

## Hitachi® HM55B デジタルコンパスモジュール (#29123)

### 概略

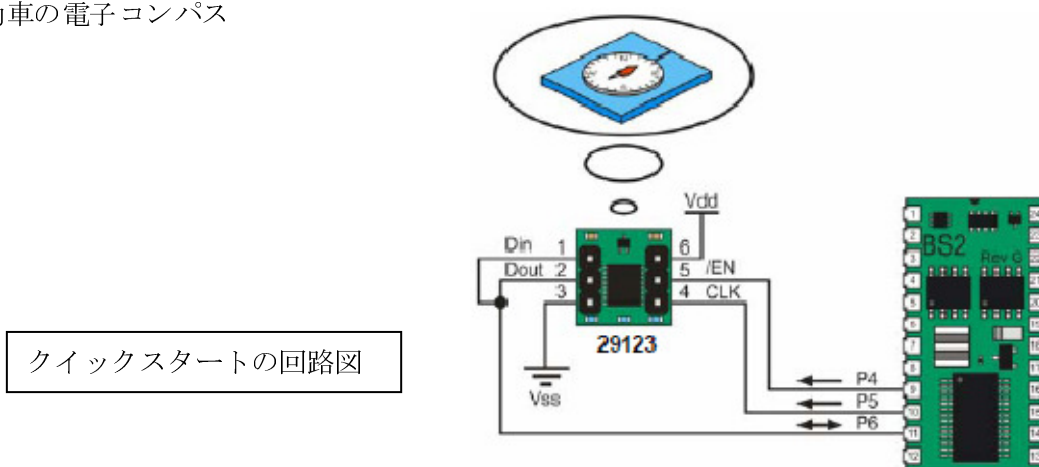
Hitachi HM55B デジタルコンパスモジュールは、あなたの次のロボットプロジェクトあるいは、電子工学のプロジェクトに適用されるかも知れません。これは、2軸の電磁場を感知するセンサーモジュールです。コンパスモジュール上の感知デバイスは、Hitachi の HM55B チップです。ボード上のレギュレーターと抵抗による保護があるので3ボルトの HM55B チップを5ボルトの BASIC Stamp® マイクロコントローラーの信号レベルに合うようになっています。コンパスモジュールはまた、全てのパワーと信号が小さな表面実装の HM55B チップにアクセスが出来、0.3インチ幅で6ピンの DIP パッケージになっていますのでブレッドボードにも簡単に取り付けることができます。モジュールから測定値を取得するには、ベーシックスタンプ2マイクロコントローラーの SHIFTOUT と SHIFTIN コマンドで簡単に出来ます。これは、HM55B のようなチップで同期シリアル通信をするために設計されたものです。

### 特徴

- 電磁場でマイクロテスラ ( $\mu\text{T}$ ) の変化に感応
- 2軸で電磁場測定したものを解析して簡単に方向を見出す
- ソフトウェア調整後の測定で6ビット (64方向) の分解度
- 測定開始とデータ取得までの時間が30から40ms
- バスの対立を防ぐためにデータピンに保護用抵抗を内蔵
- 0.3インチのコンパクトな6ピンDIPパッケージなのでブレッドボード上で簡単使用
- 全ての BASIC Stamp、Javelin Stamp そして SX マイクロコントローラーに対応
- BASIC Stamp 2 シリーズの PBASIC コマンド SHIFTOUT、SHIFTIN、そして ATN を使って通信と方位計算が特に簡単にできます

### アプリケーションアイデア

- 自走ロボットや歩行ロボットなどの方向センサー
- 携帯用の電子コンパス
- 遠隔装置天候観測の風向指示計
- 眼の不自由な方のための音声コンパス
- 自動車の電子コンパス



## クイックスタート

### サンプルプログラムをダウンロード

この解説に載っているソースコード（12 頁から 25 頁）の中のサンプルプログラムは、パララックス社の web site の Hitachi HM55B Compass Module からダウンロード出来ます。

- ✓ [www.parallax.com/detail.asp?product\\_id=29123](http://www.parallax.com/detail.asp?product_id=29123) に行きます。
- ✓ “[Hitachi HN55B Compass Module Source Code \(.zip\)](#)” リンクをクリックしてファイルをダウンロードします。
- ✓ あなたの好きなフォルダーに解凍します。

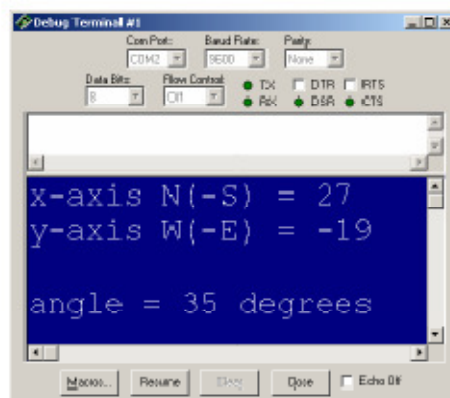
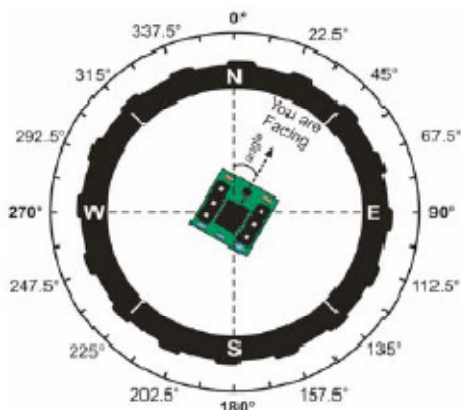
### コンパスモジュールの接続とテスト

このセクションは、パララックス社の BASIC Stamp 2® シリーズと SX マイクロコントローラーの為のテストインストラクションと接続が含まれています。

#### BASIC Stamp 2 シリーズ

BASIC Stamp 2、2sx、2p、2pe、又は、2px のどれかで Hitachi HM55B Compass Module をテストする場合は、このセクションにあるインストラクションを使って下さい。この最初のテストの目的は、モジュールの接続、機能、そして BASIC Stamp と正しく通信をしているかを確認するものです。この最初のテストはキャリブレーション（調整）の前なので、磁石の方位とモジュールの指示の間に大きな開きがあると思います。

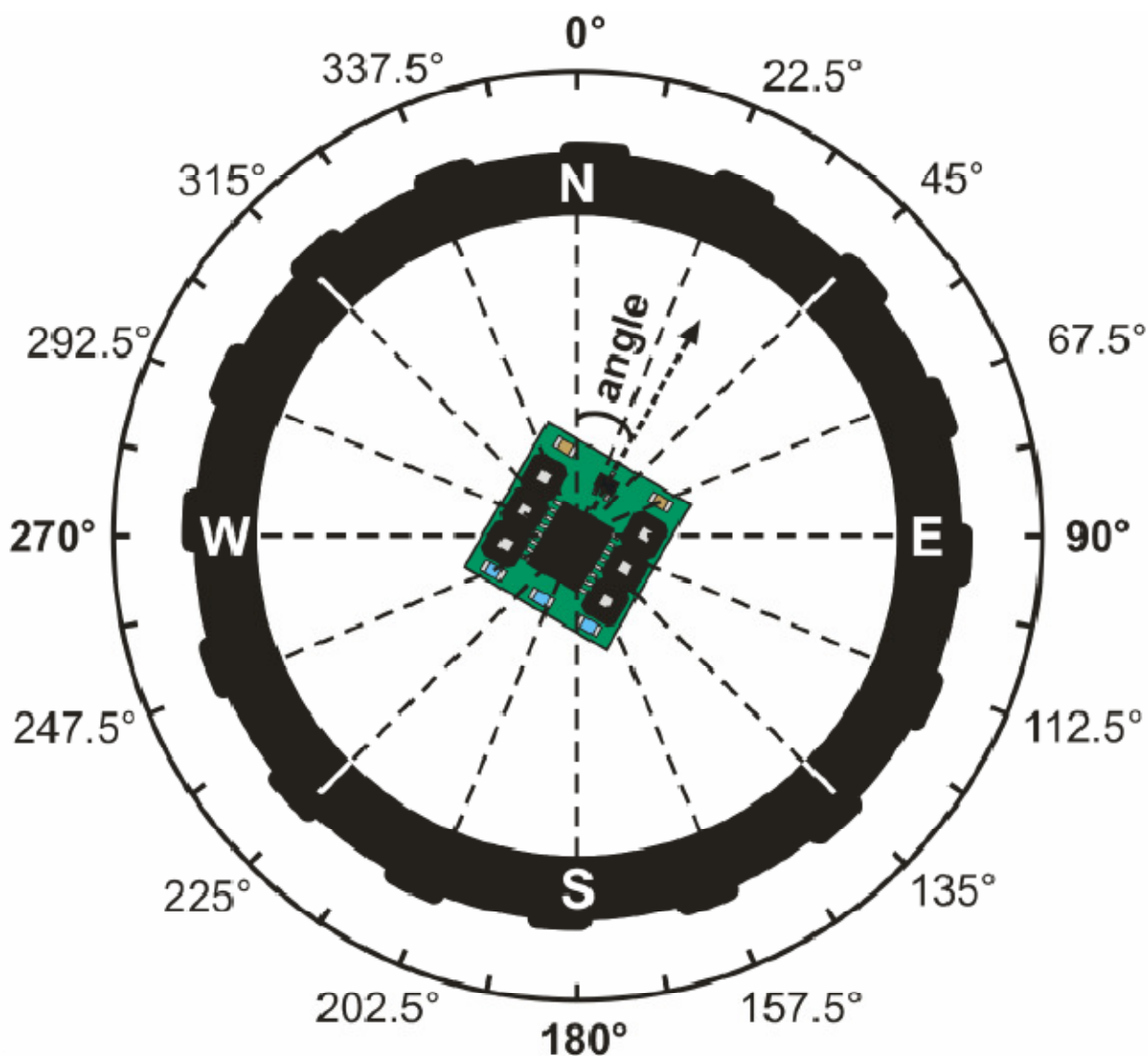
- ✓ 前ページの回路図に示された回路を組んで下さい。
- ✓ ボードを磁気が含まれたようなものから離れた水平の所に置いて下さい。一般に磁気が含まれた物に、モータ、棒状又は馬蹄形などの磁石、大きな鉄材などがあります。また、あなたのプログラム用ケーブルとパワーコードなどもセンサーの近くや周りがあると影響する事がありますので気を付けて下さい。
- ✓ BASIC Stamp エディター（v2.0 又はそれ以降）で TestHm55bCompass.bs2 を開いて下さい。必要に応じて、\$STAMP ディレクティブを指示して、プログラムを走らせて下さい。
- ✓ 下の図に見られるようにコンパスモジュールの方位検出をテストして下さい。そして挿絵のように測定角度を実際の方角と比較して下さい。



## キャリブレーション

キャリブレーションとテスト-キャリブレーションプログラムは、モジュールのプリント基板とそれを取り付けたボードやジャンパーワイヤ、電池などによる電磁場の影響を修正します。そしてこれは HM55B チップの感度やオフセットと歪みエラーなどもまた修正します。北の位置を決めるために方位磁石が必要でしょう。キャリブレーションの後、Hitachi HM55B コンパスモジュールは、北に対して 64 又はそれ以上の角度を正確に指示する事が出来るでしょう。

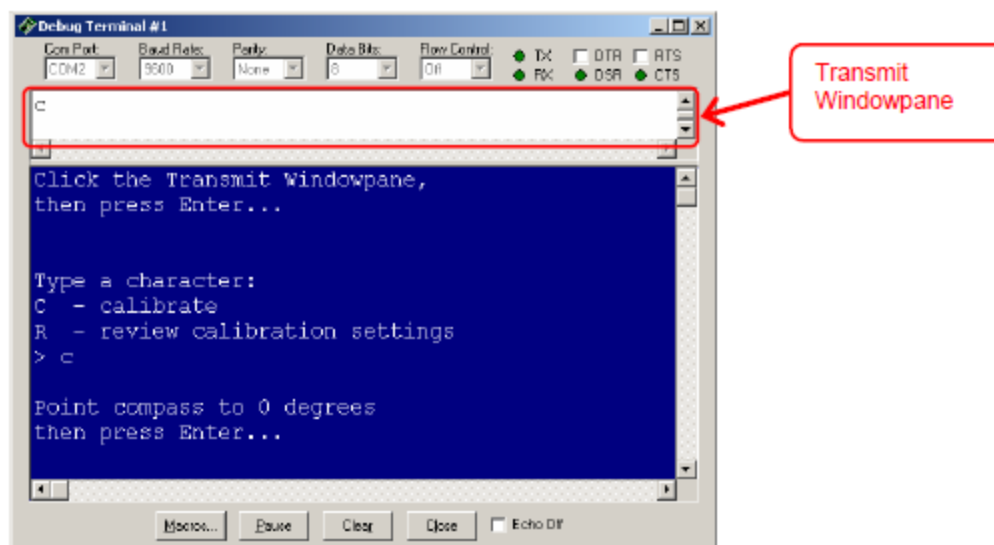
- ✓ 下に示すような 16 に分けた方位をプリントしたものを用意してください。
- ✓ 磁石を使ってそのプリントを北に合わせて下さい。
- ✓ コンパスのテストをする机の上にプリントを貼って下さい。



- ✓ 磁石はテストを続ける前に離れた場所に置いて下さい。
- ✓ 0° マークの所を通っている破線にプリント基板の端を揃えることによってコンパスモジュールを真っ直ぐに置いて下さい。

プログラム“CalibrateHm55bCompass.bs2”は、最初にコンパスオフセットを得るために 90 度ずつ測定する事を促します。それから、次にその中間、22.5 度ずつ測定します。両方の測定結果は、BASIC Stamp の EEPROM に保存されますので、TestCalibratedHM55BCompass.bs2 のプログラムはそれらのデータを使う事が出来ます。

- ✓ CalibrateHm55bCompass.bs2 を走らせて下さい。
- ✓ デバッグターミナルに応答する前に下図に示されているように Transmit windowpane の中をクリックしてください。それから“C”をタイプするとキャリブレーションが出来ます。
- ✓ キャリブレーションの処理が終わったら、TestCalibratedHM55BCompass.bs2 を走らせます。



CalibrateHM55BCompass.bs2 は、EEPROM にキャリブレーションの値を書き込みますので、例え電源を切った後でもそれらの値は残っていますので TestCalibratedHM55BCompass.bs2 用にデータを提供することが出来ます。TestCalibratedHM55BCompass.bs2 は、2 進数のラジアン (brads : binary と radians をくっつけた造語) と角度の両方で北からの角度を表示します。キャリブレーションの後でこのプログラムを走らせた時の角度に関する精度は、とても良くなっています。円を 360 に等分したものが角度で、brads (ブラズ : ラジアン) は、円を 256 に等分したものです。brads で北に対しては同じように 0 ですが、東、南、西はそれぞれ、64、128、192 となります。精度が最も大事なアプリケーションでは、測定の端数エラーがひとつ少ないので brad 測定をすると良いでしょう。より精度を高めるためには、キャリブレーションとキャリブレーションーテストプログラムの両方で数回の測定の平均をとるようにプログラムを変更すると良いでしょう。

### テストの感度

コンパスモジュールは、垂直になった 2 軸によって磁場を感知し比較する事によって働きます。磁場の強さは 1 から  $1.6 \mu\text{T}$  あたりの分解をするひとつの測定値に対応します。各軸の単位当たりのマイクロテスラを見付けるには、TestCalibratedHM55BCompass.bs2 を使ってあなたのコンパスモジュールで調べます。それは、軸の最大測定値を見付ける事から始めます。真北に X 軸を合わせ、それから最大値を見付けるために上向き、下向きに傾けます。あなたの地域での磁場の強さを比較してください。北米での総合強度マップは次の所から見付けることが出来ます：

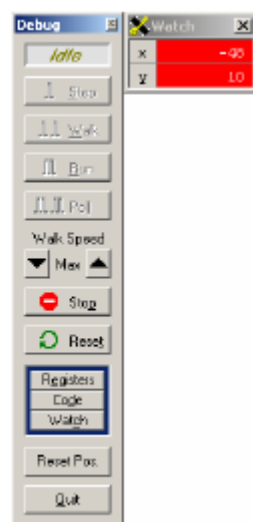
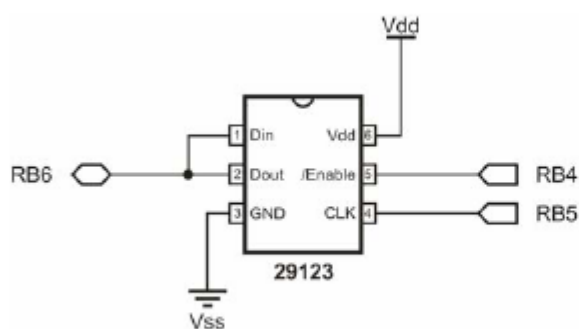
[http://nationalatlas.gov/articles/geology/a\\_geomag.html](http://nationalatlas.gov/articles/geology/a_geomag.html)

ナノテスラ (nT) の値を 1000 で割って、マイクロテスラ ( $\mu\text{T}$ ) に変換する必要があります。

## SX マイクロコントローラー

接続とテストの手順が SX-28 用に設定されているなら、SX シリーズのマイクロコントローラーならデバイスディレクティブをセットするだけで、どのチップでも働く筈です。SX-Key と SX-28 チップと一緒に、SX-Key v3.1 IDE v3.10 又はそれ以降を使う事を確かめてください。

- ✓ 下に示すような回路を組んで下さい。
- ✓ SX-Key IDE で TestHm55bCompass.SXB を開いて下さい。
- ✓ Save、Compile、そして Debug (CTRL-S、A、D) をして下さい。
- ✓ デバッグウィンドウの Poll ボタンをクリックしてください。もし Watch ウィンドウが見えない場合は、Watch ボタンをクリックして前部に持って来て下さい。
- ✓ モジュールをいろいろな方向に向けて Watch ウィンドウをチェックしてください。
- ✓ 北から反時計回りで、方角を  $\arctan(-y/x)$  を使って計算してください。



## 資料とダウンロード

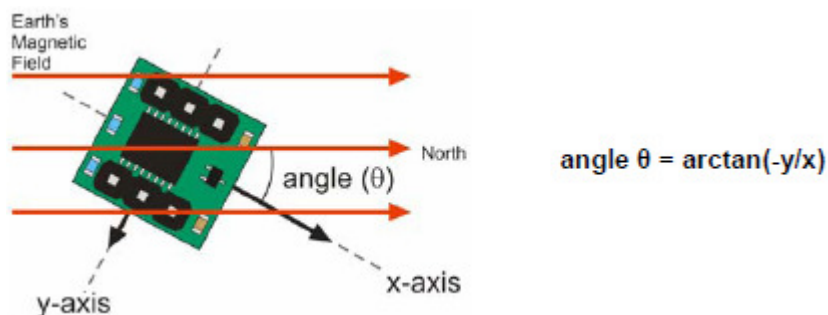
サンプルプログラムや HM55B データシートなど Hitachi HM55B Compass Module プロダクトページを参照してください：

[http://www.parallax.com/detail.asp?product\\_id=29123](http://www.parallax.com/detail.asp?product_id=29123)

## デバイスの情報

### 働き

Hitachi HM55B コンパスモジュールは、 $x$  と  $y$  の 2 軸を持っています。各軸はそれに平行する磁場の強さを知らせます。 $x$  軸は、(磁場の強さ)  $x \cos(\theta)$  を知らせ、 $y$  軸は、(磁場の強さ)  $x \sin(\theta)$  を知らせます。北から時計方向の角度に  $\theta$  を分解するには、 $\arctan(-y/x)$  を使います。PBASIC 言語 2.5 では、 $x \text{ ATN } -y$  を使います。ATN コマンドは、2 進数のラジアンで角度を返します。PBASIC で角度に変換するには、変数として保存してある 2 進数のラジアン測定値に  $*/360$  を当てるだけです。



コンパスモジュール上の Hitachi HM55B チップは、サイン付の 11 ビットの値でマイクロテスラ ( $\mu\text{T}$ ) の名称で  $x$  と  $y$  軸の測定値を知らせます。HM55B は、そのひとつの軸に対して平行な  $1 \mu\text{T}$  の北の方位の為に 1 の値を返します。もし磁場が南なら値は  $-1$  となります。但し、これらの値は名義上の値である事を記憶しておいて下さい。HM55B のデータシートによると、1 の測定のための実際の  $\mu\text{T}$  値は、1 から  $1.6 \mu\text{T}$  の範囲となっています。また、負の 11 ビット値は、mask を適用しない限りワード変数に負の値が現れない事も記憶しておいて下さい。例えば、ビット 10 が 1 の時、テストとキャリブレーションプログラムの中で mask でビット 11 から 15 もまた 1 に変わります。

HM55B に接続するマイクロコントローラーは、enable と clock 入力をコントロールしなければいけません。そして、そのデータ入力と出力ピンから測定した軸の値を得るべく同期シリアル通信を使わなければいけません。例えば、BASIC Stamp 2 は、コンパスモジュールの enable ラインの HIGH/LOW をコントロールするようにプログラムする事が出来ます。そして、デバイスをリセットする値を送り、SHIFTOUT コマンドで測定をスタートさせる事が出来ます。SHIFTOUT コマンドは、データ入力に対してデータビット値を送るように、コンパスモジュールのクロック入力をコントロールします。SHIFTOUT の逆は、SHIFTIN で、デバイスのデータ出力ピンによって送ったデータビットを纏めるようにデバイスのクロック入力をコントロールします。

要求された測定を遂行するのに HM55B は 30 から 40 ms かかります。マイクロコントローラーは、その間、他の仕事をするか、或いは、測定が完了するまでに送信されるメッセージを受け取る用意をします。この準備は、状態を問い合わせる SHIFTOUT コマンドの組み合わせと、状態を得る SHIFTIN コマンドです。SHIFTIN コマンドは、測定が完了したという事を知らせる状態フラグを受け取ると、2 番目と 3 番目の SHIFTIN コマンドが変数の中に 11 ビットの  $x$  と  $y$  軸の測定値を保存します。

### 予め注意する事

- ピンの定義と許容のセクションに述べられている範囲を超えてデバイスに電圧を加えない事。
- コンパスモジュールを強力な磁場がある所で動作させたり保管したりしないでください。強力な磁場は、棒状又は馬蹄形の磁石、電気モータ、ソレノイドのようなコイルを持ったもの、リレーや大きな誘導子などによって作られます。

- コンパスモジュールに  $300 \mu\text{T}$  以上の磁力を与えないでください。 $300 \mu\text{T}$  以上の磁力を与えるとセンサーがダメージを受けます。
- 可能な限り磁場を乱すようなものから離れた位置にコンパスモジュールを取り付けて下さい。それらは通常の方位磁針を含んだ磁石、モータ、パワーコード、コイル、鉄の箱、そして場所によりますが地面などで影響します。

## デバイスのインフォメーション

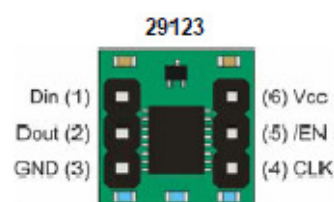
### 規格

Symbol	Quantity	Minimum	Typical	Maximum	Units
$B_{SE}$	Sensitivity <sup>†</sup>		1.0	1.6	$\mu\text{T}/\text{lsb}$
H	Linear measurement range <sup>†</sup>	-180		180	$\mu\text{T}$
d $\theta$	Individual axis offset			36	°
$T_{CONV}$	Conversion time <sup>†</sup>		30	40	ms
$T_{OPR}$	Operating temperature	0		70	°C

† From Hitachi HM55B Datasheet

### ピンの定義と許容

- |          |   |                          |
|----------|---|--------------------------|
| (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         |



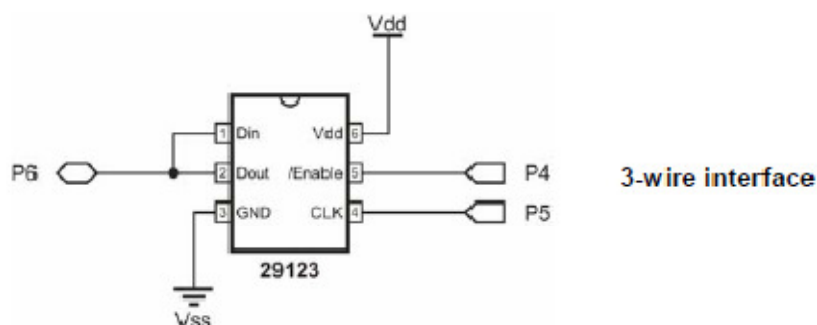
Symbol	Quantity	Minimum	Typical	Maximum	Units
Vcc	Supply Voltage	4.8	5.0	5.2	V
Icc(Ave)	Average active supply current <sup>*</sup>		5	7	mA
Icc(Pk)	Peak instantaneous current <sup>**</sup>		30	45	mA
Icc(Sb)	Standby supply current		2	3	mA
GND	Ground reference connection		0		V
V <sub>OH</sub>	Signal high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

\* Measurement cycle = 80 ms

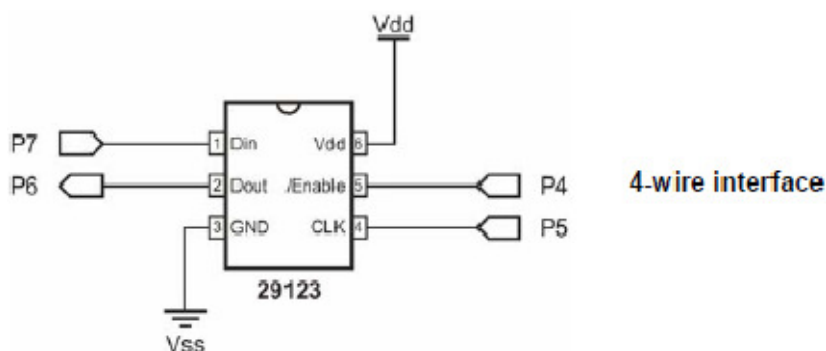
\*\* Typical duration is 5  $\mu\text{s}$

## 接続図

3本のワイヤでのインターフェイスを殆どのアプリケーションでお勧めします。ここに見られる接続は、特有の I/O ピンで、/EN ピンだけは専用の I/O ピンが必要です。Din/Dout ピンは他の同期シリアル機器とラインを共有する事が出来ます。クロックラインも同様に他のデバイスとラインを共有できます。



Din と Dout ピンは、一緒に接続しなくても構いません。それらもまた、独立してコントロールする事が出来ます。これは通信ラインを他の同期シリアル機器と専用の入力ラインと出力ラインを持つことによって共有を可能にします。



## コマンドセット

これらのコマンドはコンパスモジュールに Shifted-out されるものです。(SHIFTOUT のコマンドを使って送り出されます)

Binary Value	Quantity
0000	Reset device
1000	Start measurement
1100	Report measurement status (and transmit the measurement if it's ready)

\*注意：英文テキストの中の Start と Report の Binary Value は、間違いでこの表が正しいものです。



### ステータスフラッグ

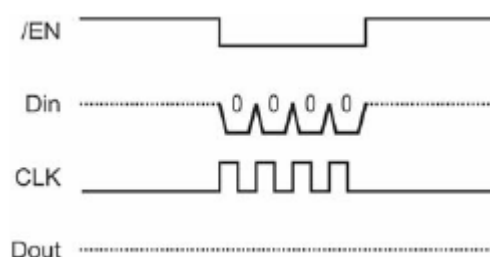
コンパスモジュールは、測定状態の要求コマンドに対して次のような応答をします。

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

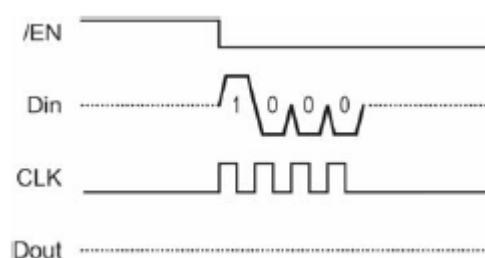
### 通信プロトコル

コンパスモジュールとの送受信に関する全ての値は、**most significant bit first**（最も上位のビットを最初に扱う）です。ビット値はクロック信号の立ち上がり後に有効です。これは、**BASIC Stamp 2** で **SHIFTOUT** コマンドのモードアークメントが **MSBFIRST** で、**SHIFTIN** コマンドのモードアークメントでは、**MSBPOST** をセットするという事になります。

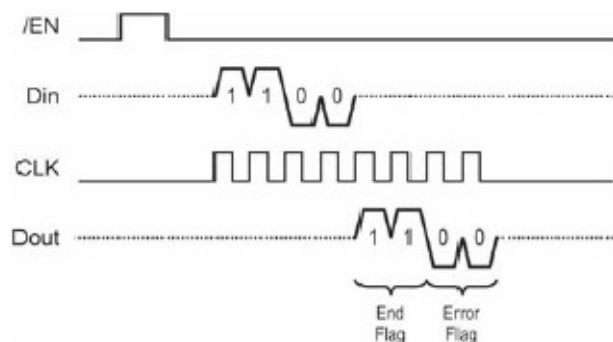
HM55B をリセットするには、**/EN** をハイからローにして、**%0000** をシフトアウトし、それから **/EN** を再びハイにします。



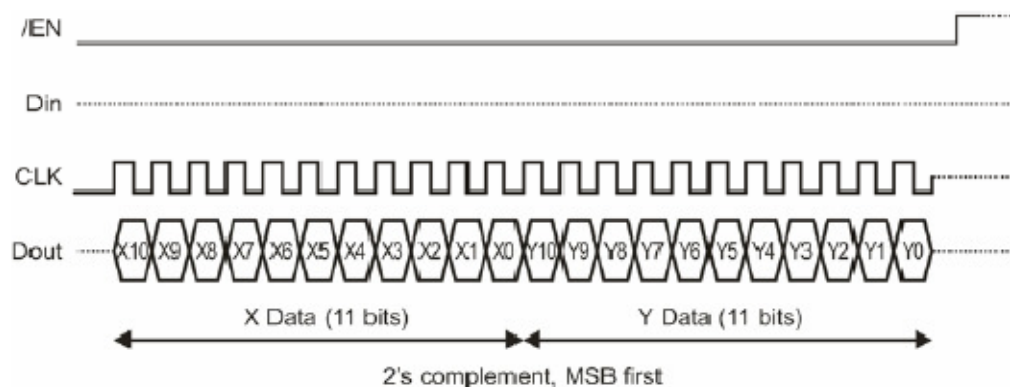
リセットの後、**/EN** が再びローになり測定が始まり、それから **%1000** がシフトアウトされます。測定の状態をチェックするまで **/EN** をローのままにします。



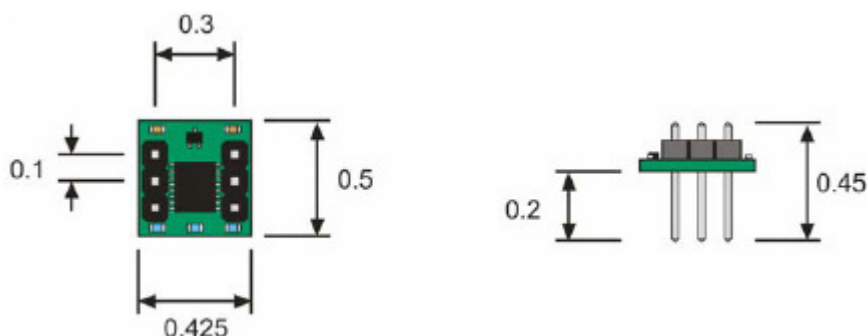
測定の状態をチェックするには、 $\overline{EN}$  に正のパルスを送る事から始めます。それから%1100 をシフトアウトして、ステータスフラッグをシフトインします。(SHIFTIN のコマンドで信号、ステータスフラッグを受ける) 測定が続いている間は、end flag と error flag の両方とも 00 になっています。コンパスモジュールは測定が完了する (end flag が 11 に変わる点) まで状態を繰り返し問い合わせます。%1100 を受け取ると (end flag が 11 で error flag が 00)、状態の問い合わせは終わります。 $\overline{EN}$  をローにしたまま、x と y 軸の値をシフトインで得ます。



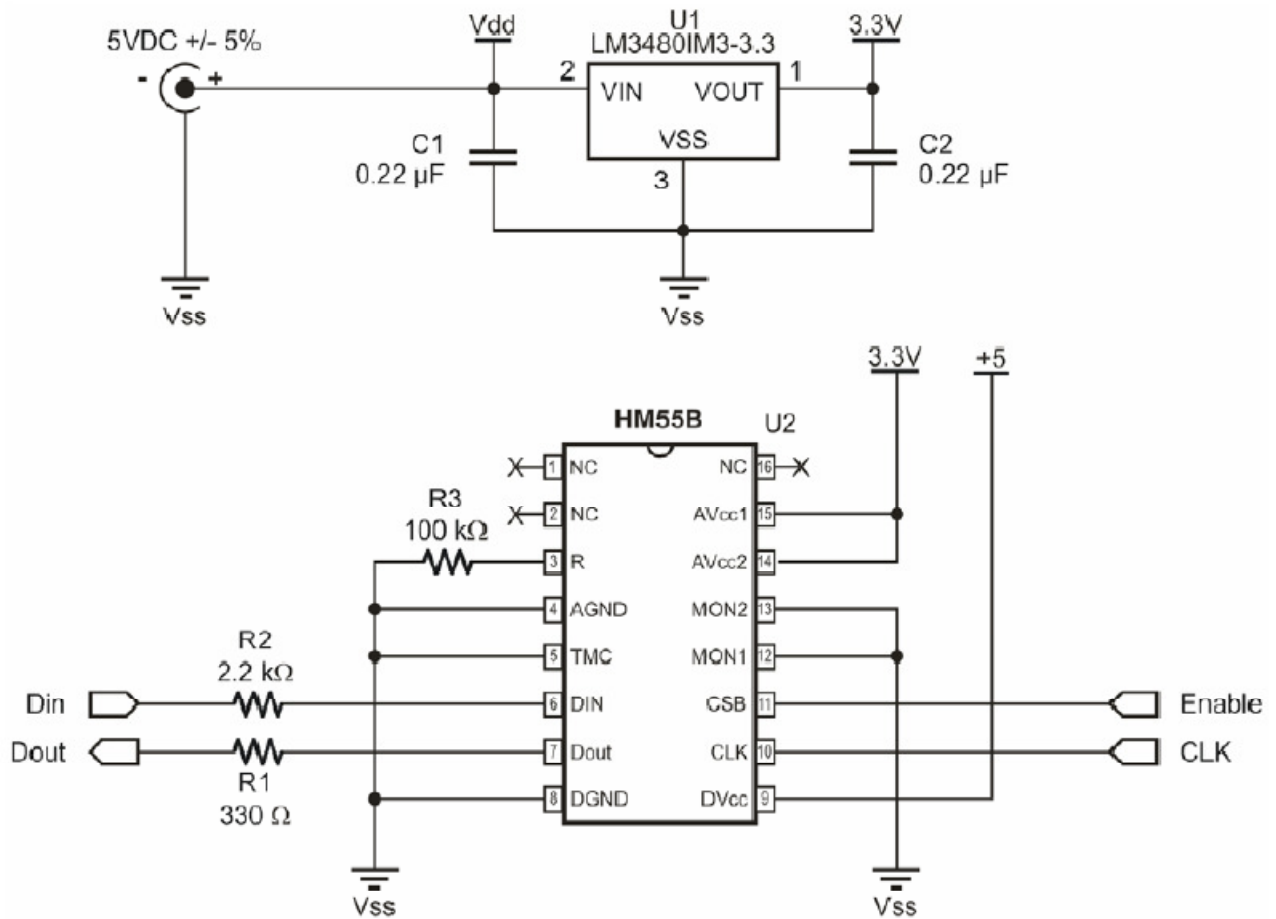
x と y 軸の値をシフトインする事は、単に x 軸の測定値の 11 ビット、そして、それに続いて y 軸の測定値の 11 ビットをシフトインする事です。y 軸のシフトインが完了した後、再び $\overline{EN}$  ピンをハイにセットします。



### モジュールのサイズ (単位はインチ)



## モジュールの回路図



## ソースコード

## BASIC Stamp® 2 シリーズ用テストプログラム

もし BS2e、BS2sx、BS2p、BS2pe、又は、BS2px を使用する場合は、プログラムを走らせる前に\$STAMP ディレクティブで設定する事を忘れないでください。

```

' =====
' 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 ]-----

D0           ' 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 ]-----

```

ソースコードの続き

```

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]
SHIFTTIN DinDout,clk,MSBPOST, [Status\4] ' Get Status
LOOP UNTIL status = Ready  ' Exit loop when status is ready

SHIFTTIN 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

```

ソースコードの終わり

## BASIC Stamp 2 シリーズのキャリブレーションプログラム

```

' =====
'
' 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 ]-----

```

キャリブレーションプログラムの続き

```

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

```

## キャリブレーションプログラムの続き

```

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

```

キャリブレーションプログラムの続き

```

    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

```



キャリブレーションプログラムの続き

```
' 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
```

キャリブレーションプログラムの続き

```

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]
  SHIF TIN DinDout,clk,MSBPOST,[Status\4]      ' Get Status
LOOP UNTIL status = Ready                       ' Exit loop when status is ready

SHIF TIN 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

```

キャリブレーションプログラムの終わり

## BASIC Stamp® 2 シリーズ キャリブレーション—テストプログラム

```

' =====
'
' 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

```

## キャリブレーションテストプログラムの続き

```

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 = ",
           DEC angle, " brads", CLREOL ' ... in brads
  angle = angle */ 360            ' Convert brads to degrees
  DEBUG CR, "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]

```

## キャリブレーションテストプログラムの続き

```

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]
  SHIFTTIN DinDout,clk,MSBPOST,[Status\4] ' Get Status
LOOP UNTIL status = Ready ' Exit loop when status is ready

SHIFTTIN 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:

```

## キャリブレーションテストプログラムの続き

```

' 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

```

## キャリブレーションテストプログラムの終わり

## ソースコード

## SX マイクロコントローラー テストプログラム

```

' =====
'
' 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

```

## SX マイクロコントローラー テストプログラムの続き

```

' -----
' 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

```



## SX マイクロコントローラー テストプログラムの続き

```
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
```

## SX マイクロコントローラー テストプログラムの終わり