How to Calibrate Fluke 700 Series DUT with COMPASS for Pressure

Summary

This article describes how to create a Fluke 700 series pressure module DUT as well as how to calibrate it using **COMPASS for Pressure.** The DUT setup and macros used for test setup in this article is available in the example database accessed through the COMPASS download page. Use the **[Database][Database Maintenance]** option to import the DUT setup named "Sample Fluke 700 Series" and Test Macro Fluke700_Zero as well as Fluke700CalibrationAdjust.

Article Topics

- How to create a Fluke 700 Series Pressure Module DUT.
- How to use a DLL to output data to **COMPASS**.
- How to use an Interface Macro device setup.
- How to activate a device using macro.

See Also

N/A

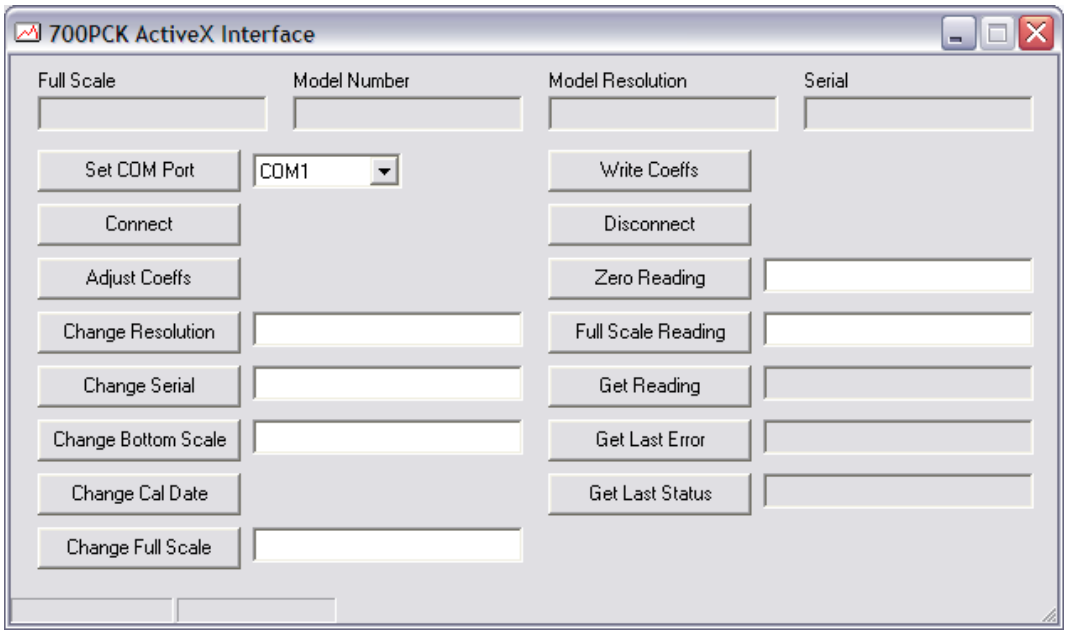
Requirements

The following items are required to implement this setup.

- **COMPASS for Pressure Enhanced**. This is an advanced DUT setup that requires the enhanced version of **COMPASS**.
- Fluke 700 Series Pressure Monitor
- Fluke 700 PCK Software version 3.3 or newer
- Fluke 700 Series **COMPASS** Interface. This is an ActiveX program that allows a **COMPASS** VBScript macro to communicate with the 700 Series interface library (dll). Download the file from the Application Notes page.
- PC with available COM ports. **COMPASS** supports COM ports 1 to 9.
- A separate 700PCK calibration adapter is required for each module when multiple modules are calibrated at the same time. Each adapter must be connected to a unique COM port.

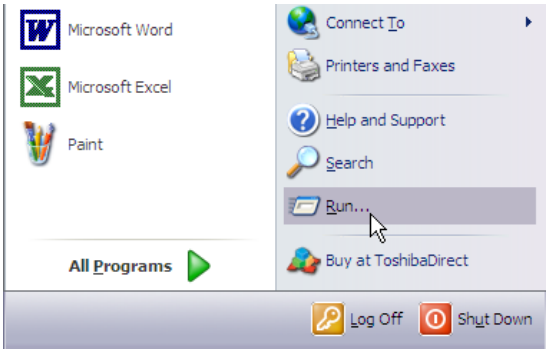
ActiveX Setup

Download the Fluke 700 Series **COMPASS** Interface file **FlukeInterface.exe**. The file is on the Application Notes page next to this Application Note article. This file must be registered with Windows prior to using it with **COMPASS**. Copy the program to the installation directory of the pressure module calibration software. The correct directory will contain the file 7PCKLib.dll. The Flukeinterface.exe file must be in the same directory as the 7PCKLib.dll file to function properly. Register the program by running it and closing it without making any selections. This manual interface is only used for ActiveX DLL registration. It is not a required part of the **COMPASS** implementation and should not be used for manual adjustments to the pressure module.

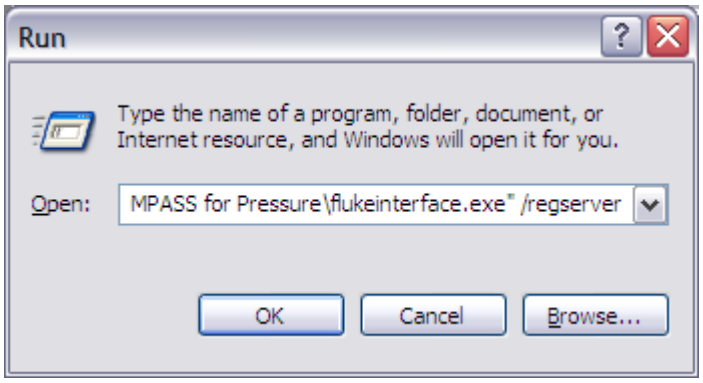


In some cases the file must be registered manually. Follow the steps below to manually register the FlukeInterface.exe

1. Press the Windows Start button and select Run



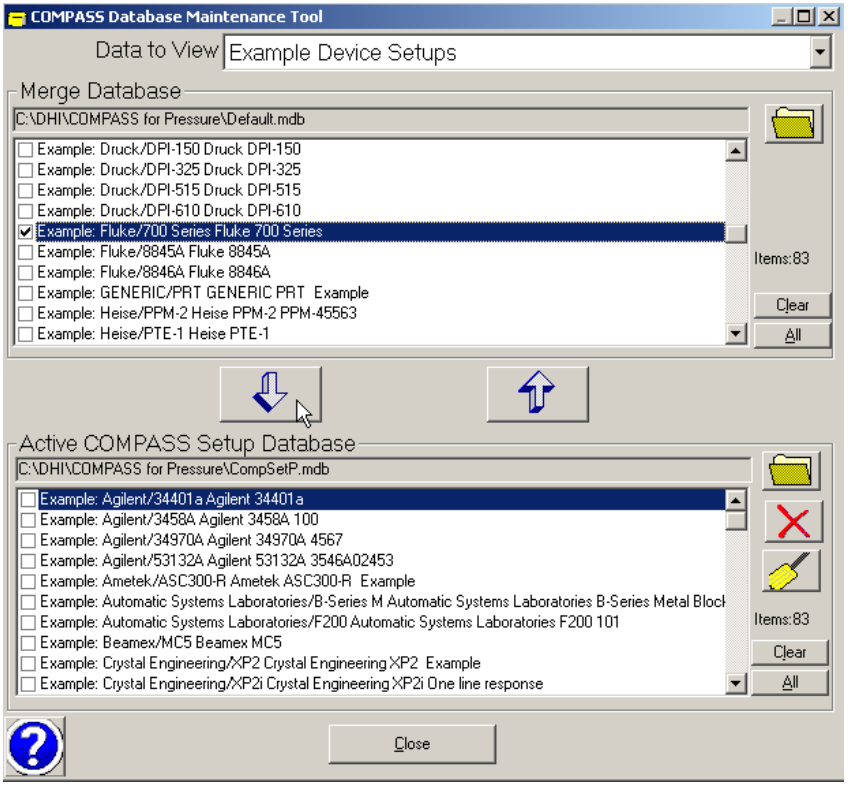
2. Enter the full path and filename to the file in quotes.
3. Enter the text "/regserver" outside of the quotes
4. Press **[OK]** to register the file. No confirmation message will display. However, an error message will display if the registration process fails.



DUT Setup

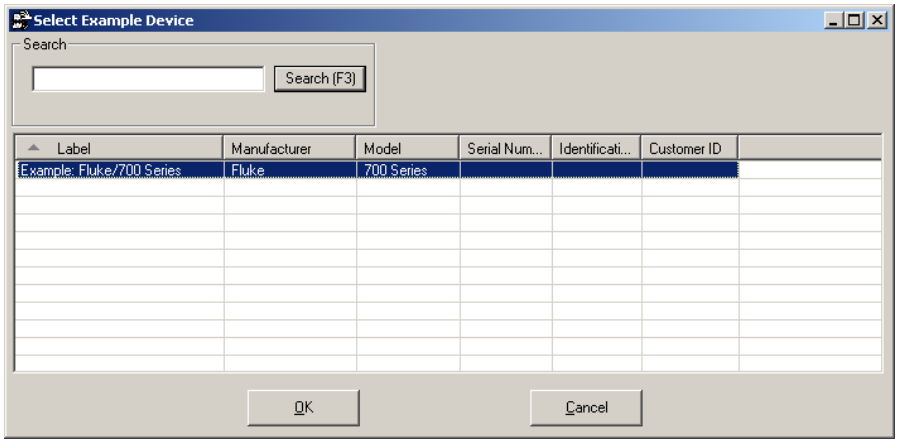
Follow the steps below to create a Fluke 700 Series DUT. Alternatively, merge the DUT information from the Example database as described in the Summary section of this article.

1. Use the **<Database><COMPASS Database Maintenance Tool>** to extract the Fluke 700 Series Pressure Module example. Select the "Example Device Setups" choice in the **<Data to View>** field, locate the "Example: Fluke/700 Series Fluke 700 Series" choice. Then press the down arrow.

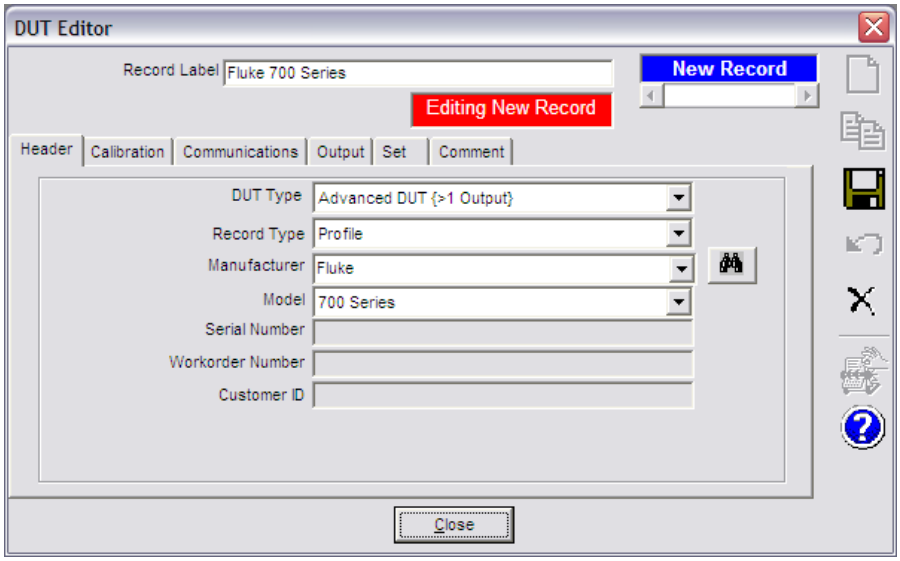


2. Run **COMPASS** and select the **[Setup][DUT]** menu choice. Create a new DUT by pressing the **[New]** button on the top of the toolbar.

3. Select “Fluke” as the **<Manufacturer>** and “Fluke 700 series” as the **<Model>**. Press  and select the appropriate example and press **[OK]**. All DUT specific setup information is copied. The selections in the remaining steps are auto-populated by the example. The steps are listed to clarify the specific requirements of the setup.

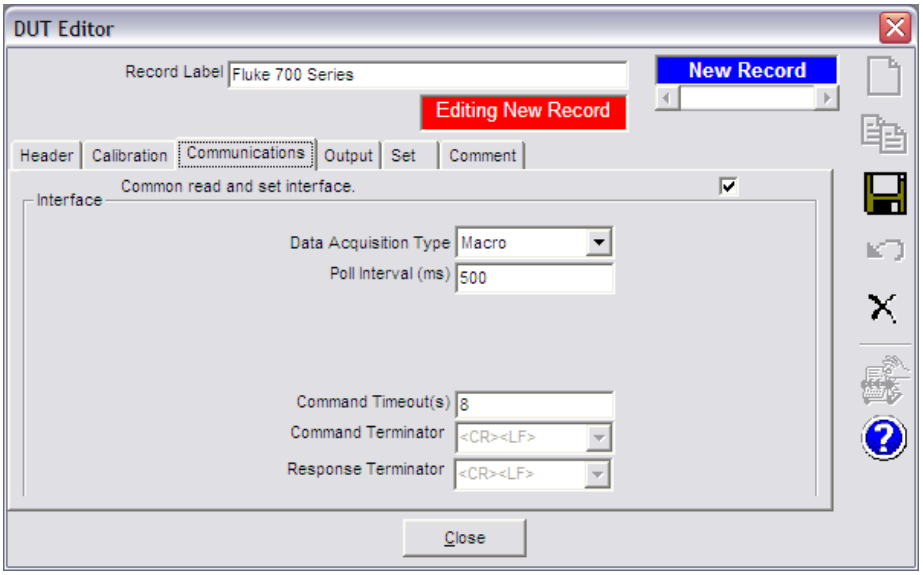


4. Enter the basic setup information for the DUT. Use **<Profile>** as the **<Record Type>** selection so that this DUT can be used for any Fluke 700 Series sensor.



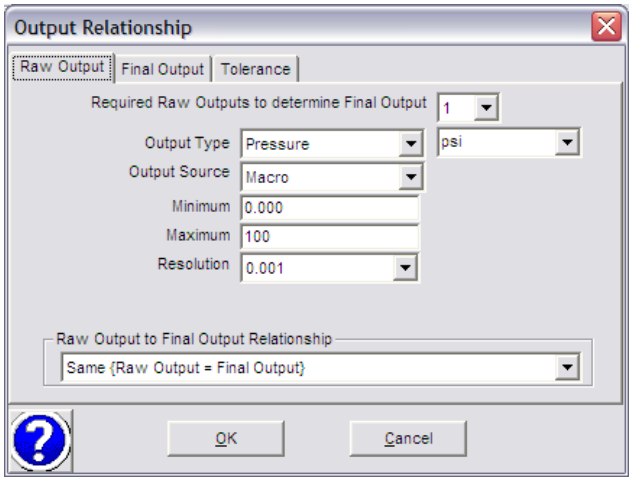
5. No specific settings are required on the **[Calibration]** tab. This tab is provided to maintain calibration information or to associate a specific test with the DUT.

6. Select **<Macro>** as the **<Data Acquisition Type>** on the **[Communications]** tab. Use 500ms as the **<Poll Interval>**. This time represents the interval between successive calls to the communications macro.

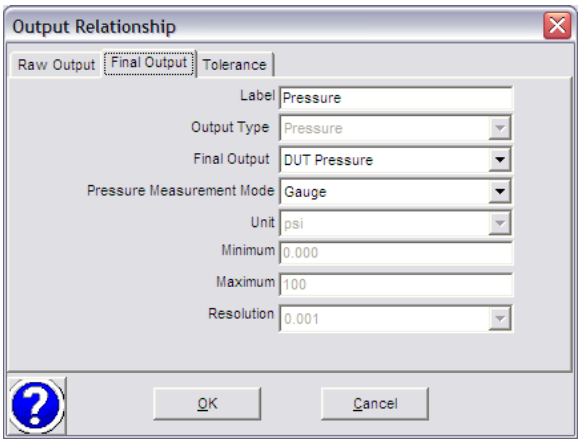


7. On the **[Output]** tab, press the **[Edit Output]** button to edit the default device output. A default output is always created for new devices.

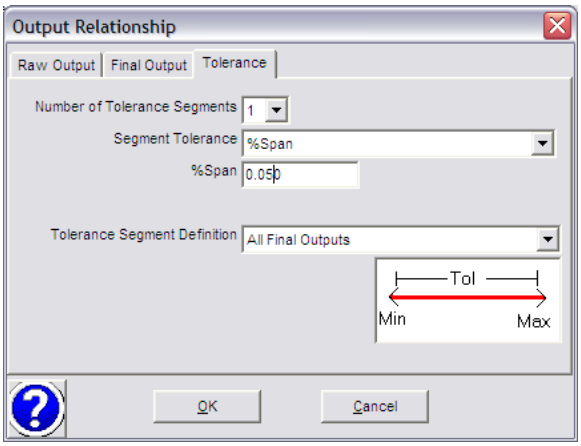
8. Select the **[Raw Output]** selections defined in the figure below. Since this device is setup as a profile, the actual test range and resolution can be changed while initializing a test. Enter the most commonly used test range to use as a default. Select the appropriate pressure unit. Most Fluke 700 models use psi with the exception that P00 and P01 use in H2O.



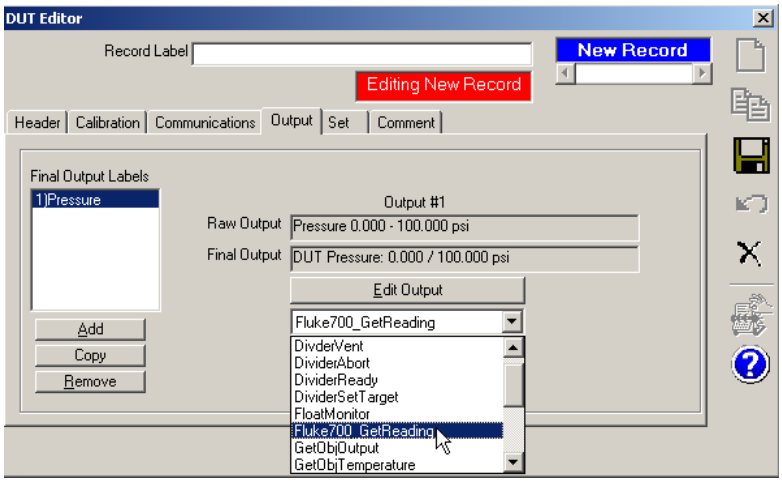
9. Since **<Same>** is selected as the output relationship, many of the selections on the **[Final Output]** tab are disabled. The most important selection is **<Final Output>**. This entry must be **<DUT Pressure>** to use this output as a DUT. Use the **<Label>** entry to customize the label for this output if desired.



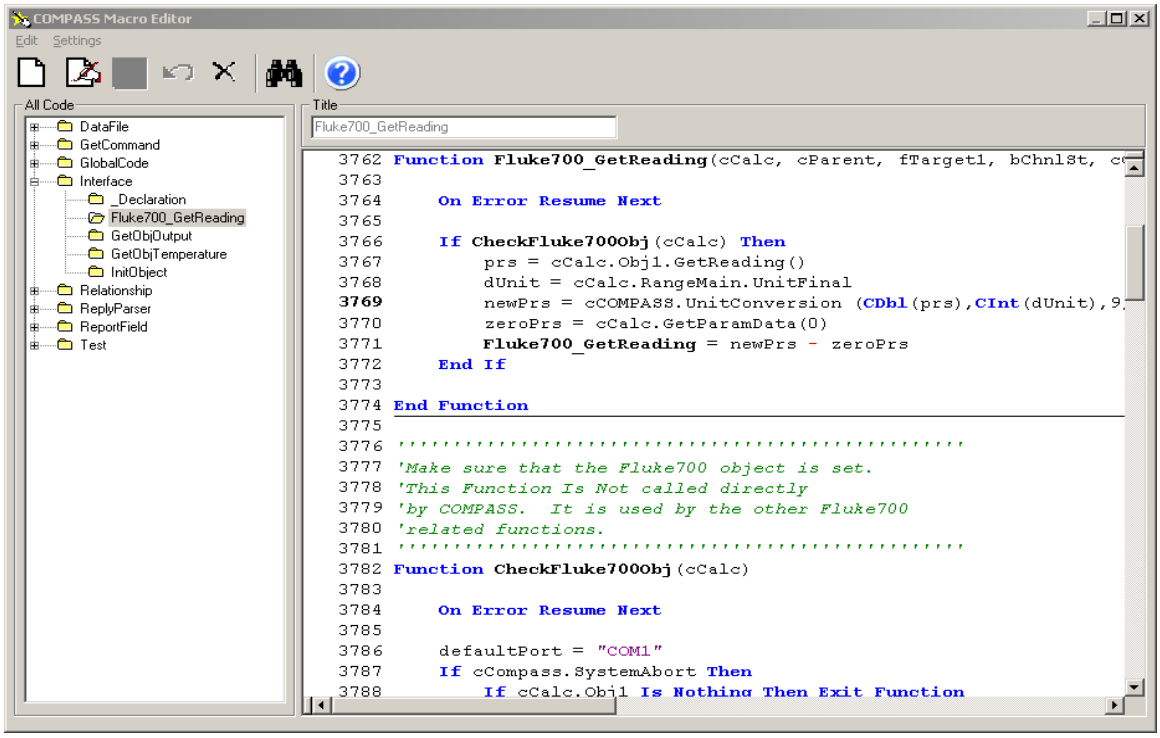
10. Use the **[Tolerance]** tab to define the DUT tolerance. Press **[OK]** after tolerance selection to return to the **DUT Editor**.



11. Select “Fluke700_GetReading” from the list of macros under the **[Edit Output]** button. This macro is available only if the database was properly merged in steps 1 and 2. The list contains the complete list of Interface Macros in the active **COMPASS** database.



12. There are no other unique selections required for this device. Press **[Save]** to activate all edits. Then close the **DUT Editor**.
13. By default the Fluke700_GetReading Interface Macro uses COM1. The macro must be edited to use an alternate port by default. Use **[Tools][COMPASS Macro Editor]** to edit the macro. Select **<Interface>** on the left then locate the Fluke700_GetReading macro. Press CTRL+E to enter edit mode. Alter the line of code `defaultPort = "COM1"`, to "COM3" or another value to alter the default RS232 port. Press the **[Save]** button to save the macro after the edits are complete.



Fluke700_GetReading Macro Code

```
'****************************************
'This Function must read and return the output of a remote instrument.
'The output must be in the Raw Output Unit defined by the Range class.
'
'cCalc     :The calculation class of the device.
'cParent   :The parent device class.
'fTarget1  :Target output for controller.
'bChnlSt   :True/False to activate or de-active a valve driver.
'cConfig   :Configuration class that holds all active calculation classes.
'
'The value is returned by setting the function name = 'to the calculated value..
'
'For example: Interface33489 =  val(mid(rawReply,5))
'****************************************

Function Fluke700_GetReading(cCalc, cParent, fTarget1, bChnlSt, cConfig)
```

```
        On Error Resume Next

        If CheckFluke700Obj(cCalc) Then
                prs = cCalc.Obj1.GetReading()
                dUnit = cCalc.RangeMain.UnitFinal
                newPrs = cCOMPASS.UnitConversion (CDbl(prs),CInt(dUnit),9,0)
                zeroPrs = cCalc.GetParamData(0)
                Fluke700_GetReading = newPrs - zeroPrs
        End If

End Function

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Make sure that the Fluke700 object is set.
'This Function Is Not called directly
'by COMPASS.  It is used by the other Fluke700
'related functions.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Function CheckFluke700Obj(cCalc)

        On Error Resume Next

        defaultPort = "COM1"
        If cCompass.SystemAbort Then
                If cCalc.Obj1 Is Nothing Then Exit Function

                cCalc.Obj1.Disconnect
                Set cCalc.Obj1  = Nothing
                cDebug.LogStatus "Disconnected Fluke Interface. Err:" & ERR
                Exit Function
        End If

        If cCOMPASS.SystemAbort Then Exit Function


        If cCalc.Obj1 Is Nothing  And cCalc.GetParamData (1)= 0 Then
             'Check to see if this is DUT #1
           PortInfo = GetVal(cCOMPASS.cConfig.DUTPrs(1).RangeMain.GetParent.ParamID)
           cDebug.LogStatus "*******Fluke Interface INIT********:"
           cDebug.LogStatus "Model: " & cCalc.RangeMain.GetParent.Model
           cDebug.LogStatus "SN: " & cCalc.RangeMain.GetParent.SN
           cDebug.LogStatus "ParamData:" & cCalc.RangeMain.GetParent.ParamID
           cDebug.LogStatus "PortInfo:" & PortInfo

           If PortInfo >0 And PortInfo < 10 Then
                  PORT = "COM" & PortInfo
           Else
                  PORT = defaultPort
                  PortInfo = GetVal(right(PORT,1))
           End If

           K1 = cConfig.DUTPrs(1).RangeMain.Key
           P1 = cConfig.DUTPrs(1).RangeMain.GetParent.ProfileID
           P2 = cCalc.RangeMain.GetParent.ProfileID

           If (cCalc.RangeMain.Key = K1 And P2 = P1) Or cConfig.DUTPrs.Count = 1 Then
                'First DUT
                PORT = "COM" & PortInfo
                cDebug.LogStatus "First DUT"
                 DUTID = 1
           Else
                  If cConfig.DUTPrs(1).Obj1 Is Nothing Then
                       cDebug.LogStatus "Wait for First DUT to init"
                       cDebug.LogStatus "Init Exit"
                       Exit Function
```

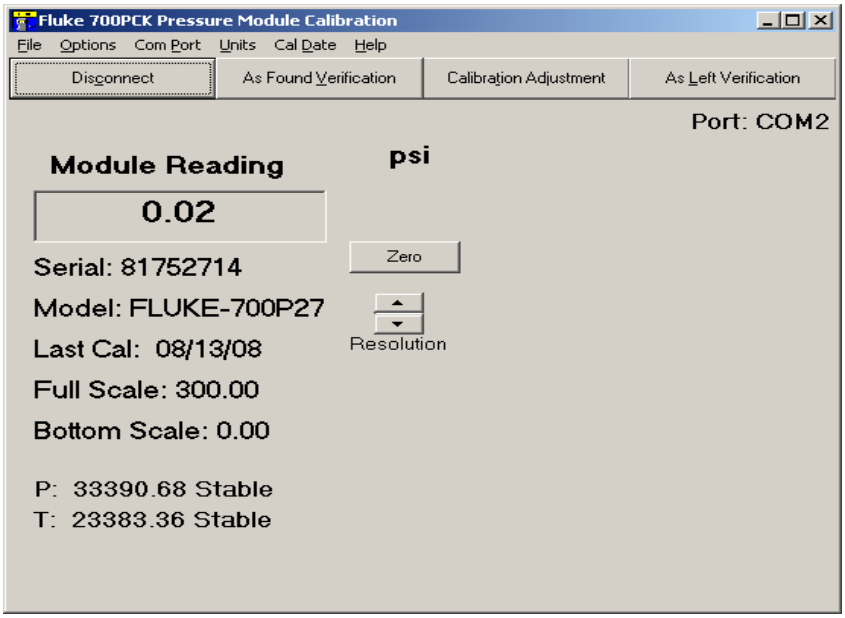
```
                End If

                For i = 2 To cConfig.DUTPrs.Count
                    If cCalc.RangeMain.Key = cConfig.DUTPrs(i).RangeMain.Key  Then
                         PortInfo = cCOMPASS.cConfig.DUTPrs(i).RangeMain.GetParent.ParamID
                        PORT = "COM" & PortInfo
                        cDebug.LogStatus "PortInfo=" & PortInfo & ",Port=" & PORT
                        DUTID = i
                        Exit For
                    End If
                Next

        End If

        'Save the FlukeINterface object
        'to the Calculation class of the device.
        cCOMPASS.StatusDisplay "Updating Fluke Interface"
        cDebug.LogStatus "Initializing Fluke Interface. Err:" & ERR
    Set cObj = GetObject("", "FlukeInterface.clsFluke7pck")

    '****Specify Port to Communicate*****
    Model = cCalc.RangeMain.GetParent.Model
    SN = cCalc.RangeMain.GetParent.SN
    cCOMPASS.StatusDisplay "Model:" & Model & "(SN" & SN & ") Initializing " & PORT
    cObj.ChangeComPort(CStr(PORT))
    cDebug.LogStatus "Setup to use " & PORT & ". Err:" & ERR

        TimeDelay 3  'It takes a few seconds to update the  port

            cCOMPASS.StatusDisplay "Connecting Fluke Interface"
            cObj.Disconnect        'Guarantee module is not connected
            cDebug.LogStatus "Disconnection complete"

            If cObj.Connect() =False Then
                    cCOMPASS.SystemAbort = True
                    cCOMPASS.StatusDisplay "Connection failed on Port " & Port
                    cDebug.LogStatus "Connection Failed on Port " & Port
                    msg = "Connection to Fluke module failed on COM Port " & Port
                    msg = msg & ".  Please verify that the port is "
                    msg = msg & "valid and that the sensor is properly connected.  "
                    msg = msg & "Enter the proper COM port in the Parameter ID field "
                 msg = msg & "to specify a port other than " & PORT & "."
                    Msgbox msg, vbCritical, "Error"
                    cCalc.SetParamData 1,1  'Flag that device failed
                    Exit Function
            End If

            Set cCalc.Obj1 = cObj
            'Update Serial Number of device.
            temp = cCalc.Obj1.Serial
            If len(temp) > 2 Then
                    cCalc.RangeMain.GetParent.SN = temp
                    cDebug.LogStatus "Set SN to :" & temp
            End If

            'Update Model Info of device.
            temp = cCalc.Obj1.ModelNumber
            If len(temp) > 2 Then
                    cCalc.RangeMain.GetParent.Model = temp
                    cDebug.LogStatus "Set Model to :" & temp
            End If

            'bottom sale and full scale
            DUTMin = cCalc.RangeMain.MinFinal
            DUTMax = cCalc.RangeMain.MaxFinal
            cdebug.LogStatus "DUT Min = " & DUTMin & ",DUT Max = " & DUTMax
```

```
                btmScale = cCalc.Obj1.BottomScale
                fullScale = cCalc.Obj1.FullScale
                cdebug.LogStatus "BottomScale = " & btmScale & ",FullScale = " & fullScale

                'If data files were already created, add the Model and SN info the files
                If (cCOMPASS.COMPASSRunState And 2^6) <> 0 Then
                        cCOMPASS.DataCollection(DUTID).DUT.Model = Model
                        cCOMPASS.DataCollection(DUTID).DUT.SerialNumber = SN
                End If

                cDebug.LogStatus "Connection active"
                cCalc.SetParamData 1,2  'Flag that the device is set...

        ElseIf cCalc.GetParamData(1) =1 Then
                'Tried and failed
                Exit Function
        End If

        CheckFluke700Obj = True

End Function
```

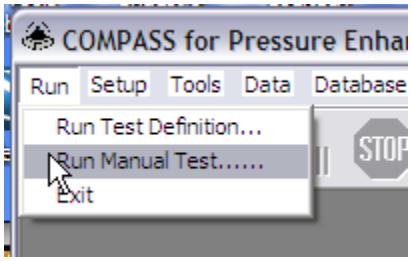
Setup Validation

It is good practice to test unique setups one at a time to minimize troubleshooting when running a complete test. The Fluke 700 Series DUT setup is easily tested by running a manual test. Follow the steps below to verify the DUT setup.

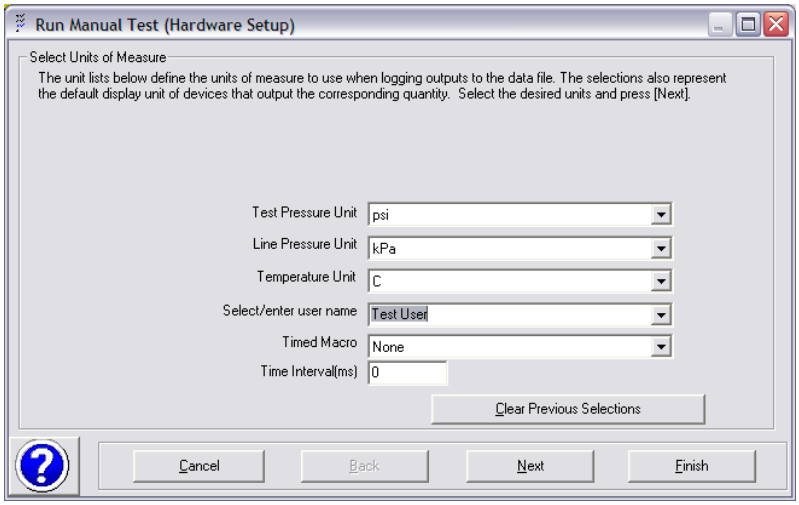
1. Connect the DUT to the PC's COM1 or COM2 port and ensure that all power and communications cables are properly connected.
2. Run the 700 Series Pressure Module Calibration Software to verify that the interface is correctly configured and functioning properly. If the software displays a pressure output, the DUT setup in **COMPASS** will work. If the pressure output does not display make the necessary changes to the COM port or physical cable connections to resolve the issue. Close the program when valid communications have been verified.



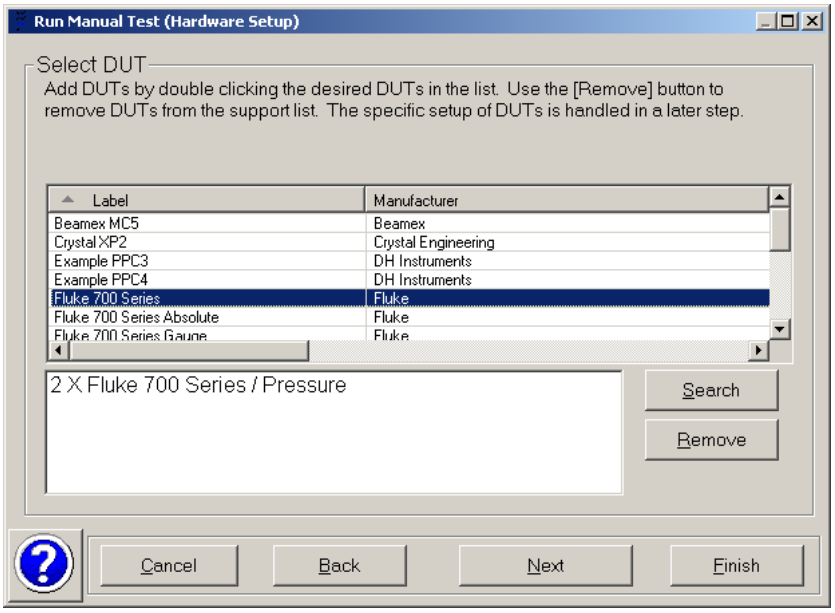
3. Run a Manual Test by choosing **[Run][Run Manual Test]** in **COMPASS.**



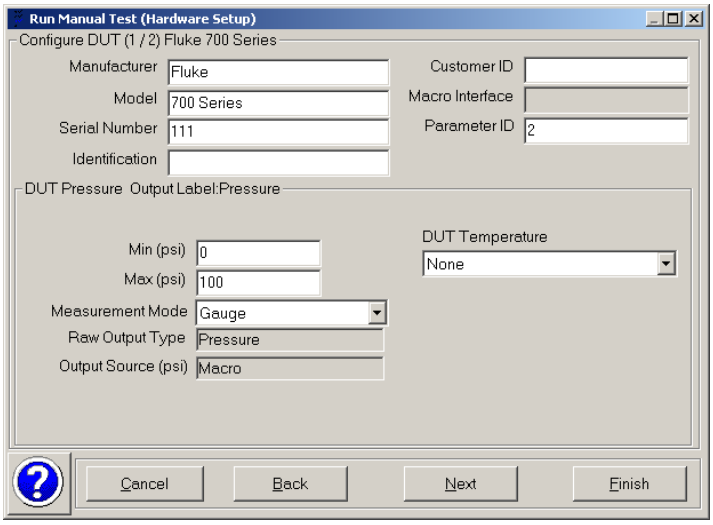
4. Select “psi” (or the DUT defined pressure unit) as the pressure unit and press the **[Clear Previous Selections]** button on the opening test initialization dialog. Press **[Next]** to move to the next initialization step.



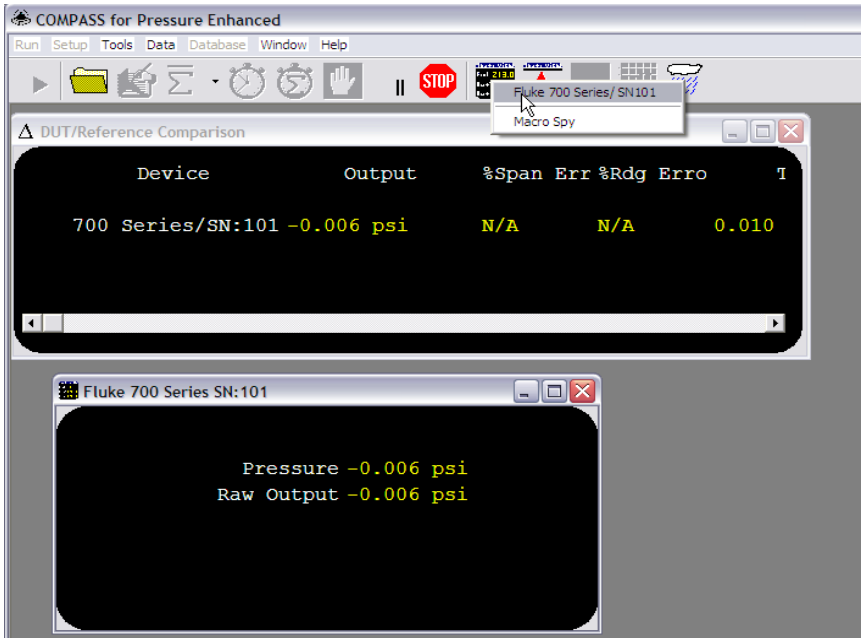
5. Double-click the Fluke 700 Series DUT from the list and press **[Next]**. Double click a second time to calibrate more than 1 module at a time.



6. This DUT is setup as a profile so it is possible to change range and other identifying characteristics during initialization. Enter the serial number of the pressure module and specify the actual pressure range. Press the **[Customize Output]** button to alter the tolerance or resolution of the DUT. Use the **<Parameter ID>** field to alter the RS232 port of the sensor. Enter the port number, i.e. "2" for COM2 or 5 for COM5. The maximum allowable COM port is 9. COM1 is used by default. Note that the 700PCK software only supports COM1 and COM2. When calibrating more than one module at a time, the COM port of the first DUT is auto incremented to support other DUTs. If the first DUT uses COM3, the third DUT initialized will use COM5. Press **[Finish]** after all changes are complete. There are no other required selections for this manual test initialization.



7. The manual test begins by initializing and polling all of the DUTs. The output of the pressure module should display on the **DUT/Reference Comparison** display. Use the toolbar to open other available output displays. If the pressure does not display, select the **[Macro Spy]** option to determine the last valid macro function.

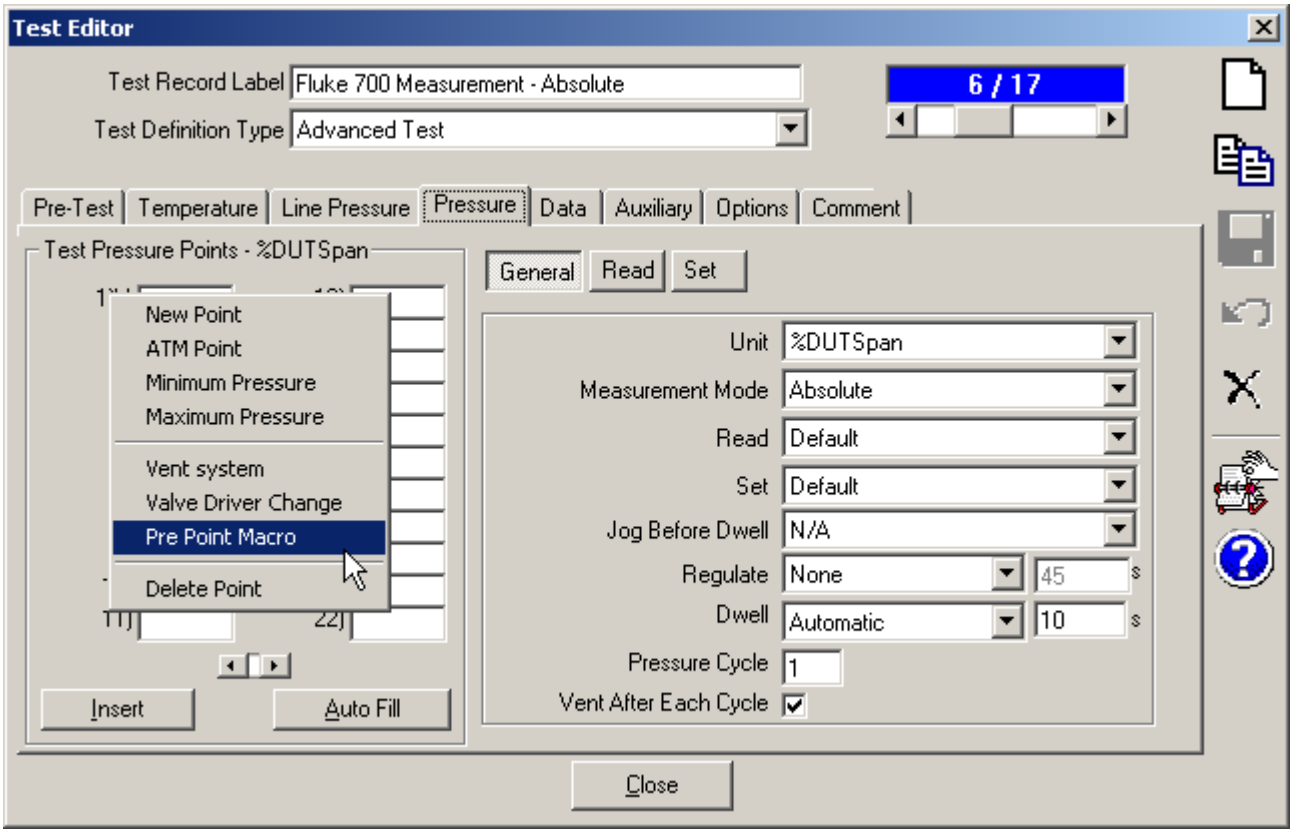


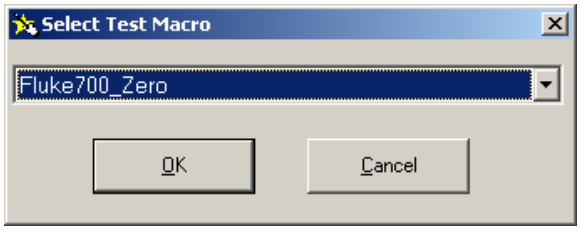
8. Press the stop sign to abort the manual test.
9. The DUT is now available for testing with any calibration reference.

Module Measurement with Zero

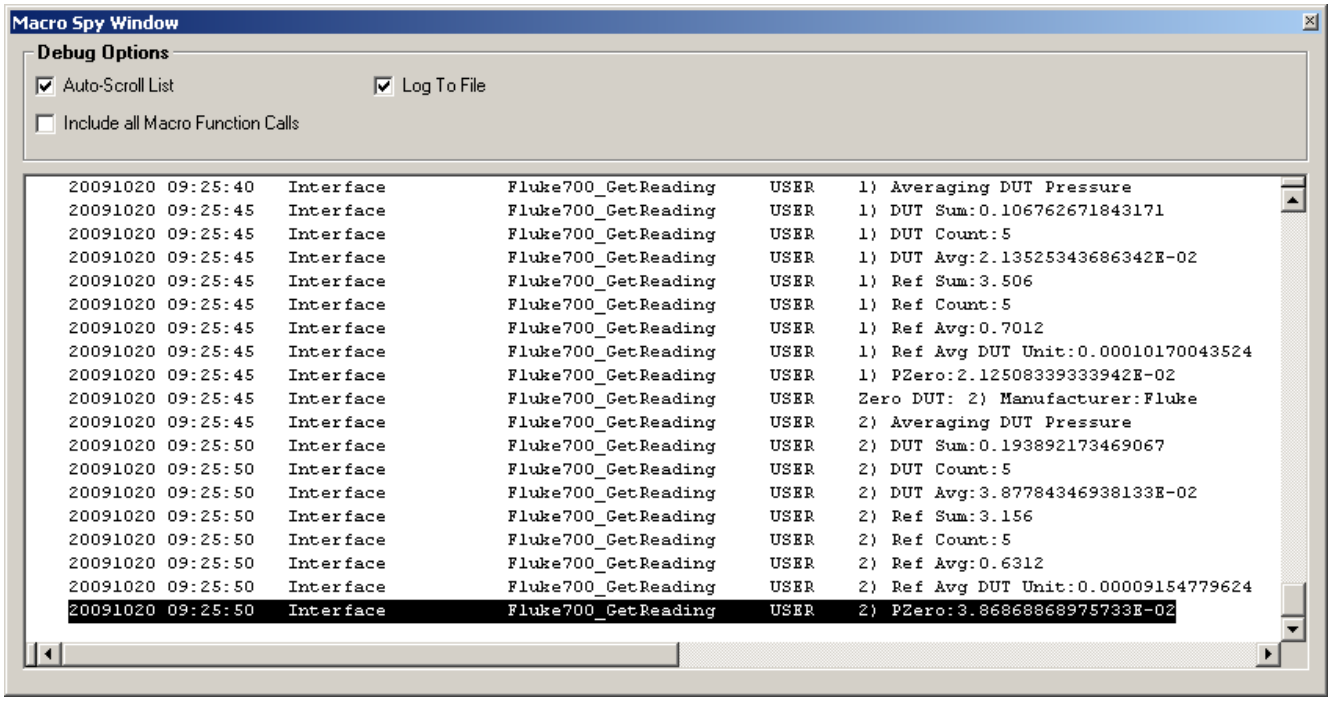
If the test is intended to measure the performance of the pressure module without conducting calibration, set “Pressure Cycle” to “1” in [Pressure][General] tab of test definition. It is common practice to zero the module prior to starting a test.

1. Create a Test Definition and click the first test point label to access the pre-point options menu selection then select **[Pre-Point Macro]**. Select ***Fluke700_Zero*** macro if present or select **<Add/Edit Macro>** to create a new Test Macro.





2. This macro vents the active test pressure, averages 5 samples of the DUT and reference values then determents the zero correction. In gauge mode the average DUT value at zero is subtracted from all subsequent readings. In absolute mode, the difference between the DUT and reference is subtracted from all subsequent values. The ***Fluke700_GetReading*** Interface macro subtracts the zero reading to correct the zero of each pressure module. This macro is designed to function with an automated reference device.



Fluke700_Zero Macro Code

```
'*************************************************
'FOR HELP ON VBSCRIPT, PLEASE VISIT:
'http://www.webf1.com/scriptdoc/vbscript/VBSTOC.htm
'
'This macro zero's the Fluke 700 pressure modules
'*************************************************
'Test Macro do not have a return value.
'Manipulate the test or device collection as desired.
'iT  The current temperature point in the test
'iL  The current line pressure point in the test.
'iC  The current pressure cycle in the test.
'IP  The current pressure point in the test.
'cTest   The test class .
'cConfig  Configuration class that holds all active devices.
'*************************************************
Function Fluke700_Zero(iT, iL, iC, iP, cTest, cConfig)

	If iC > 1 Then
		'Only zero on first cycle
		Exit Function
	End If

	If cCOMPASS.SystemAbort = True Or cCOMPASS.cConfig.DUTPrs(1).GetParamData (1)= 1 Then
		cCOMPASS.AbortTest True
		Exit Function
	End If

	Fluke700_RunZero

End Function
```

```
Function Fluke700_RunZero()
        'This macro stores the active pressure and subtracts it from
        'each reading. No internal changes to the module are made
        cCOMPASS.StatusDisplay "Adjusting module zero..."
        cDebug.LogStatus "Adjusting Module Zero"

        cDebug.LogStatus "Venting system Pressure"
        cCOMPASS.cConfig.SetPrs(1).ioSetOutput 0, 0, 1

        If Fluke700_WaitForReady(1,120) = False Then Exit Function
        If cCOMPASS.SystemAbort Then Exit Function

        Fluke700_Dwell 10
        If cCOMPASS.SystemAbort Then Exit Function

        ''' this assumes the interface macro has executed and the dut is initialized.
        For i = 1 To cCOMPASS.cConfig.DUTPrs.Count
                Manuf = cCOMPASS.cConfig.DUTPrs(i).RangeMain.GetParent.Manufacturer
                sn = cCOMPASS.cConfig.DUTPrs(i).RangeMain.GetParent.SN
                cDebug.LogStatus "Zero DUT: " & i & ") Manufacturer=" & Manuf & ",SN=" & sn

                If instr(Ucase(Manuf),"FLUKE") Then
                    cCOMPASS.cConfig.DUTPrs(i).SetParamData 0, 0 'Remove existing zero correction
                    cCOMPASS.cCOnfig.DUTPrs(i).RangeMain.FinalOutput = -9999

                    'read and average pressure
                    DPrs = Fluke700_ReadPressure(5,i)
                    RPrs = cCOMPASS.cConfig.RefPrs(1).GetParamData(1)
                    cDebug.LogStatus i & ") DPrs=" & DPrs & ",RPrs=" & RPrs
                    If cCOMPASS.SystemAbort Then Exit Function

                    If cCompass.cCurTest.TestPrsMeasMode = 0 Then '''Is this absolute mode?

                        If RPrs <= 0 Then
                            cDebug.LogStatus i & ") Invalid Absolute Reference Pressure:" & RPrs
                            Exit Function
                        End If

                        ZeroPrs = DPrs - PRef 'In absolute mode difference in Pressure is Zero
                    Else
                        ZeroPrs = DPrs
                    End If

                    'Store offset in data file as Calibration Coefficient #1.
                    'The offset is available in Reports.
                    dUnitText = cConfig.DUTPrs(i).RangeMain.UnitFinalText
                    cDebug.LogStatus i & ") PZero:" & ZeroPrs & " " & dUnitText
                    cConfig.DUTPrs(i).SetParamData 0, CDbl(ZeroPrs)
                    cCOMPASS.DataCollection(i).DUT.CalibrationCoefficient1 = formatnumber(ZeroPrs,4)

                End If
        Next

        cCOMPASS.StatusDisplay ""

End Function


Function Fluke700_Dwell(dwell)
        cDebug.LogStatus "Dwell for " & dwell & "s"
        tStart = timer
        Do
                cCOMPASS.TimeDelay 1
                td = CInt(dwell-Time_Difference(tStart))
                cCOMPASS.StatusDisplay "Dwell..." & td
                If cCOMPASS.SystemAbort Then Exit Function
```

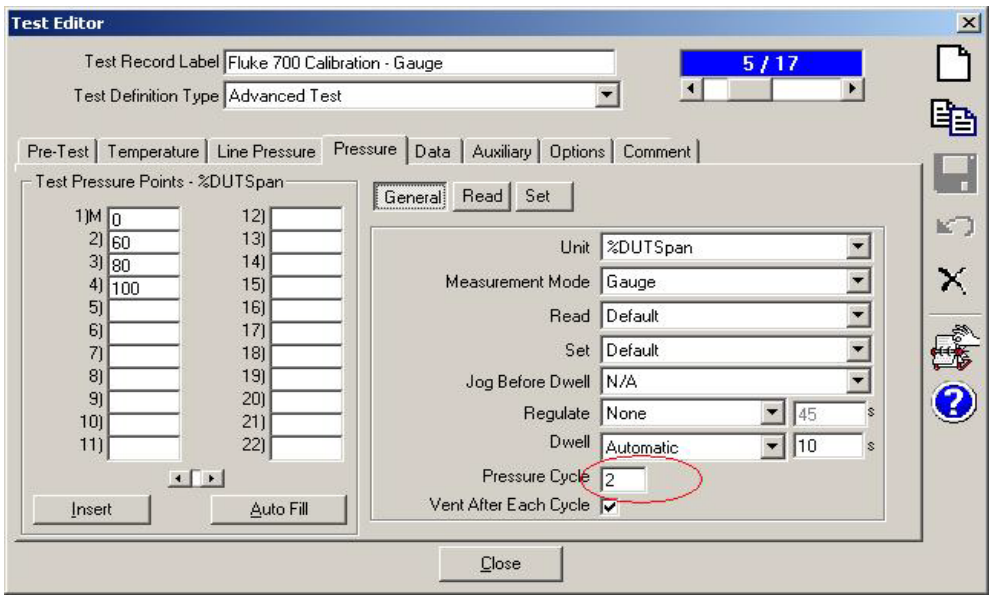
```
                If td <= 0 Then Exit Do
        Loop Until False
End Function

Function Fluke700_WaitForReady(isZero,timeout)
        cDebug.LogStatus "Wait for Ready: timeout=" & timeout
        tStart = timer
        Do
                cCOMPASS.TimeDelay 2    'delay controller
                cCOMPASS.StatusDisplay process & ": Waiting for Ready........."
                If cCOMPASS.SystemAbort Then Exit Function

                If Time_Difference(tStart) > timeout Then
                        cDebug.LogStatus "*******TIMEOUT WAITING for Ready****"
                        If (cCOMPASS.cConfig.DUTPrs(1).RangeMain.MeasMode = 0) And (isZero=1) Then
                                cDebug.LogStatus "Timeout for absolute 0"
                                Exit Do 'absolute 0, as good as possible
                        End If
                        Fluke700_SetPressure = False
                        Exit Function
                End If

                If cConfig.SetPrs(1).RangeMain.Useready Then
                        cDebug.LogStatus "Wait for ready...SetPrs.Ready=" & cConfig.SetPrs(1).Ready
                        If cConfig.SetPrs(1).Ready Then  Exit Do
                Else
                        cDebug.LogStatus "Wait for ready...RefPrs.Ready=" & cConfig.RefPrs(1).Ready
                        If cConfig.RefPrs(1).Ready Then Exit Do
                End If
        Loop
        Fluke700_WaitForReady = True
End Function

Function Fluke700_ReadPressure(nAvg,iDUT)
        DPrs = -9999
        DSum = 0
        DCnt = 0
        RPrs = -9999
        RSum = 0
        RCnt = 0

        cCOMPASS.StatusDisplay "Averaging DUT Pressure..."
        cDebug.LogStatus iDUT & ") Averaging DUT Pressure"

        Do
                cCOMPASS.TimeDelay 1
                If cCOMPASS.SystemAbort Then Exit Function

                DPrs = cCOnfig.DUTPrs(iDUT).RangeMain.FinalOutput

                If DPrs <> -9999 Then
                        DSum = DSum + DPrs   'Update Sum
                        DCnt = DCnt + 1      'Update Increment

                End If

                RPrs = cCOnfig.RefPrs(1).RangeMain.FinalOutput

                If RPrs <> -9999 Then
                        RSum = RSum + RPrs   'Update Sum
                        RCnt = RCnt + 1      'Update Increment

                End If

        loop While DCnt < nAvg And RCnt < nAvg
```

```
        cDebug.LogStatus iDUT & ") DUT Sum:" & DSum
        cDebug.LogStatus iDUT & ") DUT Count:" & DCnt
        DPrs = DSum/DCnt
        cDebug.LogStatus iDUT & ") DUT Avg:" & DPrs

        cDebug.LogStatus iDUT & ") Ref Sum:" & RSum
        cDebug.LogStatus iDUT & ") Ref Count:" & RCnt
        RPrs = RSum/RCnt
        cDebug.LogStatus iDUT & ") Ref Avg:" & RPrs

        'Reference pressure is averaged in it's Unit of measure
        'Convert to DUT's Unit of Measure
        dUnit = cConfig.DUTPrs(iDUT).RangeMain.UnitFinal
        rUnit = cConfig.RefPrs(1).RangeMain.UnitFinal
        PRef = cCOMPASS.UnitConversion(CDbl(RPrs),CInt(dUnit),CInt(rUnit),0)
        cDebug.LogStatus iDUT & ") Ref Avg DUT Unit:" & PRef

        'store Ref prs
        cCOMPASS.cConfig.RefPrs(1).SetParamData 1,CDbl(PRef)
        'return DUT prs
        Fluke700_ReadPressure = DPrs
End Function
```

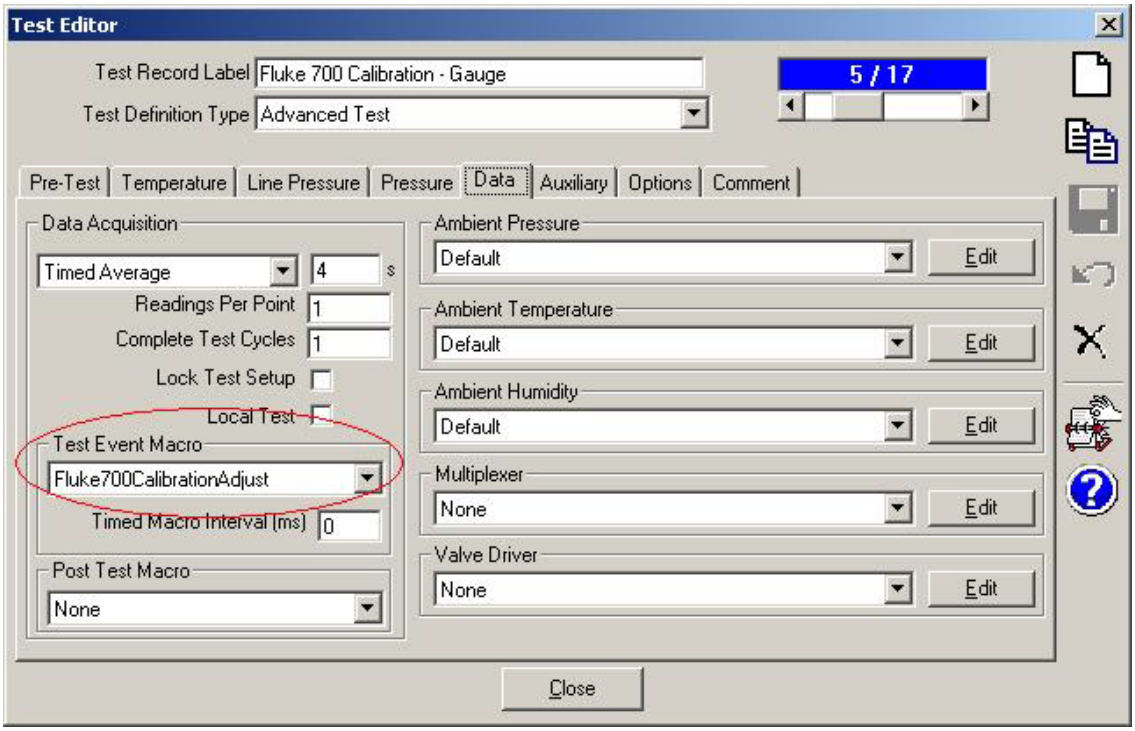
Module Calibration

Macros have been developed to support calibrating the module and activating the calibration at the end of test. The test should have 2 pressure cycles, as received data is collected in cycle 1 and as left data in cycle 2. Zero and span will be performed in prior to running cycle 2 to collect as left data. At the end of test, the user has the option to write the calibration into the module.

1. It is common practice to zero the module prior to starting a test. Create a Test Definition and click the first test point label to access the pre-point options menu selection then select **[Pre-Point Macro]**. Select ***Fluke700_Zero*** macro if present or select **<Add/Edit Macro>** to create a new Test Macro.
2. Set “Pressure Cycle” to “2” in [Pressure][General] tab of test definition. As received data is collected in cycle 1 and as left data in cycle 2.



3. On [Data] tab, select ***Fluke700CalibrationAdjust*** macro if present or select **<Add/Edit Macro>** to create a new Test Macro.



4. The macro ***Fluke700CalibrationAdjust*** conducts zero and span calibration at the beginning of cycle 2. The calibration info is store in the memory of the module at this moment. Then the data gets collected with the calibration. At the end of test, the user can elect to write the calibration into the module. If user selects not to activate the calibration, the calibration will be removed after cycling power of the module.

Fluke700CalibrationAdjust Macro Code

```
'************************************************
'FOR HELP ON VBSCRIPT, PLEASE VISIT:
'http://www.webf1.com/scriptdoc/vbscript/VBSTOC.htm

'************************************************
'Test Macro do not have a return value.
'Manipulate the test or device collection as desired.
'iT  The current temperature point in the test
'iL  The current line pressure point in the test.
'iC  The current pressure cycle in the test.
'IP  The current pressure point in the test.
'cTest   The test class .
'cConfig  Configuration class that holds all active devices.
'
'************************************************
Function Fluke700CalibrationAdjust(iT, iL, iC, iP, cTest, cConfig)

Select Case cCOMPASS.CurrentTestStep
        Case 2000 ' Test Complete
                If cTest.TestPrsCycles <> 2 Then  'only support calbration with received and as left
                        Exit Function
                End If
```

```
                msg = "The test is complete.  The coefficients are active in module memory but "
                msg = msg & "have not been written to permanent storage. Would you like to activate "
                msg = msg & "the calibration?"

                If msgbox(msg,vbquestion + vbYesNo + vbSystemModal,"Activate Calibration") = vbNO Then
                        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"
                        Exit Function

               Else
                    For i = 1 To cConfig.DUTPrs.Count
                        Manuf = cConfig.DUTPrs(i).RangeMain.GetParent.Manufacturer
                        model = cConfig.DUTPrs(i).RangeMain.GetParent.Model
                        sn = cConfig.DUTPrs(i).RangeMain.GetParent.SN
                        cDebug.LogStatus "  DUT: " & i & ") Manufacturer=" & Manuf & ",SN=" & sn

                        If instr(Ucase(Manuf),"FLUKE") Then
                            cCOMPASS.StatusDisplay " DUT(" & SN & "): Writing calibration..."
                            cDebug.LogStatus " DUT " & i & "): Writing calibration"

                            If cConfig.DUTPrs(i).GetParamData(2) = 1 Then
                               msg = "The measured zero or full scale pressure is far away from the expected value. "
                               msg = msg & "Accuracy over temperature is at risk.  If you proceed "
                               msg = msg & "with this adjustment you must check accuracy at lab temperature "
                               msg = msg & "and 35 to 40 C before return to use.  "
                               msg = msg & "Proceed anyway?"
                               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
                                msg = "A message titled ""Set Module Calibration Date"" may be behind COMPASS."
                                msg = msg & vbcrlf & "Select this message from task bar when it appears."
                                 msgbox msg,vbExclamation+vbSystemModal,"WARNING"

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

                                 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
```

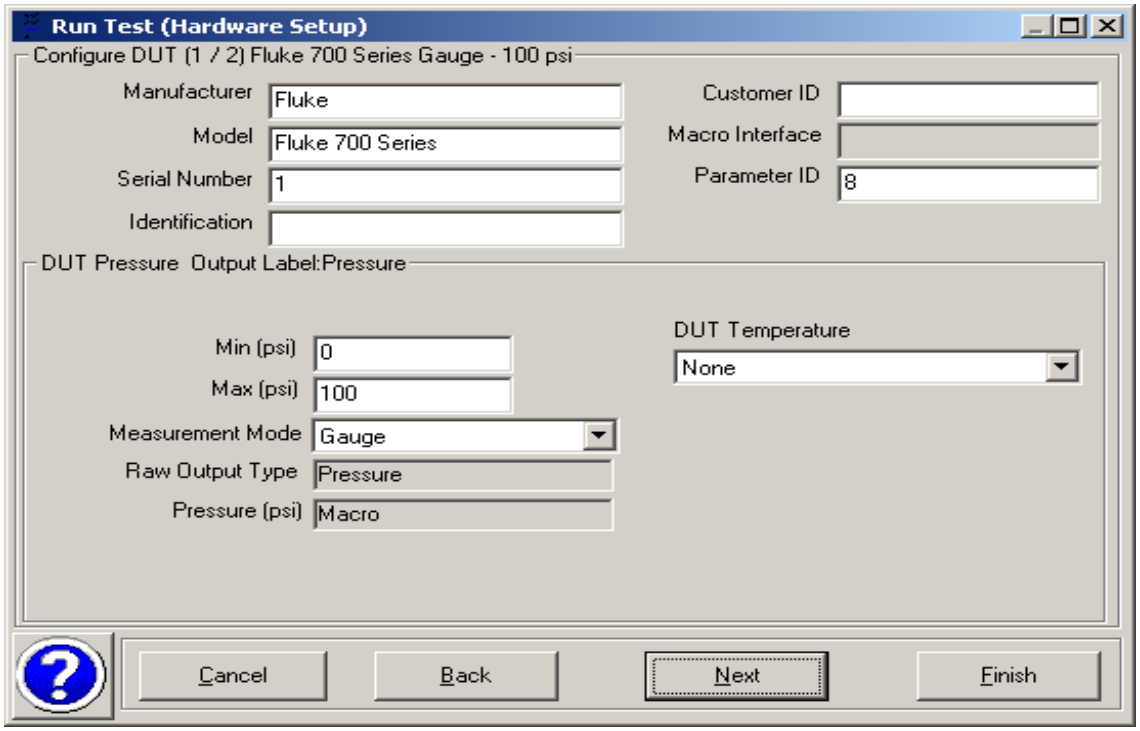
```
'This macro stores the active pressure and subtracts it from
'each reading. No internal changes to the module are made

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

        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 Fluke700_WaitForReady(1,120) = False Then Exit Function
                If cCOMPASS.SystemAbort Then Exit Function

                Fluke700_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 = Fluke700_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"

                             '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)

                             rtv = cConfig.DUTPrs(i).Obj1.GetZeroReading(CSng(newPrs)) 'reference pressure in psi
                             cDebug.LogStatus "   Return Value =" & rtv


                        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
                                             testpcnt = 0.4
                                     End If
                             End If

                             'Make sure that reference is close full scale
                             If Abs(DUTPrs-fs) > abs(testpcnt * (Max - Min)) Then
                                  cDebug.LogStatus "    Excessive difference for SN " & sn
                                  cDebug.LogStatus "    full scale= " & fs & ",DUTPrs=" & DUTPrs & ",pct=" & testpcnt
                                  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)

                             rtv = cConfig.DUTPrs(i).Obj1.GetFullScaleReading(CSng(newPrs))
                             cDebug.LogStatus "    Return Value =" & rtv

                             If rtv 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
                                     rtv = cConfig.DUTPrs(i).Obj1.AdjustCoefs
                             End If
                        End If

                    End If

                Next

            Next
```

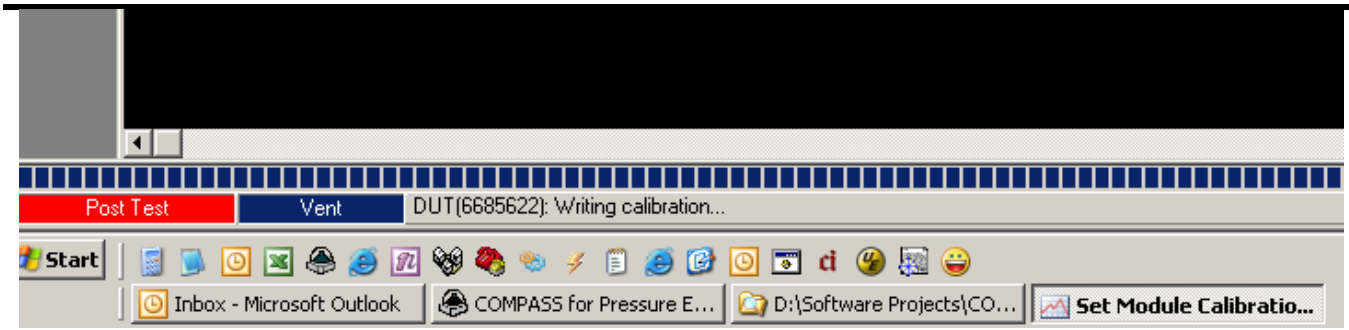
```
    'Apply rules to automatically re-adjust
    Exit Do
Loop

cDebug.LogStatus "Zero/Span cal Complete"
cCOMPASS.StatusDisplay ""

'need to zero the reading after calibration
Fluke700_RunZero

End Function
```

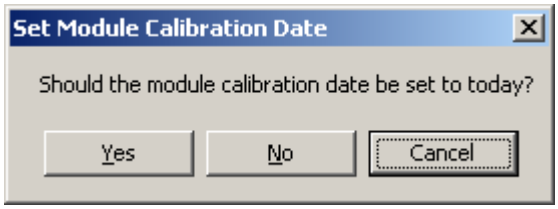
Running Test

1. Select appropriate DUT in terms of measurement mode (absolute and gauge) and pressure unit (psi and inH2O).
2. On the following screen, specify COM port in “Parameter ID”. Put any number in Serial Number field. The real Serial Number will be updated after reading from the module on the specified COM port.



3. At the end of calibration test, the use has the option to activate the calibration based on the As Left data. When writing the calibration to the module, user is asked “Should module calibration date to set to today”. Be aware that the screen may be behind COMPASS window. Click from task bar as show in the following picture to bring the window back to front.


Post Test
Vent
DUT(6685622): Writing calibration...
Start
Inbox - Microsoft Outlook
COMPASS for Pressure E...
D:\Software Projects\CO...
Set Module Calibratio...


Set Module Calibration Date
Should the module calibration date be set to today?
Yes
No
Cancel