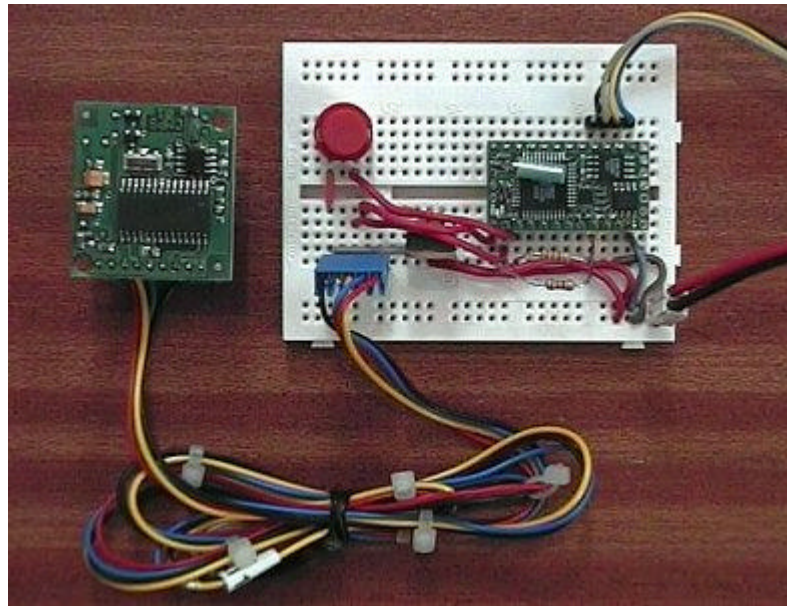


Using the I2C Bus for Connecting the CMPS01/03 to the BX-24

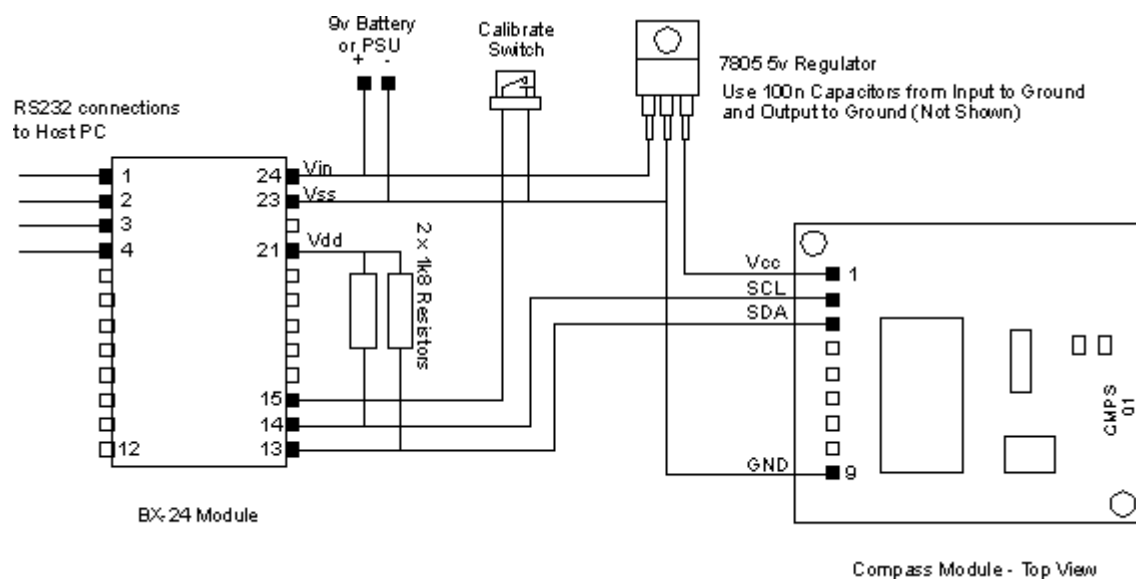


Introduction

The compass module uses either a PWM output or the I2C bus for communications. This example uses the I2C Bus. The SDA (data) and SCL (clock) lines are connected to pins 13 and 14 on the BX-24 and a push button for calibration connected to pin 15. These are arbitrarily chosen, you can use any pins you choose to. As the BX-24 does not have any I2C commands, the example provided here uses a combination of bit bashing and the SHIFTIN and SHIFTOUT commands to construct the I2C functions.

The BX-24 internal 5v regulator is not suitable for powering much external circuitry. I therefore recommend you use a separate 5v regulator.

Circuit Schematic for connecting the BX-24 to the CMPS01/CMPS03 Compass Module using the I2C bus



Note the switch connecting pin 15 on the BX-24 to Ground . This is all that is needed for [calibration](#) of Rev7 Firmware via the I2C bus. The following program contains the four lines of code needed to

do this.

Software

The sample code below displays the compass bearing as Degrees (0-359) in a Monitor window on the PC.

It can be downloaded [here](#)

```

' *****
' **
' **          I2C Routines for the BX-24          **
' **    to demonstrate the use of Compass module  **
' **
' **          Copyright 2002 - Devantech Ltd      **
' **    Commercial use of this software is prohibited **
' **    Private and educational use only is permitted **
' **
' **          Written by Gerald Coe - February 2002 **
' **
' *****

```

```

Const SCL As Byte = 14      ' I2C clock - choose any pins you
wish for SCL and SDA
Const SDA As Byte = 13      ' I2C data
Const Calibrate As Byte = 15 ' Calibrate push button

```

```

Const CmdReg As Byte = 0      ' SRF08 command register
Const Compass As Byte = &Hc0  ' Compass module is at address
0Xc0
Const BearingReg As Byte = 2  ' Bearing is in registers 2 & 3
(High:Low)
Const CalReg As Byte = 15     ' Calibration Register
Const CalCmd As Byte = 255    ' Calibration Command

```

```

Dim I2cAck As Boolean        ' Acknowledge flag

```

```

Sub Main()

```

```

  Dim Bearing As New UnsignedInteger

```

```

    Call PutPin(SCL, bxOutputHigh)
    Call PutPin(SDA, bxOutputHigh)
    Call PutPin(Calibrate, bxInputPullup)

```

```

Do

```

```

  Bearing = I2cWordRead(Compass, BearingReg)\10      ' Read Bearing
and convert to 0-359 degrees
  debug.Print "Bearing = "; CStr(Bearing)

```

```

' The following four lines of code are all you need if you wish to
calibrate the compass

```

```

' You can delete them if not required

```

```

  If GetPin(Calibrate)=0 Then                          ' If Push
Button pressed

```

```

    Call I2cByteWrite(Compass, CalReg, CalCmd ) ' then send

```

```

calibration command
    debug.Print "Calibrating... .";
End If
Loop
End Sub

'-----
'-----
' I2C subroutines follow
'-----
'-----

' writes I2cData to I2cReg at I2cAddr
Sub I2cByteWrite(ByVal I2cAddr As Byte, ByVal I2cReg As Byte,
ByVal I2cData As Byte)
    Call I2cStart()
    Call I2cOutByte(I2cAddr)           ' send device
address
    Call I2cOutByte(I2cReg)           ' send register
address
    Call I2cOutByte(I2cData)         ' send the data
    Call I2cStop()
End Sub

Function I2CByteRead(ByVal I2cAddr As Byte, ByVal I2cReg As Byte)
As Byte
    Call I2cStart()
    Call I2cOutByte(I2cAddr)         ' send device
address
    Call I2cOutByte(I2cReg)         ' send register
address
    Call I2cStart()                 ' repeated start
    I2cAddr = I2cAddr+1
    Call I2cOutByte(I2cAddr)         ' send device
address with read set
    I2cAck = False                  ' setup to send Nak
    I2cByteRead = I2cInByte()       ' get data byte with
Nak
    Call I2cStop()
End Function

Function I2CWordRead(ByVal I2cAddr As Byte, ByVal I2cReg As Byte)
As UnsignedInteger
    Set I2CWordRead = New UnsignedInteger
    Call I2cStart()
    Call I2cOutByte(I2cAddr)         ' send device
address
    Call I2cOutByte(I2cReg)         ' send register
address
    Call I2cStart()                 ' repeated start
    I2cAddr = I2cAddr+1
    Call I2cOutByte(I2cAddr)         ' send device
address with read set

```

```

        I2cAck = True                                     ' setup to send Ack
        I2cWordRead = CuInt(I2cInByte()*256)
        I2cAck = False                                   ' setup to send Nak
        I2cWordRead = I2cWordRead + CuInt(I2cInByte())
        Call I2cStop()
End Function

Sub I2cOutByte(I2cData As Byte)
    Call ShiftOut(SDA, SCL, 8, I2cData)                 ' shift data out
    Call PutPin(SDA, bxInputTristate)                   ' turn SDA around
    Call PutPin(SCL, bxOutputHigh)                       ' and clock in the
ack' bit
    Call PutPin(SCL, bxOutputLow)
End Sub

Function I2cInByte() As Byte
    I2cInByte = ShiftIn(SDA, SCL, 8)
    If I2cAck=True Then
        Call PutPin(SDA, bxOutputLow)
    Else
        Call PutPin(SDA, bxOutputHigh)
    End If
    Call PutPin(SCL, bxOutputHigh)                       ' clock out the ack'
bit
    Call PutPin(SCL, bxOutputLow)
End Function

Sub I2cStart()                                         ' I2C start bit
sequence
    Call PutPin(SDA, bxOutputHigh)
    Call PutPin(SCL, bxOutputHigh)
    Call PutPin(SDA, bxOutputLow)
    Call PutPin(SCL, bxOutputLow)
End Sub

Sub I2cStop()                                          ' I2C stop bit
sequence
    Call PutPin(SDA, bxOutputLow)
    Call PutPin(SCL, bxOutputHigh)
    Call PutPin(SDA, bxOutputHigh)
End Sub

```

You can find Compass Documentation [Here](#) as well as a small [FAQ](#)