Hitachi® HM55B Compass Module (#29123)

General Description

The Hitachi HM55B Compass Module is a dual-axis magnetic field sensor that can add a sense of direction to your next electronic or robotic project. The sensing device on the Compass Module is a Hitachi HM55B chip. An onboard regulator and resistor protection make the 3 volt HM55B chip compatible with 5 volt BASIC Stamp® microcontroller supply and signal levels. The Compass Module also makes all the power and signal connections on the tiny surface mount HM55B chip accessible in a breadboard-friendly 0.3 inch wide 6-pin DIP package. Acquiring measurements from the module is made easy with the BASIC Stamp 2 microcontroller's SHIFTIN and SHIFTOUT commands, which are designed for synchronous serial communication with chips like the HM55B.
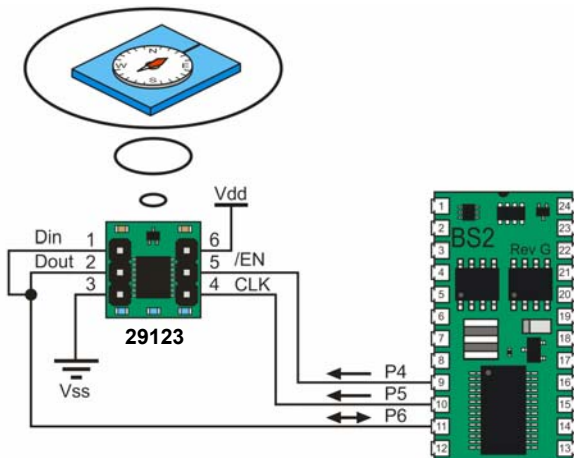
Features

- Sensitive to microtesla (µT) variations in magnetic field strength
- Simplifies direction by resolving magnetic field measurements into two component axes
- Good for 6-bit (64-direction) resolution measurements after software calibration
- Only 30 to 40 ms between start measurement and data-ready
- Built-in resistor protection for data pins eliminates bus conflict risks
- Compact and breadboard-friendly 0.3 inch, 6-pin DIP package
- Compatible with all BASIC Stamp, Javelin Stamp and SX microcontrollers
- Makes communication and direction calculations especially easy with the PBASIC commands SHIFTOUT, SHIFTIN, and ATN for the BASIC Stamp 2 series of microcontrollers

Application Ideas

- Mobile robot direction sensor
- Handheld electronic compass
- Weathervane indicator for remote weather stations
- Audible compass for the vision impaired
- Automotive electronic compass

Quick Start Circuit

Quick Start

Downloading the Example Programs

The example programs featured in the Source Code section (pages 12 - 24) are also available for download from the Hitachi HM55B Compass Module product page at www.parallax.com.

√ Go to www.parallax.com/detail.asp?product_id=29123.
√ Click the "Hitachi HM55B Compass Module Source Code (.zip)" link and download the file.
√ Unzip to a convenient folder.

## Connecting and Testing the Compass Module

This section contains connection and test instructions for the Parallax BASIC Stamp 2® series and Ubicom SX microcontrollers.

BASIC Stamp 2 Series

Use the instructions in this section to test your Hitachi HM55B Compass Module if you have a BASIC Stamp 2, 2sx, 2e, 2p, 2pe, or 2px. The goal of this first test is to verify that the module is connected, functional, and communicating properly with the BASIC Stamp. Since this first test is run before calibration, there may sizeable differences between magnetic compass directions and what the module reports.

√ Build the circuit shown on page 1.
√ Place your board on a level surface away from magnetic field disturbances. Common culprits include magnetic compasses, motors, bar/horseshoe/ring magnets, and large metal objects. Also, make sure your programming cable and power cords do not wrap around or pass near the sensor.
√ Open TestHm55bCompass.bs2 with the BASIC Stamp Editor (v2.0 or later). Update the $STAMP directive if needed, then run the program.
√ Test your Compass Module for direction detection as shown below, comparing your degree measurements to the compass legend in the figure.
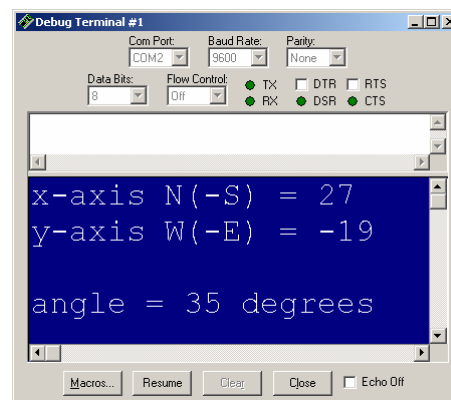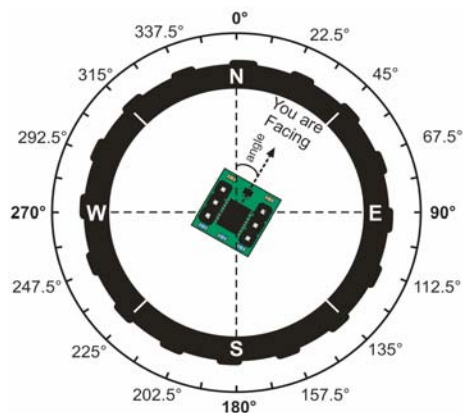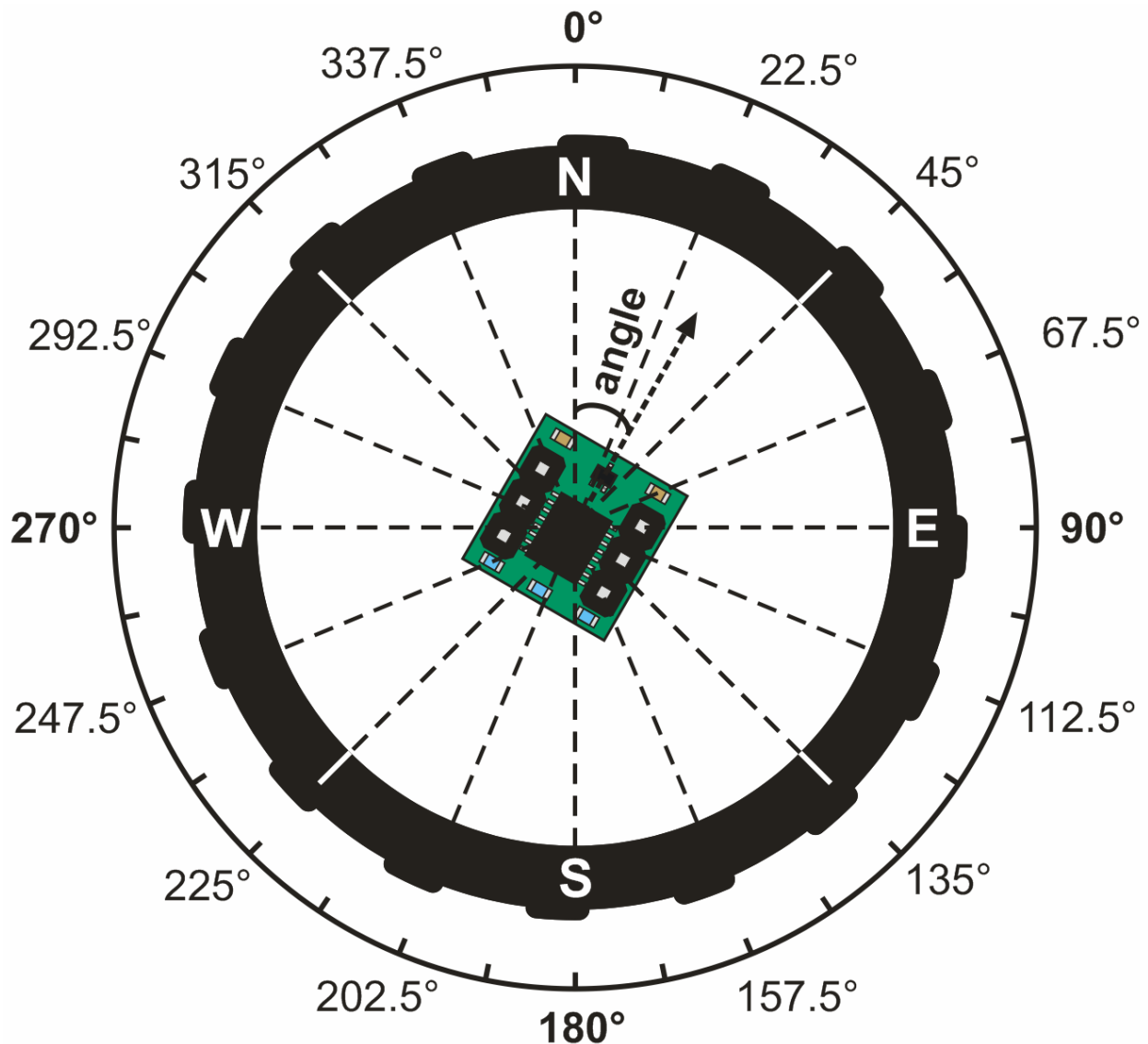
Quick Start

Calibration

The calibration and test-calibration programs will compensate for the effects of magnetic fields that may be resident to the module PCB and the carrier board it's mounted on, as well as nearby jumper wires and batteries. It also corrects for the HM55B chip's axis sensitivity, offset and skew errors, if any. You will need a magnetic compass to verify magnetic north as a reference. After calibration, the Hitachi HM55B Compass Module should be able to accurately discern 64 or more angles, referenced to magnetic north.

- √ Make a printout of the 16-segment compass shown below.
- √ Align the printout to magnetic north with the aid of the magnetic compass.
- √ Affix the aligned printout to your work surface.

- √ Make sure to set the magnetic compass well away from the printout before continuing.
- √ Align the Compass Module to magnetic north by lining up the edge of your carrier board with the dashed line that passes through the 0° mark.

Quick Start

The program CalibrateHm55bCompass.bs2 will prompt you to take a first series of measurements at 90° increments to gather compass offsets. Then, you will take a second series of measurements at 22.5° increments to gather information for a linear interpolation table. Both sets of measurements will be saved in the BASIC Stamp EEPROM so that the TestCalibratedHM55BCompass.bs2 program can use them.

- √ Run CalibrateHm55bCompass.bs2.
- √ Make sure to click the Transmit windowpane shown in this figure before following and responding to the Debug Terminal prompts.
- √ When you have completed the calibration process, run TestCalibratedHM55BCompass.bs2.
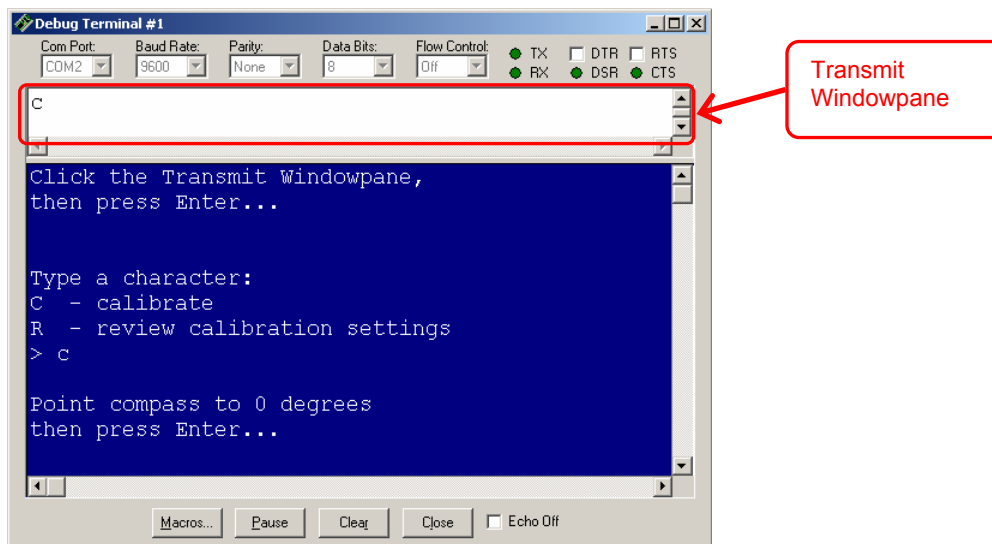
CalibrateHM55BCompass.bs2 writes the calibration values to EEPROM, so they will be available for TestCalibratedHM55BCompass.bs2 even after the power is reset. TestCalibratedHM55BCompass.bs2 will display the angle from north in both binary radians (brads) and degrees. When you run this program after calibration, the accuracy of the degree measurements should be significantly improved. While degrees parse a full circle into 360 equal segments, brads parse the circle into 256 segments. North is still 0, but the brad measurements for east, south and west are 64, 128, and 192 respectively. In applications where accuracy is the top priority, use the brad measurements because they introduce one less rounding error into the measurement. Further accuracy improvements can be realized if both the calibration and calibration-test programs are modified to average multiple measurements.

Testing Sensitivity

The Compass Module operates by comparing the magnetic field intensities sensed by its two perpendicular axes. The magnetic field strength corresponding to a measurement of 1 should resolve to somewhere between 1 and 1.6 µT. To find out how many microteslas per unit each axis reports with your particular Compass Module, use TestCalibratedHM55BCompass.bs2. Start by finding the maximum possible axis measurement. Do this by orienting the x-axis to magnetic north, then tilt up and down until you find the highest value. Compare this to the total magnetic field intensity in your area. In the continental US, you can find a Total Intensity Map on this page:

http://nationalatlas.gov/articles/geology/a_geomag.html.

Make sure to divide their nanotesla (nT) values by 1000 to convert to microteslas (µT).

Quick Start

SX Microcontroller

While this connection and testing procedure is set up for the SX-28, it should work on any of the SX series of microcontrollers with a simple update of the device directive. Make sure you are using the SX-Key v3.1 IDE v3.10 or newer, along with the SX-Key and an SX-28 chip.

- √ Build the circuit shown below.
- √ Open TestHm55bCompass.SXB with the SX-Key IDE.
- √ Save, Compile, and Debug (CTRL-S, A, D)
- √ Click the Debug window's Poll button. If the Watch window is not visible, also click the Watch button to bring it to the foreground.
- √ Monitor the Watch window as you point the module in various directions.
- √ Use arctan(-y/x) to calculate the heading angle, counterclockwise from north.
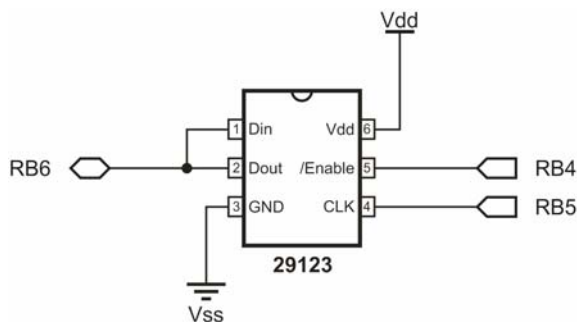
Resources and Downloads

Check out the Hitachi HM55B Compass Module product page for example programs, the HM55B datasheet, and more:

http://www.parallax.com/detail.asp?product_id=29123

Device Information

Theory of Operation

The Hitachi HM55B Compass Module has two axes, x and y. Each axis reports the strength of the magnetic field's component parallel to it. The x-axis reports (field strength) × cos(θ), and the y-axis reports the (field strength) × sin(θ). To resolve θ into a clockwise angle from north, use arctan(-y/x), which in PBASIC 2.5 is x ATN -y. The ATN command returns the angle in binary radians. To convert to degrees with PBASIC, just apply */ 360 to the variable storing the binary radian measurement.
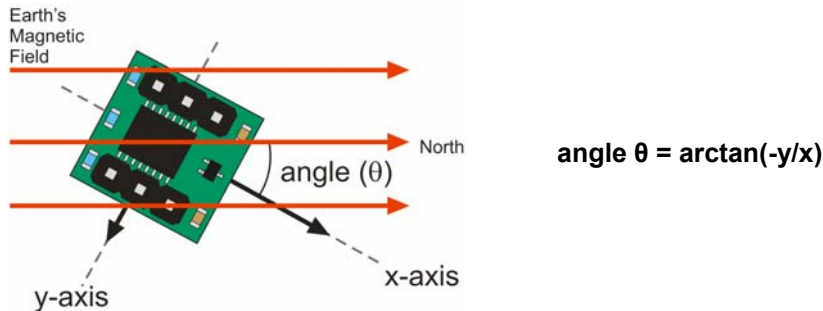

angle θ = arctan(-y/x)

The Hitachi HM55B chip on the Compass Module reports its x and y axis measurements in terms of microteslas (µT) in 11-bit signed values. The HM55B is designed to return a value of 1 for a north magnetic field of 1 µT parallel to one of its axes. If the magnetic field is south, the value will be -1. Keep in mind that these are nominal values. According to the HM55B datasheet, the actual µT value for a measurement of 1 could range anywhere from 1 to 1.6 µT. Also keep in mind that a negative 11-bit value will not appear negative in a word variable unless a mask is applied. For example, when bit-10 is 1, bits 11 to 15 are also changed to 1 with a mask in the test and calibration programs.

The microcontroller connected to the HM55B must control its enable and clock inputs and use synchronous serial communication to get the axis measurements from its data input and data output pins. For example, a BASIC Stamp 2 can be programmed to control the Compass Module's enable lines with HIGH/LOW and send values that reset the device and start a measurement with SHIFTOUT commands. The SHIFTOUT command controls the Compass Module's clock input as it sends data bit values to its data input. The converse of SHIFTOUT is SHIFTIN, which also controls the device's clock input as it collects data bits sent by the device's data output pin.

It takes the HM55B 30 to 40 ms to complete a given measurement. The microcontroller can either perform other tasks during this time or poll until the measurement is complete. The polling is a combination of SHIFTOUT commands that request the status, and SHIFTIN commands that acquire the status. When the SHIFTIN receives status flags indicating that the measurement is complete, a second and third SHIFTIN command can then store the 11-bit x and y axis measurements in variables.

Precautions

- Do not apply voltages to the device that are outside the values stated in the Pin Definitions and Ratings section.
- Do not operate or store the Compass Module near sources of strong magnetic fields. Strong magnetic fields can be created by bar and ring magnets, electric motors, and other coil elements such as solenoids, relays, and large inductors.
- Do not apply magnetic fields in excess of 300 µT to the Compass Module. Magnetic fields stronger than 300 µT can permanently damage the sensor.
- Mount the Compass Module as far away as possible from magnetic field disturbances. These include magnets (including compass needles), motors, power cords, coils, metal boxes, and sometimes the ground.

Device Information

Specifications

Symbol	Quantity	Minimum	Typical	Maximum	Units
B_{SE}	Sensitivity[†]		1.0	1.6	µT/lsb
H	Linear measurement range[†]	-180		180	µT
dθ	Individual axis offset			36	°
T_{CONV}	Conversion time[†]		30	40	ms
T_{OPE}	Operating temperature	0		70	°C

† From Hitachi HM55B Datasheet

Pin Definitions and Ratings

(1) Din - Serial data input
(2) Dout - Serial data output
(3) GND - Ground -> 0 V
(4) CLK - Synchronous clock input
(5) /EN - Active-low device enable
(6) Vcc - +5 V power input

29123

Din (1)   (6) Vcc
Dout (2)   (5) /EN
GND (3)   (4) CLK
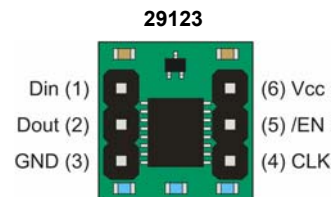
Symbol	Quantity	Minimum	Typical	Maximum	Units
Vcc	Supply Voltage	4.8	5.0	5.2	V
Icc(Ave)	Average active supply current[*]		5	7	mA
Icc(Pk)	Peak instantaneous current[**]		30	45	mA
Icc(Sb)	Standby supply current		2	3	mA
GND	Ground reference connection		0		V
V_{OH}	Signal high transmit (Dout)	Vcc × 0.9	Vcc	Vcc + 0.5	V
V_{OL}	Signal low transmit (Dout)	GND - 0.3	GND	Vcc × 0.15	V
V_{IH}	Signal high receive (/En, CLK, Din)	Vcc × 0.8	Vcc	Vcc + 0.3	V
V_{IL}	Signal low receive (/En, CLK, Din)	- 0.3	GND	Vcc × 0.12	V

*   Measurement cycle = 80 ms
** Typical duration is 5 µs

Device Information

Connection Diagrams

The 3-wire interface is recommended for most applications. While all the connections shown here are to individual I/O pins, the /EN pin is the only one that needs a dedicated I/O pin. The Din/Dout pins can share a line with other synchronous serial devices, and likewise with CLK.

3-wire interface
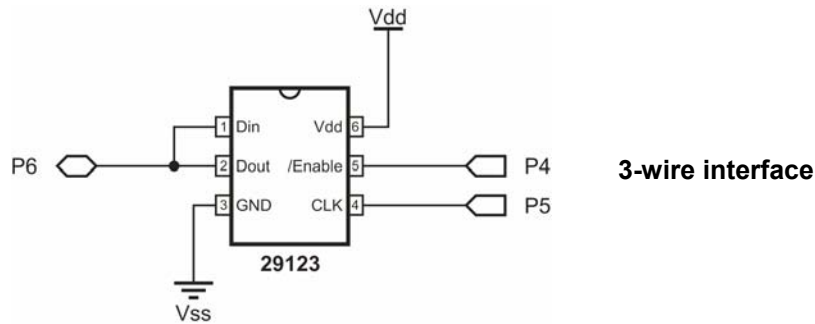
The Din and Dout pins do not have to be tied together; they can also be controlled individually. This makes it possible to share communication lines with other synchronous serial devices that have dedicated input and output lines.

4-wire interface
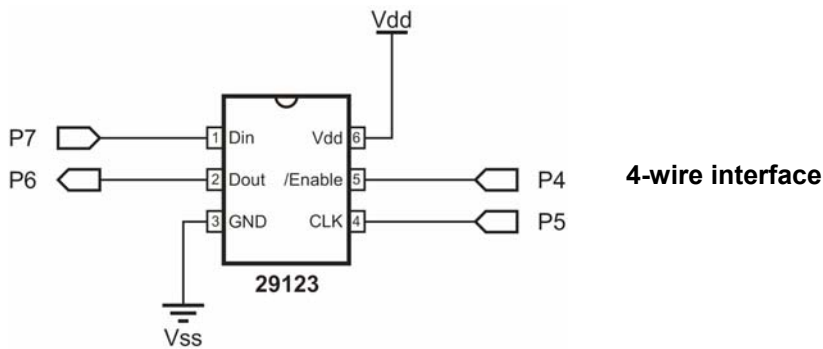
Command Set

These commands are shifted-out to the Compass Module.

Binary Value	Quantity
0000	Reset device
0001	Start measurement
0011	Report measurement status (and transmit the measurement if it's ready)

Device Information

Status Flags

The Compass Module will reply to the report measurement status command with one of these values.

Binary Value	Quantity
Bits - 3210	3 and 2 indicate measurement completion, 1 and 0 indicate measurement errors
1100	11 -> Measurement completed; 00 -> no errors
00XX	Measurement still in progress, or the device has been reset.
XX11	/EN did not receive low-high-low signal between start and report commands

Communication Protocol

All values transmitted to and received from the Compass Module are most significant bit first, with the bit value valid after the clock signal's rising edge. For the BASIC Stamp 2, this means set the SHIFTOUT command's Mode argument to MSBFIRST, and the SHIFTIN command's Mode argument to MSBPOST.

To reset the HM55B, take /EN from high to low, and shift-out %0000, then set /EN high again.
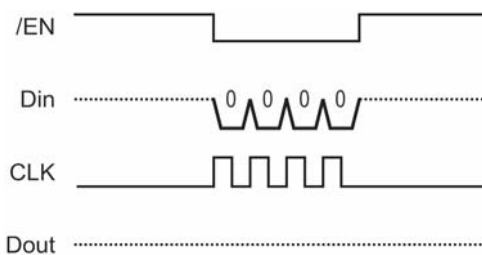
After reset, start a measurement by taking /EN low again, then shift-out %1000. Leave /EN low until checking the measurement status.
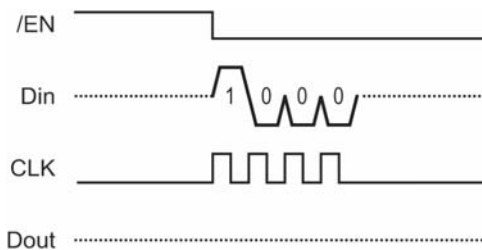
Device Information

To check the measurement status, start by sending a positive pulse to /EN. Then, shift-out %1100, and shift-in the status flags. While the measurement is in progress, the end flag and error flag will both be 00. The compass Module may be polled for status repeatedly until the measurement is complete, at which point the end flag will change to 11. Upon receipt of %1100, discontinue polling. Leave /EN low, and move on to shifting-in the x and y-axis values.
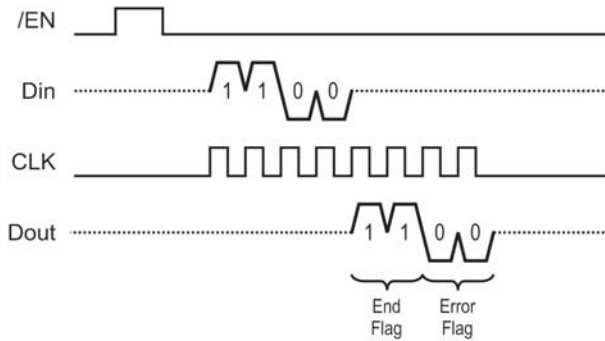
Shifting-in the x and y-axis values is a simple matter of shifting-in 11 bits for the x-axis measurement followed by 11 more bits for the y-axis measurement. After completing the y-axis shift-in, set the /EN pin high again.
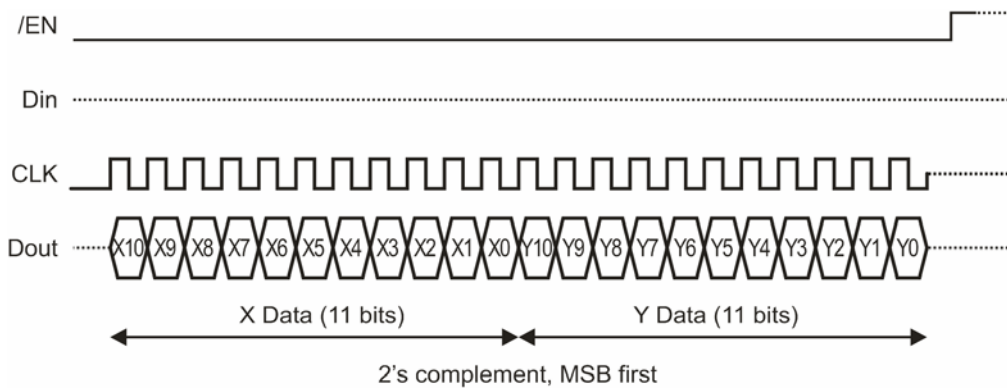
Module Dimensions
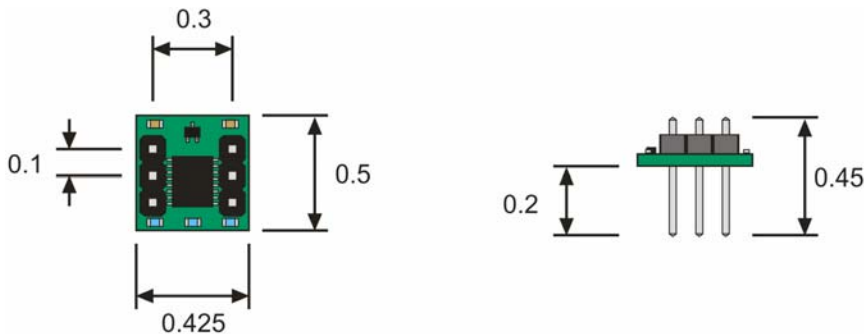
Device Information

Module Schematic
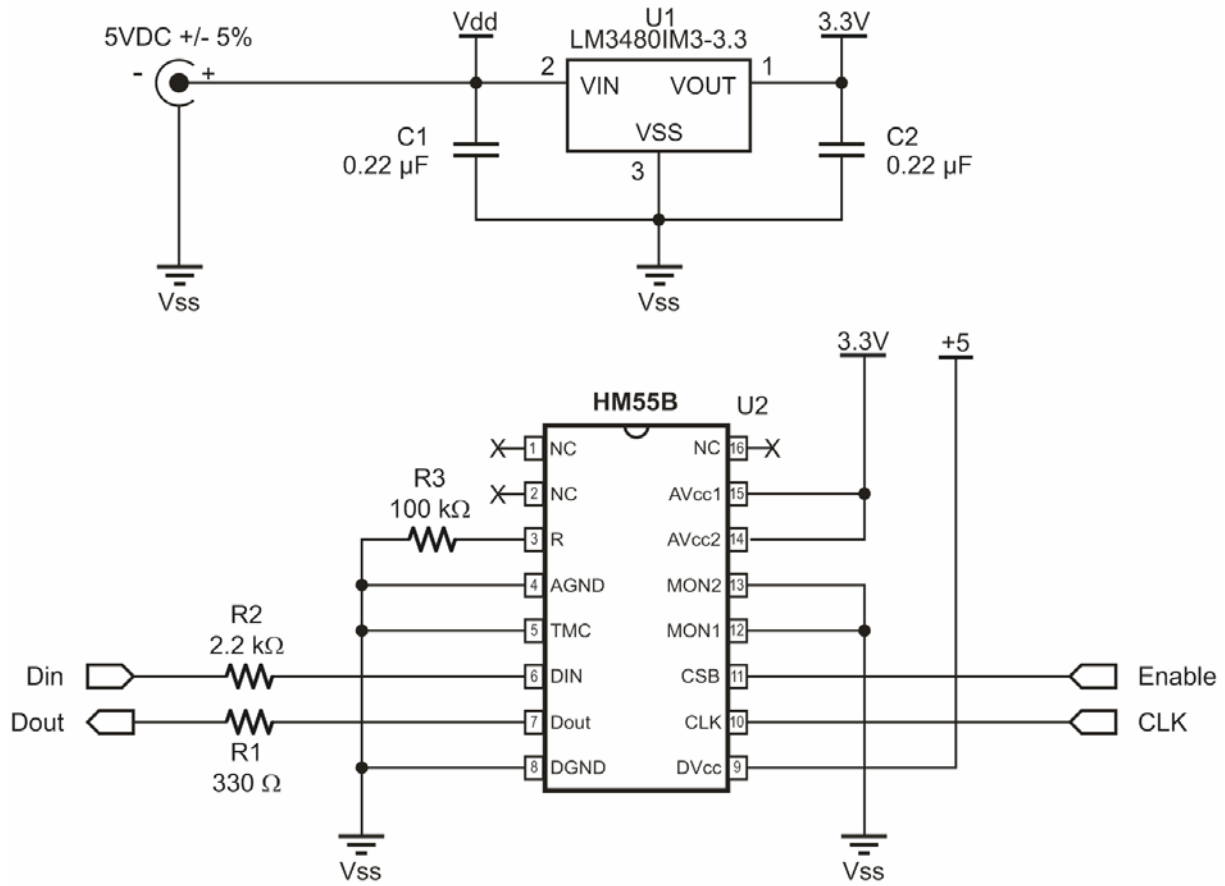
Source Code

BASIC Stamp® 2 Series Test Program

If you are using a BS2e, BS2sx, BS2p, BS2pe, or BS2px, make sure to update the $STAMP directive before running the program. No other part of the program will need to be changed.

```
' =============================================================================
' TestHm55bCompass.bs2 - This Hitachi HM55B Compass Module test program
' displays x (N/S) and y (W/E) axis measurements along with the direction the
' Compass Module is pointing, measured in degrees clockwise from north.
'
' Author.... (C) 2005 Parallax, Inc -- All Rights Reserved
' Email.....  support@parallax.com
'
' {$STAMP BS2}
' {$PBASIC 2.5}
' =============================================================================


' -----[ Pins/Constants/Variables ]--------------------------------------------
DinDout        PIN     6                    ' P6 transceives to/from Din/Dout
Clk            PIN     5                    ' P5 sends pulses to HM55B's Clk
En             PIN     4                    ' P4 controls HM55B's /EN(ABLE)

Reset          CON     %0000                ' Reset command for HM55B
Measure        CON     %1000                ' Start measurement command
Report         CON     %1100                ' Get status/axis values command
Ready          CON     %1100                ' 11 -> Done, 00 -> no errors
NegMask        CON     %1111100000000000    ' For 11-bit negative to 16-bits

x              VAR     Word                 ' x-axis data
y              VAR     Word                 ' y-axis data
status         VAR     Nib                  ' Status flags
angle          VAR     Word                 ' Store angle measurement

' -----[ Main Routine ]---------------------------------------------------------

DO                                          ' Main loop

  GOSUB Compass_Get_Axes                    ' Get x, and y values

  angle = x ATN -y                          ' Convert x and y to brads
  angle = angle */ 360                      ' Convert brads to degrees

  DEBUG HOME, "x-axis N(-S) = ",SDEC x,     ' Display axes and degrees
        CLREOL, CR, "y-axis W(-E) = ",
        SDEC y, CLREOL, CR, CR, "angle = ",
        DEC angle, " degrees", CLREOL

  PAUSE 150                                 ' Debug delay for slower PCs

LOOP                                        ' Repeat main loop


' -----[ Subroutines ]----------------------------------------------------------
```

Source Code

```
Compass_Get_Axes:                                ' Compass module subroutine

  HIGH En: LOW En                                ' Send reset command to HM55B
  SHIFTOUT DinDout,clk,MSBFIRST,[Reset\4]

  HIGH En: LOW En                                ' HM55B start measurement command
  SHIFTOUT DinDout,clk,MSBFIRST,[Measure\4]
  status = 0                                     ' Clear previous status flags

  DO                                             ' Status flag checking loop
    HIGH En: LOW En                              ' Measurement status command
    SHIFTOUT DinDout,clk,MSBFIRST,[Report\4]
    SHIFTIN  DinDout,clk,MSBPOST,[Status\4]      ' Get Status
  LOOP UNTIL status = Ready                      ' Exit loop when status is ready

  SHIFTIN  DinDout,clk,MSBPOST,[x\11,y\11]       ' Get x & y axis values
  HIGH En                                        ' Disable module

  IF (y.BIT10 = 1) THEN y = y | NegMask          ' Store 11-bits as signed word
  IF (x.BIT10 = 1) THEN x = x | NegMask          ' Repeat for other axis

  RETURN
```

BASIC Stamp 2 Series Calibration Program

```
' ============================================================================
'
'   File...... CalibrateHm55bCompass.bs2
'   Purpose... Software calibration of Hitachi HM55B Compass Sensor
'   Author.... (C) 2005 Parallax, Inc -- All Rights Reserved
'   E-mail.... support@parallax.com
'   Started... 5/31/05
'   Updated...
'
'   {$STAMP BS2}
'   {$PBASIC 2.5}
'
' ============================================================================

' -----[ Program Description ]-------------------------------------------------

' This program collects and stores Hitachi HM55B Compass Module measurements
' in EEPROM for axis offset and linear interpolation corrections that will be
' performed by TestCalibratedHm55bCompass.bs2.
'
' IMPORTANT: Follow the instructions in the Quick Start section of "Hitachi
'            HM55B Compass Module Documentation (.pdf)",  available for
'            download from the Hitachi HM55B Compass Module product page at
'            www.parallax.com.


' -----[ EEPROM Data ]---------------------------------------------------------
```

Source Code

```
CompassOffsets DATA @ 0, (4)                    ' Stores x and y axis offsets
CompassLowVal  DATA     (1)                      ' Stores index of lowest angle
CompassCal     DATA     (16)                     ' 16 reference compass angles


' -----[ Pin Definitions ]-------------------------------------------------
DinDout        PIN      6                        ' P6 transceives to/from Din/Dout
Clk            PIN      5                        ' P5 sends pulses to HM55B's Clk
En             PIN      4                        ' P4 controls HM55B's /EN(ABLE)


' -----[ Constants ]-------------------------------------------------------
Reset          CON      %0000                    ' Reset command for HM55B
Measure        CON      %1000                    ' Start measurement command
Report         CON      %1100                    ' Get status/axis values command
Ready          CON      %1100                    ' 11 -> Done, 00 -> no errors
NegMask        CON      %1111100000000000        ' For 11-bit negative to 16-bits


Current        CON      0                        ' Index for table array
Previous       CON      1                        ' Index for table array


' -----[ Variables ]-------------------------------------------------------
x              VAR      Word                     ' x-axis data
y              VAR      Word                     ' y-axis data
status         VAR      Nib                      ' Status flags
angle          VAR      Word                     ' Angle measurement
counter        VAR      Byte                     ' Loop counter
index          VAR      Nib                      ' EEPROM index
character      VAR      Byte                     ' Stores a DEBUGIN character
integer        VAR      Word                     ' Integer values for display
fraction       VAR      Nib                      ' Fractional values for display
brads          VAR      Byte                     ' Binary radian measurements
offset         VAR      Word                     ' Axis offset
table          VAR      Byte(2)                  ' Stores table values
temp           VAR      Word(2)                  ' Stores axis measurements
axisOffset     VAR      Word                     ' Stores axis offset value


' -----[ Main Routine ]----------------------------------------------------

DEBUG "Click the Transmit Windowpane, ", CR, ' Wait for user.
      "then press Enter... ", CR, CR

DEBUGIN character

DO                                               ' Main loop

  DEBUG "Type a character: ", CR,                ' Menu
        "C  - calibrate ", CR,
        "R  - review calibration settings", CR,
        "> "
  DEBUGIN Character                              ' Get user selection
  DEBUG CR

  IF character = "c" OR character = "C" THEN ' "c" -> calibrate
    GOSUB Compass_Calibrate                      ' "r" -> review settings
```

Source Code

```
    ELSEIF character = "r" OR character = "R" THEN
      GOSUB Calibration_Review
    ENDIF

    DEBUG CR, "Press any key to",              ' wait for user
          CR, "continue"
    DEBUGIN character
    DEBUG CR, CR

LOOP                                           ' Repeat main loop

' -----[ Subroutine - Compass_Calibrate ]-------------------------------------

  Compass_Calibrate:

    GOSUB Get_And_Store_Axis_Offsets
    GOSUB Get_And_Store_Interpolation
    GOSUB Get_And_Store_Low_Value_Address
    DEBUG CR, "CALIBRATION COMPLETED...", CR,
          "You are now ready to run ", CR,
          "TestCalibratedHm55bCompass.bs2.", CR
    RETURN

' -----[ Subroutine - Get_And_Store_Axis_Offsets ]---------------------------

' This subroutine prompts the user to point the compass north, then east, then
' south, then west.  It then averages the maximum and minimum values for each
' axis and stores that average in the EEPROM area reserved by the
' CompassOffsets DATA directive.

Get_And_Store_Axis_Offsets:

  ' FOR...NEXT loop repeats for four axis measurements.
  FOR counter = 0 TO 3

    ' Instruct user to point compass to a particular direction, then wait
    ' for ENTER character.
    DEBUG CR, "Point compass to "
    LOOKUP counter, [ 0, 90, 180, 270 ], integer
    DEBUG DEC integer
    DEBUG " degrees", CR, "then press Enter..."
    DEBUGIN character

    GOSUB Compass_Get_Axes                     ' Get axis measurements

    ' Calculate offsets based on max and min values for each axis, then store
    ' in EEPROM.
    SELECT counter
      CASE 0                                   ' North
        temp(0) = x
      CASE 1                                   ' East
        temp(1) = y
      CASE 2                                   ' South
```

Source Code

```
        x = x + temp(0)
        IF x.BIT15 = 1 THEN
          x = ABS(x)/2
          x = -x
        ELSE
          x = x / 2
        ENDIF
        WRITE CompassOffsets, Word x
      CASE 3                                  ' West
        y = y + temp(1)
        IF Y.BIT15 = 1 THEN
          y = ABS(y)/2
          y = - y
        ELSE
          y = x / 2
        ENDIF
        WRITE CompassOffsets + 2, Word y
    ENDSELECT

  NEXT

  RETURN

' -----[ Subroutine - Get_And_Store_Interpolation ]--------------------

' This subroutine prompts the user to point the compass to directions
' separated by 22.5 degrees and stores the angle for each of the measurements
' in the EEPROM area reserved by the CompassCal DATA directive.

Get_And_Store_Interpolation:

  FOR counter = 0 TO 15
    DEBUG CR, "Point compass to "
    LOOKUP counter, [0,    22,  45,  67,  90, 112, 135, 157,
                     180, 202, 225, 247, 270, 292, 315, 337], integer
    LOOKUP counter, [ 0,    5,   0,   5,   0,   5,   0,   5,
                      0,    5,   0,   5,   0,   5,   0,   5 ], fraction
    DEBUG DEC integer
    IF fraction = 5 THEN DEBUG ".", DEC fraction
    DEBUG " degrees", CR, "then press Enter..."
    DEBUGIN character                         ' Wait for user
    GOSUB Compass_Get_Axes                    ' Get x, and y values
    GOSUB Compass_Correct_Offsets             ' Correct axis offsets
    angle = x ATN - y                         ' Convert x and y to brads
    WRITE CompassCal + counter, angle         ' Store as brad value
  NEXT

  RETURN

' -----[ Subroutine - Get_And_Store_Low_Value_Address ]----------------------

' This subroutine finds and stores the address of the lowest value in the
' EEPROM area reserved by the CompassCal DATA directive and stores it in
```

Source Code

```
' a byte reserved by the CompassLowVal DATA directive.  This reduces the
' code overhead in TestCalibratedHm55bCompass.bs2.

Get_And_Store_Low_Value_Address:

  index = 8
  table(current) = 0: table(previous) = 0
  DO
    index = index + 1
    READ CompassCal + index, table(current)
    READ CompassCal + (index - 1 & $F), table(previous)
  LOOP UNTIL table(current) < table(previous)
  WRITE CompassLowVal, index

  RETURN

' -----[ Subroutine - Calibration_Review ]-----------------------------------

  ' Display EEPROM values.

  Calibration_Review:

    DEBUG CR, "Axis Offsets:", CR
    READ CompassOffsets, Word x
    DEBUG CR, "x-Offset = ", SDEC x
    READ CompassOffsets + 2, Word y
    DEBUG CR, "y-Offset = ", SDEC y, CR

    DEBUG CR, "Index of low value in CompassCal:", CR
    READ CompassLowVal, index
    DEBUG CR, "Low value ", ? index

    DEBUG CR, "TestCalibratedHm55bCompass.bs2", CR,
              "uses the 'actual' values to ", CR,
              "correct measurement errors:", CR

    DEBUG CR, "Brad Angle       Degree Angle",
          CR, "Ideal    Actual   Ideal    Actual",
          CR, "------   ------   ------   ------", CR

    FOR counter = 0 TO 15
      brads = counter * 16
      DEBUG CRSRX, 1, DEC3 brads
      READ CompassCal + counter, angle
      DEBUG CRSRX, 10, DEC3 angle
      LOOKUP counter, [0,    22, 45,  67,  90, 112, 135, 157,
                       180, 202, 225, 247, 270, 292, 315, 337], integer
      LOOKUP counter, [ 0,    5,   0,   5,   0,   5,   0,   5,
                        0,    5,   0,   5,   0,   5,   0,   5 ], fraction
      DEBUG CRSRX, 19, DEC3 integer, ".", DEC fraction
      angle = angle */ 360                        ' Convert brads to degrees
      DEBUG CRSRX, 28, DEC3 angle, CR
      PAUSE 50                                    ' Debug delay for slower PCs
```

Source Code

```
    NEXT

    DEBUG CR

    RETURN

' -----[ Subroutine - Compass_Get_Axes ]-------------------------------------

Compass_Get_Axes:                                  ' Compass module subroutine

  HIGH En: LOW En                                  ' Send reset command to HM55B
  SHIFTOUT DinDout,clk,MSBFIRST,[Reset\4]

  HIGH En: LOW En                                  ' HM55B start measurement command
  SHIFTOUT DinDout,clk,MSBFIRST,[Measure\4]
  status = 0                                        ' Clear previous status flags

  DO                                               ' Status flag checking loop
    HIGH En: LOW En                                ' Measurement status command
    SHIFTOUT DinDout,clk,MSBFIRST,[Report\4]
    SHIFTIN  DinDout,clk,MSBPOST,[Status\4]        ' Get Status
  LOOP UNTIL status = Ready                        ' Exit loop when status is ready

  SHIFTIN  DinDout,clk,MSBPOST,[x\11,y\11]         ' Get x & y axis values
  HIGH En                                          ' Disable module

  IF (y.BIT10 = 1) THEN y = y | NegMask            ' Store 11-bits as signed word
  IF (x.BIT10 = 1) THEN x = x | NegMask            ' Repeat for other axis

    RETURN

' -----[ Subroutine - Compass_Correct_Offsets ]------------------------------

' This subroutine corrects cumulative magnetic field interference that can
' come from sources such as the PCB, jumper wires, a nearby batter, or a
' nearby current source.  This subroutine relies on values stored in
' the EEPROM space that was reserved by the CompassOffsets DATA directive.
' These EEPROM values are written by this program during calibration.

Compass_Correct_Offsets:

  READ CompassOffsets, Word axisOffset             ' Get x-axis offset
  x = x - axisOffset                               ' Correct x-axis
  READ CompassOffsets + 2, Word axisOffset         ' Get y-axis offset
  y = y - axisOffset                               ' Correct y-axis

    RETURN
```

Source Code

BASIC Stamp® 2 Series Calibration-Test Program

```
' ===============================================================================
'
'   File...... TestCalibratedHM55BCompass.bs2
'   Purpose... Demonstrates Hitachi HM55B Compass Module's accuracy after
'              calibration with CalibrateHM55BCompass.bs2.
'   Author.... (C) 2005 Parallax, Inc -- All Rights Reserved
'   E-mail.... support@parallax.com
'   Started... 5/31/05
'   Updated...
'
'   {$STAMP BS2}
'   {$PBASIC 2.5}
'
' ===============================================================================


' -----[ Program Description ]----------------------------------------------------

' This program displays the following Hitachi HM55B Compass Sensor measurements:
'
'  - Offset corrected x and y-axis magnetic field measurements
'  - Binary radian angle clockwise from north corrected by linear
'    interpolation table
'  - Degree angle clockwise from north corrected by linear interpolation
'    table

' IMPORTANT: This program relies on EEPROM values that are stored by
'            CalibrateHM55BCompass.bs2 during the calibration process.
'            This calibration process must be performed prior to running
'            this test program.
'
'            For instructions on how to perform the calibration process,
'            consult the Quick Start section in "Hitachi HM55B Compass Module
'            Documentation (.pdf)".  It's available for download from the
'            Hitachi HM55B Compass Module product page at www.parallax.com.


' -----[ EEPROM Data ]----------------------------------------------------------

CompassOffsets DATA @ 0, (4)                 ' Stores x and y axis offsets
CompassLowVal  DATA      (1)                  ' Stores index of lowest angle
CompassCal     DATA      (16)                 ' 16 reference compass angles

' -----[ Pin Definitions ]------------------------------------------------------

DinDout        PIN      6                     ' P6 transceives to/from Din/Dout
Clk            PIN      5                     ' P5 sends pulses to HM55B's Clk
En             PIN      4                     ' P4 controls HM55B's /EN(ABLE)

' -----[ Constants ]------------------------------------------------------------

Reset          CON      %0000                 ' Reset command for HM55B
```

Source Code

```
Measure          CON     %1000                    ' Start measurement command
Report           CON     %1100                    ' Get status/axis values command
Ready            CON     %1100                    ' 11 -> Done, 00 -> no errors
NegMask          CON     %1111100000000000        ' For 11-bit negative to 16-bits
current          CON     0                        ' Table array index
previous         CON     1                        ' Table array index


' -----[ Variables ]-------------------------------------------------------------
x                VAR     Word                     ' x-axis data
y                VAR     Word                     ' y-axis data
status           VAR     Nib                      ' Status flags
angle            VAR     Word                     ' Angle measurement
axisOffset       VAR     angle                    ' Axis offset

index            VAR     Status                   ' EEPROM index
table            VAR     Byte(2)                  ' Stores EEPROM table values
span             VAR     x                        ' Span between table entries
angleOffset      VAR     y                        ' Offset btwn measured and table


' -----[ Initialization ]--------------------------------------------------------

DEBUG CLS


' -----[ Main Routine ]----------------------------------------------------------

DO                                                ' Main loop

  GOSUB Compass_Get_Axes                          ' Get x, and y values
  GOSUB Compass_Correct_Offsets                   ' Correct axis offsetes
  angle = x ATN -y                                ' Convert x and y to brads
  DEBUG HOME, "x-axis N(-S) = ",SDEC x,           ' Display corrected axes
        CLREOL, CR, "y-axis W(-E) = ",
        SDEC y, CLREOL
  GOSUB Compass_Interpolate                       ' Linear interpolation
  DEBUG CR, CR, "angle = ",                       ' Display inrerpolated angle
        DEC angle, " brads", CLREOL               ' ... in brads
  angle = angle */ 360                            ' Convert brads to degrees
  DEBUG CR,"angle = ",                            ' Display inrerpolated angle
        DEC angle, " degrees", CLREOL             ' ... in degrees
  PAUSE 150                                       ' Debug delay for slower PCs


LOOP                                              ' Repeat main loop

' -----[ Subroutine - Compass_Get_Axes ]-----------------------------------------

' This subroutine handles BASIC Stamp - HM55B communication and stores the
' magnetic field strength measurements returned by the device in the x and
' y axis variables.

Compass_Get_Axes:                                 ' Compass module subroutine

  HIGH En: LOW En                                 ' Send reset command to HM55B
  SHIFTOUT DinDout,clk,MSBFIRST,[Reset\4]
```

Source Code

```
  HIGH En: LOW En                               ' HM55B start measurement command
  SHIFTOUT DinDout,clk,MSBFIRST,[Measure\4]
  status = 0                                    ' Clear previous status flags

  DO                                            ' Status flag checking loop
    HIGH En: LOW En                             ' Measurement status command
    SHIFTOUT DinDout,clk,MSBFIRST,[Report\4]
    SHIFTIN  DinDout,clk,MSBPOST,[Status\4]     ' Get Status
  LOOP UNTIL status = Ready                      ' Exit loop when status is ready

  SHIFTIN  DinDout,clk,MSBPOST,[x\11,y\11]      ' Get x & y axis values
  HIGH En                                       ' Disable module

  IF (y.BIT10 = 1) THEN y = y | NegMask         ' Store 11-bits as signed word
  IF (x.BIT10 = 1) THEN x = x | NegMask         ' Repeat for other axis

  RETURN

' -----[ Subroutine - Compass_Correct_Offsets ]-------------------------------

' This subroutine corrects cumulative magnetic field interference that can
' come from sources such as the PCB, jumper wires, a nearby battery, or a
' nearby current source.  This subroutine relies on values stored in
' the EEPROM space that was reserved by the CompassOffsets DATA directive.
' These EEPROM values were written by CalibrateHM55BCompass.bs2.

Compass_Correct_Offsets:

  READ CompassOffsets, Word axisOffset          ' Get x-axis offset
  x = x - axisOffset                            ' Correct x-axis
  READ CompassOffsets + 2, Word axisOffset      ' Get y-axis offset
  y = y - axisOffset                            ' Correct y-axis

  RETURN

' -----[ Subroutine - Compass_Interpolate ]-----------------------------------

' This subroutine applies linear interpolation to the refine the compass
' measurement.  This second level of refinement can be performed after the
' Compass_Correct_Offsets subroutine, and it can correct axis skew and other
' errors inherent to the HM55B chip.
'
' The subroutine relies on sixteen actual compass measurements that were stored
' in the sixteen EEPROM locations reserved by the CompassCal DATA directive.
' These measurements were stored by CalibrateHM55BCompass.bs2, and they
' represent the actual compass measurements for 0, 22.5, 45, 90,..., 337.5
' degrees.  The subroutine finds the two EEPROM measurements that the current
' angle measurement falls between.  It then updates the angle measurement
' based on where the angle measurement falls between the two known table values.

Compass_Interpolate:
```

Source Code

```
' Start with the lowest value in the CompassCal table.

READ CompassLowVal, index

' Load current and previous table values.

READ CompassCal + index, table(current)
READ (CompassCal + (index - 1 & $F)), table(previous)


' The IF...ELSEIF...ELSE...ENDIF code block finds the two EEPROM CompassCal
' table values that the current angle measurement falls between and calculates
' the difference between the current angle measurement and the lower of the
' two table values.  The IF and ELSEIF blocks deal with values that are
' greater than the highest or less than the lowest table values.  The ELSE
' block everything between the highest and lowest table values.

IF (angle >= table(previous)) THEN
  span = (255 - table(previous)) + table(current)
  angleOffset = angle - table(previous)
ELSEIF (angle <= table(current)) THEN
  span = table(current) + (255 - table(previous))
  angleOffset = angle + (255 - table(previous))
ELSE
  index = index - 1
  READ CompassCal + index, table(current)
  DO
    table(previous) = table(current)
    index = index + 1
    READ CompassCal + index, table(current)
    IF (angle <= table(current)) AND (angle > table(previous)) THEN
      span = table(current) - table(previous)
      angleOffset = angle - table(previous)
      EXIT
    ENDIF
  LOOP
ENDIF


' After the offset between the current angle measurement and the next lower
' table measurement has been determined, this code block uses it along with
' the span between the table entries above and below the angle measurement
' to solve for: angle(corrected) = angle(offset) * 16 / span.
' This code block also rounds up or down by comparing the remainder of
' the angleOffset / span division to the value of (span / 2).

angleOffset = angleOffset * 16
angle = (angleOffset / span) + ((angleOffset // span) / (span / 2))
angle = ((index - 1 & $F) * 16) + angle
angle = angle & $ff

RETURN
```

Source Code

SX Microcontroller Test Program

```
' ===========================================================================
'
'   File...... TestHm55bCompass.SXB
'   Purpose... Detect axis measurements with the SX chip
'   Author.... (c) Parallax, Inc. -- All Rights Reserved
'   E-mail.... support@parallax.com
'   Started...
'   Updated... 04 MAY 2004
'
' ===========================================================================


' ---------------------------------------------------------------------------
' Program Description
' ---------------------------------------------------------------------------
'
' This Hitachi HM55B Compass Module test program displays x (N/S) and
' y (W/E) axis measurements.  To calculate the module's angle from north,
' use arctan(-y/x).
'
' To view the x and y axis measurements, click Run -> Debug.  Then , click
' Poll in the Debug window.  As you rotate the compass, the values in the
' Watch window will update.


' ---------------------------------------------------------------------------
' Device Settings
' ---------------------------------------------------------------------------

DEVICE          SX28, OSC4MHZ, TURBO, STACKX, OPTIONX
FREQ            4_000_000


' ---------------------------------------------------------------------------
' IO Pins
' ---------------------------------------------------------------------------

DinDout         VAR    RB.6                ' RB.6 transceives to/from Din/Dout
Clk             VAR    RB.5                ' RB.5 sends pulses to HM55B's Clk
En              VAR    RB.4                ' RB.4 controls HM55B's /EN(ABLE)


' ---------------------------------------------------------------------------
' Constants
' ---------------------------------------------------------------------------

YOffset         CON    0                   ' Enter measured y at north here
XOffset         CON    0                   ' Enter measured x at west here

ResetHM         CON    %0000               ' Reset command for HM55B
Measure         CON    %1000               ' Start measurement command
Report          CON    %1100               ' Get status/axis values command
Ready           CON    %1100               ' 11 -> Done, 00 -> no errors
NegMask         CON    %1111100000000000   ' For 11-bit negative to 16-bits
```

Source Code

```
' -----------------------------------------------------------------------------
' Variables
' -----------------------------------------------------------------------------

x               VAR     Byte(2)                 ' y-axis data
y               VAR     Byte(2)                 ' x-axis data
statusFlags     VAR     Byte                    ' HM55B Status flags

' -----------------------------------------------------------------------------
' Watch Directives
' -----------------------------------------------------------------------------

watch x, 11, SDEC                               ' Watch x-axis variable
watch y, 11, SDEC                               ' Watch y-axis variable

' =============================================================================
  PROGRAM Start
' =============================================================================

Start:                                          ' Program execution starts here

' -----------------------------------------------------------------------------
' Initialization
' -----------------------------------------------------------------------------

HIGH En                                         ' Disable HM55B
LOW  Clk                                        ' Start with clock line output-low

' -----------------------------------------------------------------------------
' Main Routine
' -----------------------------------------------------------------------------

DO                                              ' Main loop
  GOSUB Get_Compass_Axes                        ' Get x and y axis values
  BREAK()                                        ' Update x and y in watch window
  PAUSE 100                                      ' 1/10 second delay for display
LOOP                                             ' Repeat main loop

' -----------------------------------------------------------------------------
' Subroutines
' -----------------------------------------------------------------------------

Get_Compass_Axes:

  HIGH En                                       ' Reset HM55B
  Low En
  SHIFTOUT DinDout,Clk,MSBFIRST,ResetHM \ 4
  HIGH En
  Low En                                        ' Start measurement
  SHIFTOUT DinDout,clk,MSBFIRST,Measure \ 4
  statusFlags = 0                               ' Clear previous status flags
  DO                                            ' Repeat until measurement ready
    HIGH En                                     ' Request measurement status
```

Source Code

```
   LOW En
   SHIFTOUT DinDout,clk,MSBFIRST,Report \ 4
   SHIFTIN  DinDout,clk,MSBPOST,StatusFlags\4 ' Get measurement status
LOOP UNTIL statusFlags = Ready
SHIFTIN  DinDout,clk,MSBPOST, x(1) \3       ' Get 11 signed x-axis bits
SHIFTIN  DinDout,clk,MSBPOST, x(0)
SHIFTIN  DinDout,clk,MSBPOST, y(1) \3       ' Get 11 signed y-axis bits
SHIFTIN  DinDout,clk,MSBPOST, y(0)
HIGH En                                     ' Disable HM55B

RETURN
```