**COMPASS for Flow**

**Test Macro to calculate real time expanded uncertainty with guardband**

**Make sure UserDefined8, UserDefined9, UserDefined10, StringData9 and StringData10 are saved to the data file in [Tools], <Options>, <Data in File>**

Copy and Paste all of the following text to a Test Macro in the COMPASS Macro Editor

*'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*'For help with the programming syntax,*

*'search for keyword 'vbscript' on the internet*

*'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*'Test Macros do not have a return value, but can save data as specified*

*'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 Flow cycle in the test*

*'IP The current Flow point in the test*

*'cTest The test class*

*'cConfig Configuration class that holds all active devices*

*'*

*'This test macro calculates uncertainty and a guardbanded tolerance for each setpoint in the test*

*'Calculation is done per Fluke Calibration Application Note -*

*'"Implementing ISO 17025 Measurement Uncertainty Requirements In Software"*

*'dated March 2012, document 1282496C*

*'Seach www.flukecal.com for "1282496C" to find this App Note*

*'*

*'Revision history at bottom of document*

*'Make sure UserDefined8 to UserDefined10, & StringData9 to 10 are saved to the data file in*

*'[Tools], <Options>, <Data In File>, and ensure that they are not being used in other macros or*

*'by regular COMPASS features (e.g. when using actual flow units COMPASS for Flow saves*

*'pressure and temperature to Userdefined1 and UserDefined2*

*'Can only use this macro in Advanced tests, and specify this as the "Test Event Macro"*

*'on the Data tab, and "Timed Macro Interval(ms)" needs to be 0 (zero)*

*'*

*'It is recommended to manually verify the calculated values before relying solely on this macro*

*'Contact Pressure/Flow Technical Support at flowsupport@flukecal.com with questions*

*'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**Function** **UncCalc** (iT, iL, iC, iP, cTest, cConfig)

*'Renaming column headers and set resolution of Meas. Unc. and GB Tol.*

**If** cCOMPASS.CurrentTestStep = 1100 **Then** *'Do this at start of a new pressure/flow cycle*

**For** i = 1 **To** cCOMPASS.DataCollection.Count

**With** cCOMPASS.DataCollection(i).DataInfoCollection("ID410008")

.Nomenclature = "Exp Unc" *'usually UserDefined8*

*'.Unit = "% rdg"*

*'.DataFormat = "0.000E-0" 'Alternate fixed way to specify resolution*

*'.DataFormat = "0.000" 'Alternate fixed way to specify resolution*

DUTrez = cConfig.DUTPrs(1).RangeMain.ResFinal

*' cDebug.LogStatus "DUT Resolution = " & DUTrez*

.DataFormat = 0.1^(DUTrez+3) *'3 additional digits of resolution for Exp Unc*

*' cDebug.LogStatus "Exp Unc Resolution = " & .DataFormat*

**End With**

**With** cCOMPASS.DataCollection(i).DataInfoCollection("ID410009")

.Nomenclature = "GB Tol" *'usually UserDefined9*

.DataFormat = 0.1^(DUTrez+3) *'3 additional digits of resolution for GB Tol*

*' cDebug.LogStatus "GB Tol Resolution = " & .DataFormat*

**End With**

cCOMPASS.DataCollection(i).DataInfoCollection("ID410010").Nomenclature = "TUR" *'usually UserDefined10*

cCOMPASS.DataCollection(i).DataInfoCollection("ID410508").Nomenclature = "Status GB" *'usually StringData8*

cCOMPASS.DataCollection(i).DataInfoCollection("ID410509").Nomenclature = "Status Tol" *'usually StringData9*

**Next**

**End If**

**If** cCompass.CurrentTestStep = 325 **Then** *'Averaging is completed but not yet written to data file*

*'Using 330 instead of 325 in above step is "Averaging is completed and logged in the data file" but doesn't update data grid*

**For** i = 1 **To** cConfig.DUTFlow.Count

DUTmodel = cConfig.DUTPrs(**CInt**(i)).RangeMain.GetParent.Model

DUTsn = cConfig.DUTPrs(**CInt**(i)).RangeMain.GetParent.SN

*' cDebug.LogStatus "DUT Model " & DUTmodel & " SN " & DUTsn*

dUnit = cCOMPASS.cConfig.DUTPrs(i).RangeMain.UnitFinal

*' cDebug.LogStatus "DUT Units = " & dUnit*

dUnitText = cCOMPASS.cConfig.DUTPrs(i).RangeMain.UnitFinalText

*' cDebug.LogStatus "DUT Unit Text = " & dUnitText*

*'Get inputs for unc calc*

UDut = cCOMPASS.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).Tolerance

*' cDebug.LogStatus "DUT Tolerance = " & UDut*

DutFlow = cCOMPASS.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).DUTFlow

*' cDebug.LogStatus "DUT Flow = " & DutFlow*

DutRes = cConfig.DUTPrs(i).RangeMain.ResFinal

*' cDebug.LogStatus "DUT Res Rtv= " & DutRes*

Res = 0.1^DutRes

cDebug.LogStatus "DUT Res = " & Res

DUTStd = cCOMPASS.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).DUTStnDev

cDebug.LogStatus "DUT StdDev = " & DUTStd

**If** DUTStd < 0 **Then** *'If DUT StDev <=0 set to 0*

DUTStd = 0

cDebug.LogStatus "DUT StdDev (Corrected if <=0) = " & DUTStd & " " & dUnitText

**End If**

DUTSamples=cCOMPASS.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).DUTCounts

cDebug.LogStatus "DUT Samples = " & DUTSamples

RefF = cCOMPASS.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).RefPressure *'Use RefPressure*

RefF\_Unit = cCOMPASS.cConfig.RefPrs(1).RangeMain.UnitFinal *'number of Ref Pressure unit*

RefFunitTxt = cCOMPASS.cConfig.RefPrs(1).RangeMain.UnitFinalText

*' cDebug.LogStatus "Reference Flow = " & RefF & " " & RefFunitTxt*

*'Below is the normal way to get Reference Uncertainty from COMPASS Support Device setup file*

URef = cCOMPASS.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).RefTolerance

URefCP = URef *'Assign to URefCP in case we get Ref Unc by another method*

*' cDebug.LogStatus "Reference Tolerance = " & UrefCP & " " & RefFunitTxt*

*'Uncomment the 3 following lines if you want to calculate Reference Uncertainty from the Global Macro named UncRefByRange*

*' URefRange = UncRefByRange(RefF, RefFunit, RefFunitTxt)*

*' URef = URefRange*

*' cDebug.LogStatus "Ref Unc from Range Macro: " & URefRange & " " & RefFunitTxt*

**If** Uref < 0 **Then** *'Manual entry Ref devices report tolerance as -9999, convert to a really high number*

Uref = UDut \* 1000000

cDebug.LogStatus "Ref Tol (Corrected if <=0) = " & Uref & " " & RefFunitTxt

**End If**

cDebug.LogStatus "Reference Tolerance = " & Uref & " " & RefFunitTxt

Conf = 2 *'Confidence for U1 calculation*

cDebug.LogStatus "Confidence = " & Conf

k=2

cDebug.LogStatus "k = " & k

*'Degrees of Freedom for S1 calculation. Comment out the lines after F=1 if you just want F=1*

F = 1 *'Use F=1 if number of DUT samples is 0 (zero) or less*

**If** DUTSamples > 9 **Then**

F = 1.0

**ElseIf** DUTSamples > 7 **Then**

F = 1.2

**ElseIf** DUTSamples > 4 **Then**

F = 1.3

**ElseIf** DUTSamples > 3 **Then**

F = 1.7

**ElseIf** DUTSamples > 2 **Then**

F = 2.3

**ElseIf** DUTSamples > 0 **Then**

F = 7.0

**End If**

cDebug.LogStatus "F = " & F & ", Related to # DUT samples, lower is better"

*'Calculate uncertainty*

U1 = URef / Conf *'Convert Ref Unc to k=1*

cDebug.LogStatus "U1 = " & U1 & " " & dUnitText & ", Uref/Confidence (Confidence usually = 2)"

S1 = (DUTStd / (DUTSamples^0.5))\*F *'Calculate standard error of the mean @ k=1 and multiply by Students T factor*

cDebug.LogStatus "S1 = " & S1 & " " & dUnitText & ", F \* DUT Std Dev / SQRT(DUT Samples)"

S2 = (Res\*0.5) / (3^0.5) *'Convert DUT resolution to k=1*

cDebug.LogStatus "S2 = " & S2 & " " & dUnitText & ", Convert DUT resolution to k=2"

U2 = (((S1)^2 + (S2)^2))^0.5

cDebug.LogStatus "U2 = " & U2 & " " & dUnitText & ", RSS of S1 and S2"

Unc = (((U1)^2 + (U2)^2)^0.5) *'Combined standard uncertainty, k=1*

ExpUnc = Unc \* k *'Combined & expanded, k=2*

cDebug.LogStatus "Unc = " & Unc & " " & dUnitText

*'Test Uncertainty Ratio (TUR) with 1 digit of resolution*

TUR = formatnumber(UDut/ExpUnc,1)

DUTErr = cCOMPASS.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).DUTDiff

cDebug.LogStatus "\*" *'blank line in macro spy window*

cDebug.LogStatus "\* DUT Model " & DUTmodel & ", SN " & DUTsn

cDebug.LogStatus "\* DUT Flow = " & DutFlow & " " & dUnitText

cDebug.LogStatus "\* Ref Flow = " & RefF & " " & RefFunitTxt

cDebug.LogStatus "\* DUT-Ref = " & DUTErr & " " & dUnitText

cDebug.LogStatus "\* Abs(DUT-Ref) = " & Abs(DUTErr) & " " & dUnitText

cDebug.LogStatus "\* DUT Tolerance = " & UDut & " " & dUnitText

cDebug.LogStatus "\* ExpUnc = " & ExpUnc & " " & dUnitText

*'Guardband calculation and check*

**If** (ExpUnc)^2 > (UDut)^2 **Then**

msgbox "Expanded Uncertainty is greater than DUT Tolerance"

GB\_Tol = 0 *'Set guardbanded tolerance to zero so it will fail if uncertainty is greater than DUT tolerance*

**Else**

GB\_Tol = ((UDut)^2 - (ExpUnc)^2)^0.5

cDebug.LogStatus "\* GB Tol = " & GB\_Tol

cDebug.LogStatus "\* TUR = " & TUR

**End If**

**If** GB\_Tol > Abs(DUTErr)**Then**

PF = "GB Pass"

**Else**

PF = "GB Fail"

*'Uncomment the next line if you want the test to stop if the guardbanded tolerance fails*

msgbox "Tolerance with Guardband failed. Press [OK], then the back button |<< if you want to repeat the last point. Press the back button until the window shows the point that you want to repeat."

**End If**

cDebug.LogStatus "\* GB Tolerance Pass/Fail = " & PF

**If** UDut > Abs(DUTErr) **Then**

Status = "Pass"

**Else**

Status = "Fail"

**End If**

cDebug.LogStatus "\* Regular Tolerance Pass/Fail = " & Status

cCompass.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).UserDefined8 = **CDbl**(ExpUnc) *'Expanded Uncertainty to UserDefined8*

cCompass.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).UserDefined9 = **CDbl**(GB\_Tol) *'Guardbanded Tolerance to UserDefined9*

cCompass.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).UserDefined10 = TUR *'TUR to UserDefined10*

cCompass.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).StringData9 = PF *'GB Pass/Fail to StringData9*

cCompass.DataCollection(i).DataPointRef (**CInt**(iT),**CInt**(iL),**CInt**(iC),**CInt**(iP)).StringData10 = Status *'Pass/Fail based on DUT-REF vs Tol to StringData10*

cDebug.LogStatus "\*" *'blank line in macro spy window*

**Next**

**End If**

**End Function**

*'Author: Kyle Clark*

*'Rev 3, 28-August-2015 added multiple DUT capability & TUR*

*'Rev 4, 02-Sept-2015 added units to UserDefined 1 to 3 (Changed to User8 to 10 on 05-Apr-2016)*

*'Rev 5, 23-Sept-2015 Changed 330 in "If cCompass.CurrentTestStep = 330" to 325 because Uncertainty*

*'Parameters were not updating in data grid*

*'Rev 6, 07-Mar-2016 Added more comments in debug statements for more info on what S1, S2,... are.*

*'Set TUR resolution to 1 digit*

*'Rev 7, 08-Mar-2016 Added resolution for Unc. and GB Tol. in the first few lines of code*

*'Added CInt(iC) for multiple Flow Cycle support (PresCyclIndex)*

*'Rev 8, 05-Apr-2016 Changed to UserDefined 8 to 10 and StringData9 and 10*

*'Moved "Tolerance with Guardband failed" message box to the correct if-then-else statement*

*'Rev 12, 13-Apr-2022 Changed i to 1 when getting Ref Unit and Ref Unit Text (highlighted in green in above text).*

*'The macro only supports 1 reference in a test.*

*'Changed Rev number To 12 To match the Rev number of similar COMPASS For Pressure macro.*

*'Rev 14, 20-May-2022 Added CDbl assignments to some values to ensure that the values are double precision numbers and maintain their values when using commas as the decimal separator.*

*'Skipped Rev 13 To match COMPASS For Pressure Rev number.*