o

If you’re using a COM port other than COM1, then change the default COM port here in the Interface macro named “Fluke700P\_GetPressure”. Save the changes.



If this line is not already commented out (by adding an apostrophe at the beginning of it), then comment it out. Make sure it turns green. Save the changes.



If you are using the example or sample 750P module DUT setup in COMPASS, then uncheck this box in the [Database], <Setup> menu. If you don’t uncheck it, then any changes that you make to the macros related to it will revert automatically!

Interface macro “Fluke700P\_GetPressure” edited 2018-Dec-04 by Kyle Clark with the above changes.

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

*'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** **Fluke700P\_GetPressure**(cCalc, cParent, fTarget1, bChnlSt, cConfig)

 **On** **Error** **Resume** **Next**

 **If** **CheckFluke700PObj**(cCalc) **Then**

 prs = cCalc.Obj1.GetPressureP()

 *'cDebug.LogStatus "700P Pressure: " & prs*

 *'cDebug.LogStatus "Error Code:" & cCalc.Obj1.ErrorCode*

 dUnit = cCalc.RangeMain.UnitFinal

 newPrs = cCOMPASS.UnitConversion(**CDbl**(prs), **CInt**(dUnit), 9, 0)

 zeroPrs = cCalc.GetParamData(0)

 **Fluke700P\_GetPressure** = newPrs - zeroPrs

 **End If**

**End Function**

*'''''''''''''''''''''''''''''''''''''''''''''''''''*

*'Make sure that the Fluke700P object is set.*

*'This Function Is Not called directly*

*'by COMPASS. It is used by the other Fluke700P*

*'related functions.*

*'''''''''''''''''''''''''''''''''''''''''''''''''''*

**Function** **CheckFluke700PObj**(cCalc)

 **On** **Error** **Resume** **Next**

 defaultPort = 1 *'If you're not using RS-232 COM1 then change to defaultPort = n where n Is the default COM port you want To use*

 *'*cDebug.LogStatus "Default Port:" & defaultPort

 **If** cCOMPASS.SystemAbort **Then**

 *'If cCalc.Obj1 Is Nothing Then Exit Function*

 **If** cCalc.GetParamData(5) = 0 **Then**

 temp = cCalc.Obj1.Errordescription

 cCOMPASS.**StatusDisplay** "Shutting down 700 PCK interface..."

 cdebug.LogStatus "Shutting down 700 PCK interface..."

 cCalc.Obj1.Disconnect

 cCOMPASS.SystemAbort = **False**

 cCOMPASS.**TimeDelay** 5

 cCOMPASS.SystemAbort = **True**

 **Set** cCalc.Obj1 = **Nothing**

 cDebug.LogStatus "PCK Object Error Description: " & temp

 cDebug.LogStatus "Disconnected Fluke Interface. Err:" & Err

 cCalc.SetParamData 5,1

 **End If**

 **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 "COM PortInfo from macro:" & defaultPort

 cDebug.LogStatus "COM PortInfo from Parameter ID:" & PortInfo

 **If** PortInfo > 0 And PortInfo < 255 **Then**

 Port = PortInfo

 cDebug.LogStatus "Port, after check for 255 > ParamID > 0 :" & Port

 **Else**

 Port = defaultPort

 cDebug.LogStatus "Port, if no Parameter ID (from defaultPort):" & Port

 **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*

 DUTID = 1

 *'Port = PortInfo '(Don't do this. It sets Port back to PortInfo from DUT #1 ParamID. If DUT #1 ParamID is blank this will be wrong.)*

 cCOMPASS.**StatusDisplay** "First DUT. Default COM port in macro is " & PortInfo & ". COM" & Port & " will be used for DUT#1"

 cDebug.LogStatus "First DUT," & " SN " & cCalc.RangeMain.GetParent.SN & " on COM " & Port

 **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 = PortInfo *'This is correct. COM port for more than 1 DUT has to come from ParamID*

 DUTID = i

 cCOMPASS.**StatusDisplay** "DUT#" & DUTID & ". ParamID value is " & Port & ". COM" & Port & " will be used for DUT# " & DUTID

 cDebug.LogStatus " DUT # " & DUTID & ". ParamID = " & PortInfo & ", COM Port = " & Port

 **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("", "Fluke.Calibration.PCKsupport")

 *'\*\*\*\*Specify Port to Communicate\*\*\*\*\**

 Model = cCalc.RangeMain.GetParent.Model

 SN = cCalc.RangeMain.GetParent.SN

 cCOMPASS.**StatusDisplay** "Model:" & Model & "(SN" & SN & ") Initializing " & Port

 *'No property by the name ChangeComPort in the support library*

 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.Connect **CInt**(Port), **CInt**(rtv)

 cDebug.LogStatus "Fluke 700 Connection"

 cDebug.LogStatus "RS232 Port:" & Port

 cDebug.LogStatus "Error Code:" & cObj.ErrorCode

 cDebug.LogStatus "Error Description:" & cObj.ErrorDescription

 **If** cObj.ErrorCode <> 0 **Then** *'len(temp)> 5 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 & "."

 msg = msg & vbcrlf & "ERR:" & vbcrlf & temp

 MsgBox msg, vbCritical, "Error"

 cCalc.SetParamData 1, 1 *'Flag that device failed*

 cCOMPASS.SystemAbort = **True**

 **Exit** **Function**

 **End If**

 **Set** cCalc.Obj1 = cObj

 *'Update Serial Number of device.*

 temp = cCalc.Obj1.SerialNumber

 **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.Model

 **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 = cCalc.RangeMain.GetParent.Model

 cCOMPASS.DataCollection(DUTID).DUT.SerialNumber = cCalc.RangeMain.GetParent.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**

 **CheckFluke700PObj** = **True**

**End Function**