How to Create a Beamex MC5 Support Device with COMPASS for Pressure

Summary

This article describes the setup requirements to create a Beamex MC5 support device for pressure measurement using **COMPASS for Pressure**. The setup can be used as DUT or reference in any automated pressure test. The support device 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 Support Device setup named "Beamex MC5".

Article Topics

- How to create a Beamex MC5 Support Device with a pressure output.
- How to use a ReplyParser Macro to manipulate a response.
- How to set up device with byte data command.
- How to interpret a response using an IEEE-754 single precision floating point number.

See Also

N/A

Requirements

The following items are required to implement this setup.

- **COMPASS for Pressure Basic** or **Enhanced**.
- PC with an available COM port.

Beamex MC5 Setup

To simplify remote communication, manually setup the Beamex MC5 to measure pressure in the target unit and measurement mode. In this article kPa absolute is used for measurement. Change the MC5 display to include pressure in the top window frame.

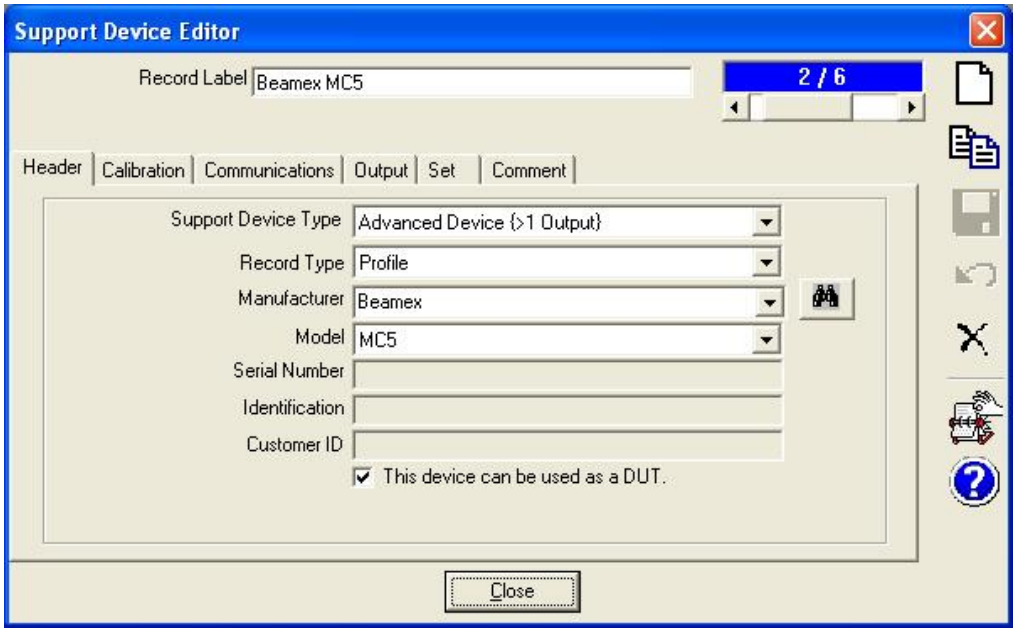
Connect the MC5 to a COM port of the computer. Setup the MC5 RS232 communication parameters: baud rate 38400, parity none, data bits 8, stop bits 1.

COMPASS Support Device Setup

Follow the steps below to create a Beamex MC5 Support Device. Alternatively, merge the Support Device information from the Example database as described in the Summary section of this article.

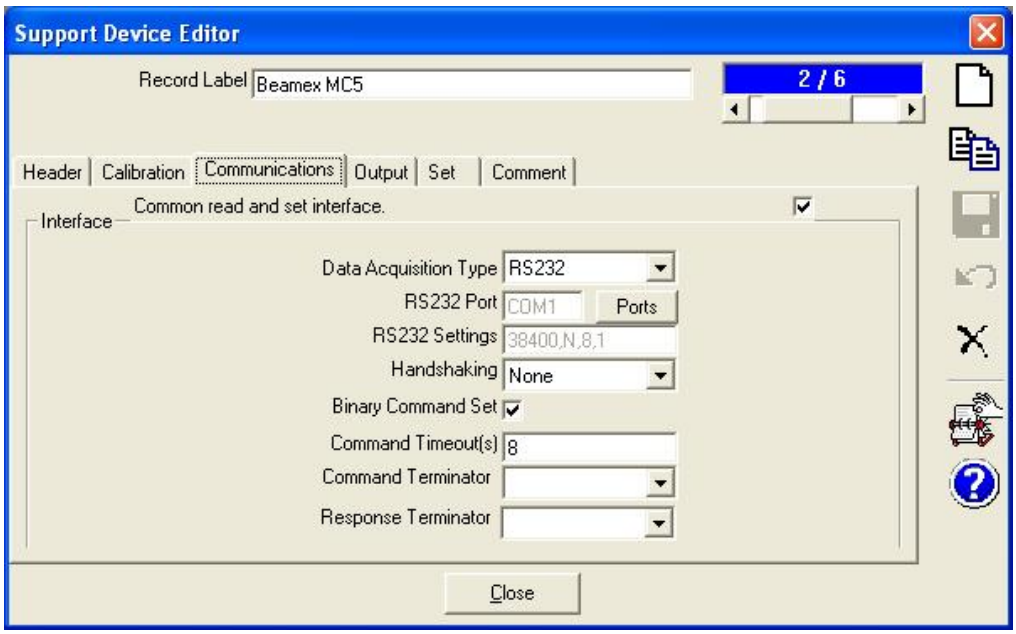
1. Run **COMPASS** and select the **[Setup][Support Device]** menu choice. Create a new Support Device by pressing the **[New]** button on the top of the toolbar.

2. Enter the basic setup information for the Support Device. Check the **<This device can be used as a DUT>** option so that the Support Device is available as both a DUT and a reference.



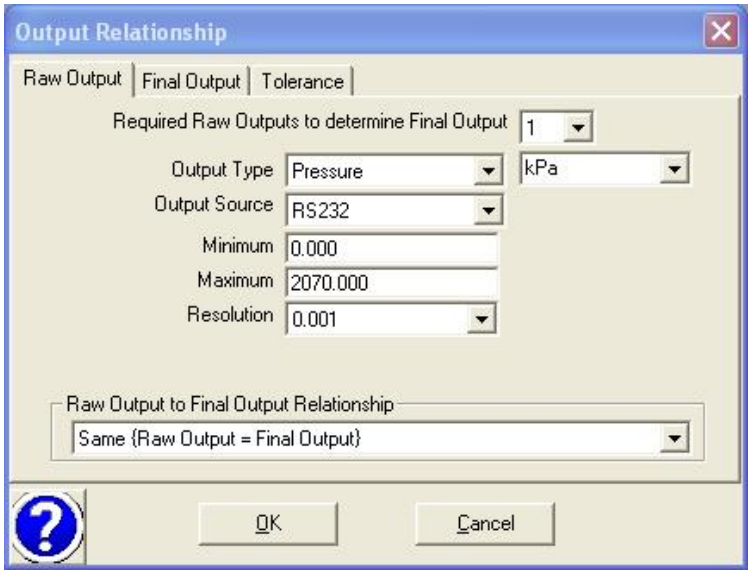
3. No specific settings are required on the **[Calibration]** tab. This tab is provided to maintain calibration information or to associate a specific test with the DUT.

4. Select **<RS232>** as the **<Data Acquisition Type>** on the **[Communications]** tab. Click **[Ports]** to select the appropriate RS232 port and settings. The default setting is "38400,N,8,1" for Beamex MC5. Choose "None" for **<Handshaking>**, check **<Binary Command Set>** since MC5 uses byte messages, and use "8" as **<Command Timeout(s)>**. The fields **<Command Terminator>** and **<Response Terminator>** must be empty. Most instruments that use binary communication do not have a specific character termination method.

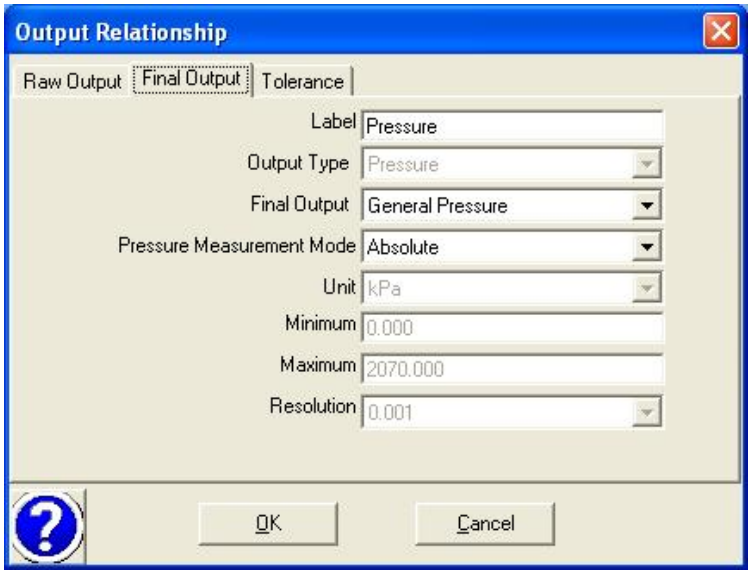


5. On the **[Output]** tab, press the **[Edit Output]** button to edit the default device output. A default output is always created for new devices.

6. Select the **[Raw Output]** selections defined in the figure below. Enter the appropriate range and resolution information. The unit of measure selection must correspond to the unit of measure defined on the front panel of the MC5. Note that this device is defined as a profile so the actual range information can be changed while initializing a test. Select "RS232" as **<Output Source>** and "Same" as **<Raw Output to Final Output Relationship>**.

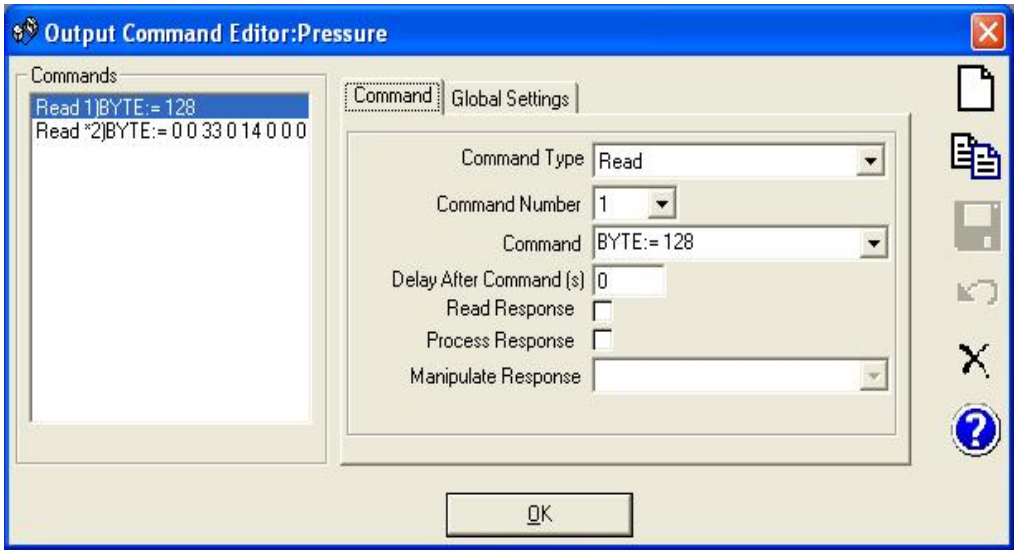


7. Since **<Same>** is selected as the output relationship, many of the selections on the **[Final Output]** tab are disabled. Use "General Pressure" as **<Final Output>**, so that this device can be used as either DUT or reference.



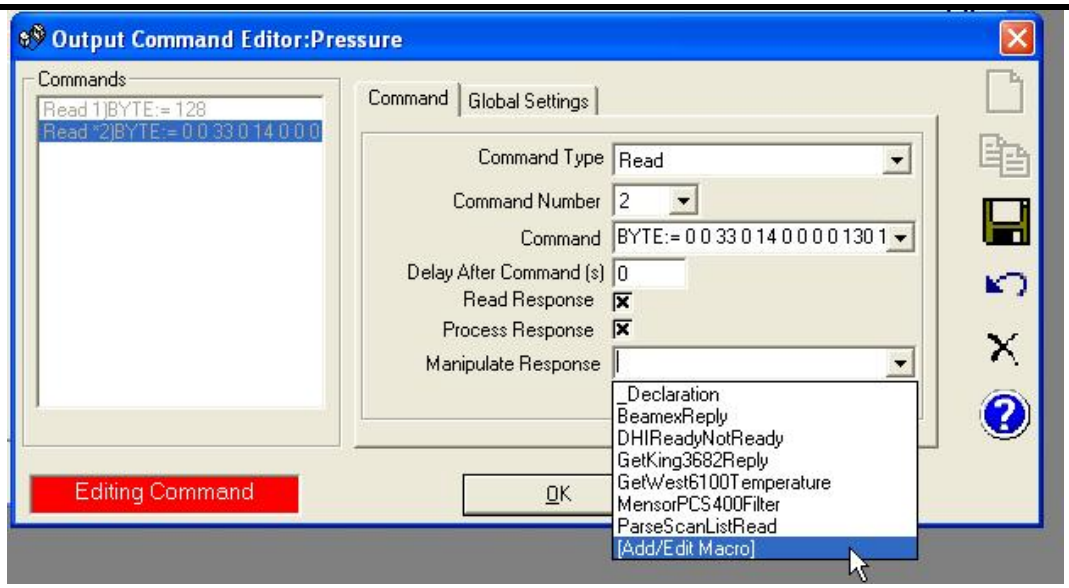
8. Use the **[Tolerance]** tab to define the DUT tolerance. Press **[OK]** to return to the **Support Device Editor**.

9. Click **[Edit Commands]** to get to **[Output Command Editor]**. Create the first “Read” command “BYTE:= 128”. “BYTE:=” is used to send discrete data bytes in **COMPASS**. The byte value in the command should be a decimal number not Hex. Hex 0x80 is 128 in decimal. This stands for “STX – request to send” in the MC5 communications protocol. This command is used to establish the connection before the pressure measurement command is sent. Uncheck **<Read Response>** and **<Process Response>** since this command does not reply. Do not use “Initialize” as the **<Command Type>**. STX must be sent prior to each pressure measurement command. An initialize command is sent only once at the beginning of the test. Save the command by pressing the **[Save]** option.



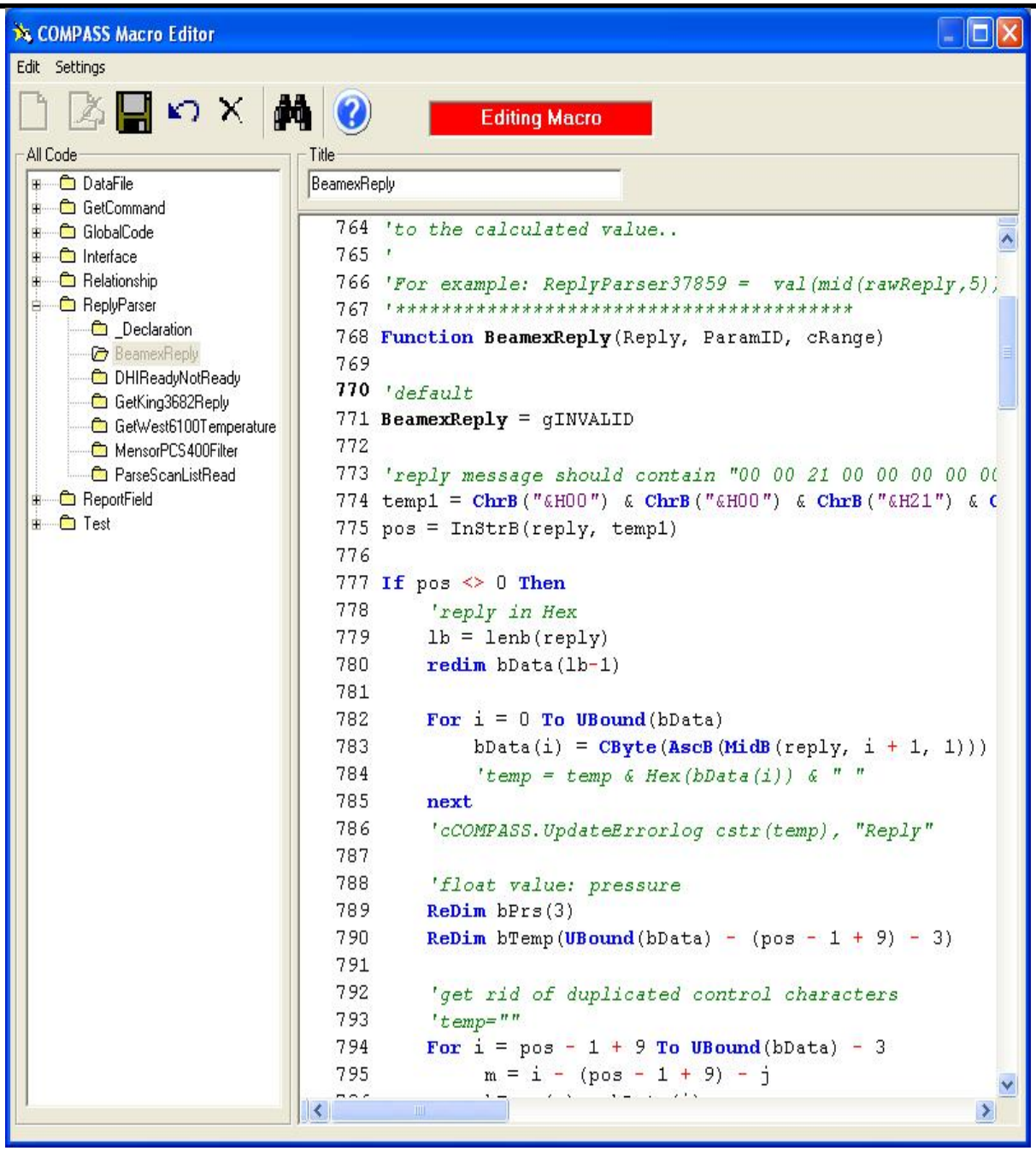
10. Press the **[New]** icon to create the second “Read” command “BYTE:= 0 0 33 0 14 0 0 0 0 130 131 102”. The command is “0x00 0x00 0x21 0x00 0x0E 0x00 0x00 0x00 0x00 0x82 0x83 0x66” in Hex. In the MC5 communications protocol, the message (or **COMPASS** command) consists of a message frame, “0x00 0x00 0x21 0x00 0x0E 0x00 0x00 0x00 0x00” and end of transmission: “0x82 0x83 0x66”. 0x21 is the ID of the message, which is “MSG_READING”. 0x0E, or decimal 14, is the byte size of the response expected. The message ends with 0x82 (DLE – data link enabled), 0x83 (ETX – End of transmission) and the BCC check-sum byte, which is 0x66 in this message. The MC5 communication protocol uses Cyclic Redundancy Check algorithm to calculate the check-sum BCC byte.

    Check **<Read Response>** since this command has a response, and also check **<Process Response>** since the response of this command is used as the output value of this support device. The response of this command contains a 14 byte message and end of communication bytes (DLE, ETX and BCC). Bytes 2 to 5 of the response are the pressure reading in IEEE754 single precision floating point format.



A ReplyParser Macro is required to convert from the IEEE754 floating point format to a number accessible by **COMPASS**. Select **[Add/Edit Macro]** from the **<Manipulate Response>** list to activate the **<COMPASS Macro Editor>**. The list contains the complete list of ReplyParser Macros in the active **COMPASS** database.

11. Create a new ReplyParser Macro using the **COMPASS Macro Editor**. Copy the contents of the **BeamexReply ReplyParser Macro** and paste the text into the macro editor overwriting the default macro text in the newly created macro. Press the **[Save]** button to save the macro after the edits are complete. Close the macro editor to return to the **Output Command Editor**. Many of the support functions in this macro are reusable and can copied to the **GlobalCode** section. This would allow other macros to perform IEEE754 manipulation.



BeamexReply ReplyParser Macro

```
'****************************************
'This Function must return the fully formatted response
'of a device.  The output is used as the Raw output
'in the relationship determination of the Final Output.
'
'Reply     :Raw unmanipulated response of a device.
'ParamID   :Parameter ID of the device
```

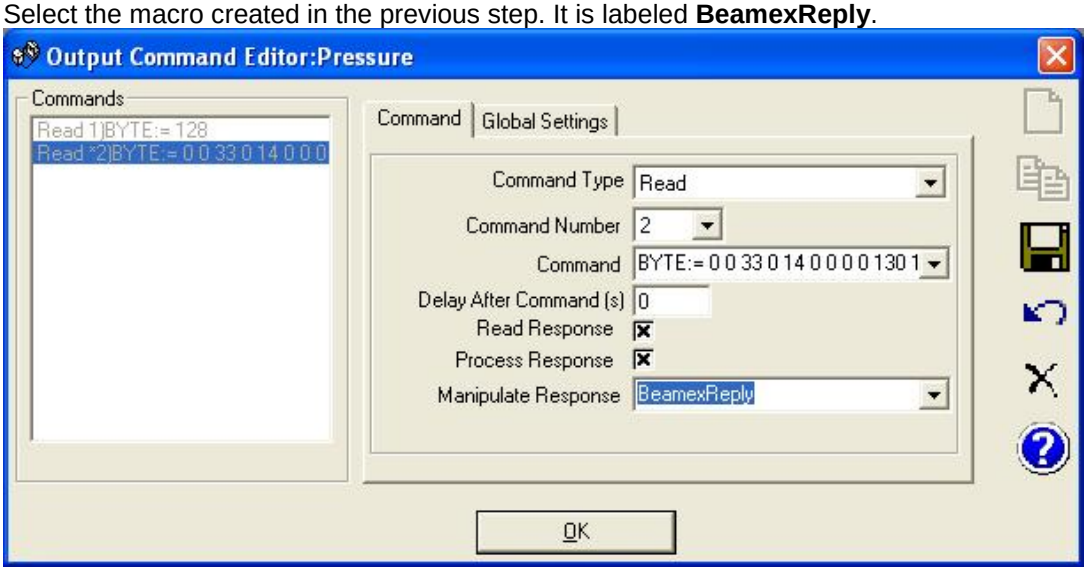
```
'cRange    :Range class that the output applies to.
'
'The value is returned by setting the function name =
'to the calculated value..
'
'For example: ReplyParser37859 =  val(mid(rawReply,5))
'**************************************
Function BeamexReply(Reply, ParamID, cRange)

'default
BeamexReply = gINVALID

'reply contains "0x00 0x00 0x21 0x00 0x00 0x00 0x00 0x00 0x0E" at the beginning
temp1 = ChrB("&H00") & ChrB("&H00") & ChrB("&H21") & ChrB("&H00") & ChrB("&H00")
temp1 = temp1 & ChrB("&H00") & ChrB("&H00") & ChrB("&H00") & ChrB("&H0E")
pos = InStrB(reply, temp1)

If pos <> 0 Then
    'reply in Hex
    lb = lenb(reply)
    redim bData(lb-1)

    For i = 0 To UBound(bData)
        bData(i) = CByte(AscB(MidB(reply, i + 1, 1)))
        'temp = temp & Hex(bData(i)) & " "
    Next
    'cCOMPASS.UpdateErrorlog cstr(temp), "Reply"

    'float value: pressure
    ReDim bPrs(3)
    ReDim bTemp(UBound(bData) - (pos - 1 + 9) - 3)

    'get rid of duplicated control characters
    'temp=""
    For i = pos - 1 + 9 To UBound(bData) - 3
         m = i - (pos - 1 + 9) - j
         bTemp(m) = bData(i)
         'temp = temp & " " & Hex(bTemp(m))

        If bTemp(m) = 128 Or bTemp(m) = 129 or bTemp(m) = 130 Or bTemp(m) = 131 then
            j = j + 1
            i = i + 1
        End If
    Next
    'cCOMPASS.UpdateErrorlog cstr(temp), "No Control Char"

    'get first 4 bytes and switch position to get the byte array
    bPrs(0) = bTemp(3)
    bPrs(1) = bTemp(2)
    bPrs(2) = bTemp(1)
    bPrs(3) = bTemp(0)
    'cCOMPASS.UpdateErrorlog Hex(bPrs(0)) & " " & Hex(bPrs(1)) & " " & Hex(bPrs(2)) &
" " & Hex(bPrs(3)), "Float"

    BeamexReply = IEEE754_Float(bPrs)
    'cCOMPASS.UpdateErrorlog cstr(BeamexReply), "Pressure"
End If

End Function
```

```
'////////////////////////////////////////////////////////////////////////////////
'Passed byte array of ASCII values formatted with the IEEE754 Floating
'Point format,  return a single precision number
'
'bData must be a byte array as MSB - LSB  formatted in single
'
'The IEEE single precision floating point standard representation
'requires a 32 bit word, which may be represented as numbered from
'0 to 31, left to right. The first bit is the sign bit, S, the next
'eight bits are the exponent bits, 'E', and the final 23 bits are
'the fraction 'F':
'
'  S EEEEEEEE FFFFFFFFFFFFFFFFFFFFFFF
'  0 1      8 9                     31
'
'The value V represented by the word may be determined as follows:
'
'
'If E=255 and F is nonzero, then V=NaN ("Not a number")
'If E=255 and F is zero and S is 1, then V=-Infinity
'If E=255 and F is zero and S is 0, then V=Infinity
'If 0<E<255 then V=(-1)**S * 2 ** (E-127) * (1.F) where "1.F" is intended to
'  represent the binary number created by prefixing F with an implicit
'  leading 1 and a binary point.
'If E=0 and F is nonzero, then V=(-1)**S * 2 ** (-126) * (0.F) These are
"unnormalized" values.
'If E=0 and F is zero and S is 1, then V=-0
'If E=0 and F is zero and S is 0, then V=0
Function IEEE754_Float(bData)

On Error Resume Next

if UBound(bData) = 3 then
        'Must be Single
        ReDim AllBytes(31)
        ReDim boolDat(7)

        inc = 0
        For i = 0 To 3
            NumberToBoolArray bData(i), boolDat

            For j = 0 To 7
                AllBytes(inc) = boolDat(j)
                inc = inc + 1
            Next
        Next

        'sign
        If AllBytes(0) = 1 then
            S = -1
        Else
            S = 1
        End If

        E = CalcNumber(1, 8, AllBytes,False)

        F = CalcNumber(9, 31, AllBytes, True)

        b = 127  'Single Float BIAS
```

```
else
        'This is Garbage
        IEEE754_Float = gINVALID

        Exit Function
end if


If E = 0 And F = 0 Then
    rtv = 0

ElseIf E = 255 And F <> 0 Then
    rtv = gINVALID

ElseIf E = 255 And F = 0 And S = -1 Then
    'NEGATIVE infinity
    'rtv = -999999999
    rtv = gINVALID

ElseIf E = 255 And F = 0 And S = 1 Then
    'Positive infinity
    'rtv = 999999999
    rtv = gINVALID

ElseIf E > 0 And E < 255 Then
    rtv = S * 2 ^ (E - b) * (1 + F)

ElseIf E = 0 And F <> 0 Then
    rtv = S * 2 ^ (b + 1) * F

End If

IEEE754_Float = rtv


End Function

'////////////////////////////////////////////////////
'Passed an integer, return a boolean array containing
'the TRUE false States of the individual bits in the
'value.  The order is MSB to LSB
'
'inVal   : Number 0 - 255
'BoolData : array 0 to 7  2^7 -- 2^0
'////////////////////////////////////////////////////
Sub NumberToBoolArray(inval, boolDat)

max = 7

For i = 0 To max
    If (inval And 2 ^ i) <> 0 Then
        boolDat(max - i) = 1
    Else
        boolDat(max - i) = 0
    End If
Next

End Sub
```

```
'////////////////////////////////////////////////////////////////////////////////
'Bassed an array of bytes boolean values representing bytes from MSB to
'LSB.  Start at the bStart position and stop at the bStop Position
'and determine the value by a  bitwise operation.  It is assumed that
'bStop > bStart
'
'When FRACTION is true compute the value bitwise fractional value
'still MSB-LSB but... 1/2^1, 1/2^2, 1/2^3 ect...
'
'////////////////////////////////////////////////////////////////////////////////
Function CalcNumber(bStart, bStop, boolArray, FRACTION )

If bStop = 0 And bStart = 0 Then
    rtv = gINVALID

ElseIf FRACTION = False Then
    'Assume B stop is the LSB
    ID = bStop - bStart
    For i = bStart To bStop
        If boolArray(i) Then
            rtv = rtv Or 2 ^ ID
        End If
        ID = ID - 1
    Next


Else 'If bStop < bStart Then
    'Assume B stop is the LSB in fraction terms..
    ID = 1
    For i = bStart To bStop
        If boolArray(i) Then
            rtv = rtv + 1 / (2 ^ ID)
        End If
        ID = ID + 1
    Next


End If

CalcNumber = rtv

End Function
```

12. Select the macro created in the previous step. It is labeled **BeamexReply**.



13. There are no other unique selections required for this device. Press **[Save]** to activate all edits.
14. Run the support device as a DUT or a pressure reference.