatus "iC,iP, Cycles:" & iC & "," & iP & "," & cTest.TestPrsCycles
cDebug.LogStatus "*****"

Do 'A loop is around the entire process in the event of a failure

'This macro stores the active pressure and subtracts it from
'each reading. No internal changes to the module are made

For calPt = 1 To 2 'min/max
    If calPt = 1 Then
        cCOMPASS.StatusDisplay "Zero/Span Cal: Adjusting module zero ..."
        cDebug.LogStatus "Zero/Span Cal: Adjusting module zero ..."
        tval = 0 'absolute 0 for absolute module, vent for others
    Else
        cCOMPASS.StatusDisplay "Zero/Span Cal: Adjusting module span..."
        cDebug.LogStatus "Zero/Span Cal: Adjusting module span..."
        tval = cConfig.DUTPrs(1).RangeMain.MaxFinal
        valMin = cConfig.DUTPrs(1).RangeMain.MinFinal
        cDebug.LogStatus "Min=" & valMin & ", Max=" & tval
        If tval < 0.0001 And valMin < 0 Then ' vacuum module, should use min as full scale
            cDebug.LogStatus "Vacuum module, use Min as full scale"
            tval = valMin
        End If
    End If
Next

```

```

    End If
End If

DUTUnit = cConfig.DUTPrs(1).RangeMain.UnitFinal
DUTUnitText = cConfig.DUTPrs(1).RangeMain.UnitFinalText
RefUnit = cConfig.RefPrs(1).RangeMain.UnitFinal
setUnit = cConfig.SetPrs(1).RangeMain.UnitFinal

cCOMPASS.StatusDisplay "Zero/Span Cal: Setting target to " & FormatNumber(tval, 2) & " " &
DUTUnitText
cDebug.LogStatus "Zero/Span Cal: Setting target to " & FormatNumber(tval, 2) & " " & DUTUnitText

If cConfig.SetPrs(1).RangeMain.InterfaceMode = 0 Then
    ' Manual then
    'Manual Control
    msg = "Please set the target pressure to " & FormatNumber(tval, 2) & " " & DUTUnitText
    msg = msg & " and press OK when the pressure is stable"
    cCOMPASS.Message CStr(msg), 0
    If cCOMPASS.SystemAbort Then Exit Function

Else
    'Convert to the controllers unit of measure
    tval = cCOMPASS.UnitConversion(CDbl(tval), CInt(setUnit), CInt(DUTUnit), 0)

    If (tval = 0) And (cCOMPASS.cConfig.DUTPrs(1).RangeMain.MeasMode <> 0) Then 'gauge 0
        cDebug.LogStatus "Venting system Pressure"
        cConfig.SetPrs(1).ioSetOutput 0, 0, 1
    Else
        cDebug.LogStatus "Setting Pressure in Controller unit: " & tval
        cConfig.SetPrs(1).ioSetOutput CDbl(tval), 0, 0
    End If

    If Fluke700P_WaitForReady(1, 120) = False Then Exit Function
    If cCOMPASS.SystemAbort Then Exit Function

    Fluke700P_Dwell 10

    If cCOMPASS.SystemAbort Then Exit Function

    cDebug.LogStatus "Zero/Span Cal: Pressure is ready... Err:" & Err
End If

dUnit = cConfig.DUTPrs(1).RangeMain.UnitFinal

```

```

For i = 1 To cConfig.DUTPrs.Count
  Manuf = cConfig.DUTPrs(i).RangeMain.GetParent.Manufacturer
  sn = cConfig.DUTPrs(i).RangeMain.GetParent.sn
  cDebug.LogStatus "DUT: " & i & ") Manufacturer=" & Manuf & ",SN=" & sn
  If InStr(Ucase(Manuf), "FLUKE") Then
    'Only for Fluke Modules
    'Send the zero or Span Flag...

    cConfig.DUTPrs(i).SetParamData 0, 0 'Remove existing zero correction
    cConfig.DUTPrs(i).RangeMain.FinalOutput = -9999

    DUTPrs = Fluke700P_ReadPressure(5, CInt(i))
    RPrs = cCOMPASS.cConfig.RefPrs(1).GetParamData(1)
    cDebug.LogStatus " DUT: " & i & ") DPrs=" & DUTPrs & ",RPrs=" & RPrs
    If cCOMPASS.SystemAbort Then Exit Function

    Max = cConfig.DUTPrs(i).RangeMain.MaxFinal
    Min = cConfig.DUTPrs(i).RangeMain.MinFinal
    MinPSI = cCOMPASS.UnitConversion(Cdbl(Min), 9, CInt(dUnit), 0)
    MaxPSI = cCOMPASS.UnitConversion(Cdbl(Max), 9, CInt(dUnit), 0)
    cDebug.LogStatus " DUT: " & i & ") Min=" & Min & ",Max=" & Max
    cDebug.LogStatus " DUT: " & i & ") MinPSI=" & MinPSI & ",MaxPSI=" & MaxPSI

    'Boundary Check the value
    testpcnt = 0.05

    If ((MinPSI > 0) And (MinPSI < 0.5)) Then ' p01 and p00
      testpcnt = 0.08
    End If

    If calPt = 1 Then 'zero
      cDebug.LogStatus " Zero Cal"
      cConfig.DUTPrs(i).Obj1.StartCalibration CInt(rtv) 'Start the calibration

      'Make sure that the DUT-Ref is close
      If Abs(DUTPrs - RPrs) > Abs(testpcnt * (Max - Min)) Then
        cDebug.LogStatus " Zero Cal: Excessive difference for SN " & sn
        cDebug.LogStatus " Refpressure = " & RPrs & ",DUTPrs=" & DUTPrs & ",pct=" &
testpcnt
          cCOMPASS.cConfig.DUTPrs(i).SetParamData 2, 1
        End If

```

```

cCOMPASS.StatusDisplay "DUT " & i & "): Zero Cal: Logging value " & RPrs
cDebug.LogStatus "    Logging value " & RPrs

newPrs = cCOMPASS.UnitConversion(CDbl(RPrs), 9, CInt(dUnit), 0)
cConfig.DUTPrs(i).Obj1.AppliedZeroReading CSng(newPrs), CSng(newPrs), CInt(rtv)
'reference pressure in psi
cDebug.LogStatus "    AppliedZero: " & newPrs
cDebug.LogStatus "    Return Value =" & cConfig.DUTPrs(i).Obj1.ErrorCode

Else 'full scale
cDebug.LogStatus " Full Scale Cal"
fs = Max
If MaxPSI < 0.0001 Then 'vacuum
    fs = Min
    If MinPSI < -10 Then
        '15psi vacuum must allow the low scale reading value to be as high as -10
        'to accommodate altitude and weather
        testpct = 0.4
    End If
End If

'Make sure that reference is close full scale
If Abs(DUTPrs - fs) > Abs(testpct * (Max - Min)) Then
    cDebug.LogStatus "    Excessive difference for SN " & sn
    cDebug.LogStatus "    full scale=" & fs & ",DUTPrs=" & DUTPrs & ",pct=" & testpct
    cCOMPASS.cConfig.DUTPrs(i).SetParamData 2, 1
End If

cCOMPASS.StatusDisplay "DUT " & i & " ): Span Cal: Logging value " & RPrs
cDebug.LogStatus "    Span Cal: Logging value " & RPrs
newPrs = cCOMPASS.UnitConversion(CDbl(RPrs), 9, CInt(dUnit), 0)
cConfig.DUTPrs(i).Obj1.AppliedFullScaleReading CSng(newPrs), CSng(newPrs), CInt(rtv)
'reference pressure in psi
rtv = cConfig.DUTPrs(i).Obj1.ErrorCode
cDebug.LogStatus "    AppliedFullScale: " & newPrs
cDebug.LogStatus "    Return Code =" & rtv
cDebug.LogStatus "    Return Value =" & rtv

If rtv =0 Then
    'If the return value is True then Adjust the sensor
    cCOMPASS.StatusDisplay "DUT " & i & "): Zero/Span Cal: Adjusting coefficients.."
    cDebug.LogStatus "Zero/Span cal: Adjusting coefficients"

```

```
cCOMPASS.TimeDelay 1
cConfig.DUTPrs(i).Obj1.AdjustCoefs CInt(rtv)
rtv = cConfig.DUTPrs(i).Obj1.ErrorCode
      cDebug.LogStatus " AdjustCoefs Return Code: " & rtv
```

```
Else
```

```
End If
```

```
End If
```

```
End If
```

```
Next
```

```
Next
```

```
'Apply rules to automatically re-adjust
```

```
Exit Do
```

```
Loop
```

```
cDebug.LogStatus "Zero/Span cal Complete"
```

```
cCOMPASS.StatusDisplay ""
```

```
'need to zero the reading after calibration
```

```
'Fluke700P_RunZero
```

```
End Function
```