



Scanned and converted to PDF by HansO, 2002

温度, 音, 光 に感じる

センサーキット



温度の変化を感じる
サーミスタ



音をとらえる
コンデンサ・マイク



光を鋭い指向性で感知する
フォト・トランジスタ

3種のセンサーがとらえた周囲の変化を、リアルタイムにパソコンに送り込む、実験派待望のカートリッジです。
楽しみながらセンサーのことを学べる、Dr. Dのユーモラスな「センサー講座」のオートデモも入っています。
測定結果をグラフ化し、さらにプリンターで打出せる「サンプル・プログラム」を動かすことができます。
BASIC言語でセンサーの測定内容を読みだしたり、その結果を外部に出したりすることもOK! 君の知恵と工夫しだいで、さまざまな実験に取り組むことができます。



※詳しいごくりの取扱いが考えられる本製品ですが、不注意な扱い方をすると、製品を壊したり/パソコン本体を壊したりする虞
色もあります。必ず「取扱い説明書」をお読み下さい。

標準価格 6,800円

RAM 64k
VRAM 128k **MSX2**



1 センサーキットをカートリッジ・スロットに挿込み、電源をONにすると、この画面が表示される。



2 オートテラを渡すときはまるくD/Dにも、「センサー」種類の画面。



3 温度センサーもやはり「ハックション」。



4 音センサーに大声で話しかけると「ウーサーイ！」



5 光センサーに突然光を当てると「ピッカリ！」



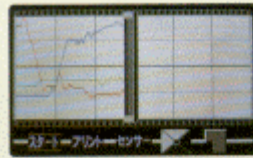
6 「サンプル」を起動させた画面。センサーの感度も変えられる。



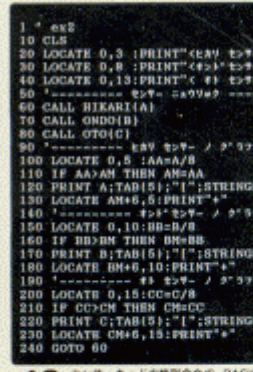
7 一定の値になったときに、アラーム音を鳴らす設定もできる。



8 測定結果をリアルタイムにグラフに描くことも可能。

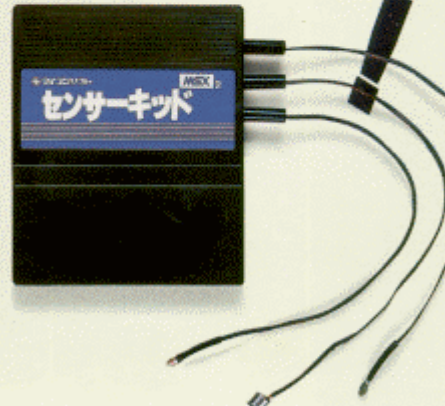


9 拡大グラフモードもある。



10 センサーキットの特別命令で、BASIC実行で自分で独自のプログラムもOK!

機能いっぱい！
いろいろな実験、工作に活用しよう！



・お使いになる前に、取扱説明書を必ずお読みください。
・万一当社の製造上の原因による不良があった場合は、新しい製品とお取りかえします。それ以外の責はご容赦ください。
・このプログラムは、無断で複製、複製、貸貸業その他の営業目的に使用することはできません。

この製品は MSX2 のマークのついていないコンピューターでは使用できません。MSX マークは、アスキー社の商標です。本製品を使用したことによるお客様の損害あるいは第三者からのいかなる請求についても、当社はその責任を負いかねます。

温度、音、光に感じる
センサーキット



T4901780084852

発売元：電通新報社 東京都品川区東五反田1-11-15 電話(445)0111

標準価格 **6,800円**

“センサーキット” マニュアル

丹 治 佐 一

センサーキットをお買い上げいただき、ありがとうございます。

お使いになる前に、このマニュアル（説明書）をお読みください。本書には、基本的な使い方の他に、プログラム例、使用上の注意も掲載してあります。

創意工夫することでさらにおもしろく、発展した使いかたが可能です。初めは、プログラム例を打ち込んで実行してみてください。

それを改造して、オリジナルのプログラムやハードウェアを作ってみましょう。

センサーキットの特徴

- 特徴は以下の通りです。
- 付属のセンサーとして、温度センサー、音センサー、光センサーがあり接続できる。
 - 同時に三つのセンサーが利用できる。
 - A/Dコンバータにより、センサーのアナログ入力をデジタル・データ(0~255)に変換可。
 - 内蔵ソフト（オートデモ、サンプルプログラム）により、リアルタイムでセンサーの状態がわかり、センサーの学習ができる。
 - サンプル・プログラムを利用することで、センサー情報のグラフが描け、状況の変化が一目でわかる。また、設定値によりアラームを鳴らすことができ、簡単な監視システムに利用できる。
 - センサーの情報がBASICで利用できる

専用のコマンドを内蔵。これにより、オリジナル・プログラムでのセンサー活用が可能。

- 外部出力端子を2系統装備。専用のコマンドによりコントロールできる。
- センサー入力と外部出力を組み合わせることで、画面内だけではなく、外部の機器を動作させるなどの実用的利用が可能。

まさにセンサーキットは、自分単独で動作するパソコンから、外部の状態に応じて動作する「感覚を持ったパソコン」に変身させるものなのです。

センサーキットの扱いかた

扱う上で基本的なことですので必ず守ってください。守れないと“センサーキット”だけではなく、パソコン本体も傷めてしまいます（第1図）。

①MSX/MSX2の電源を切った状態で“センサーキット”を差し込む

カートリッジの方向を確かめて、必ずパソコンの電源を切った状態で差し込みます。

②センサーをカートリッジに差し込む
カートリッジの上から順に“温度センサー（サーミスタ）”、“音センサー（マイク）”、“光センサー（フォト・トランジスタ）”を差し込みます。

抜き差しは、プラグの部分を持って行い、センサーを引っ張って抜かないようにします。

【注】センサーはデリケートなので、ていねいに扱います。

③パソコンの電源を入れる

準備が良ければ、パソコンの電源をONにしましょう。内蔵ソフトがスタートし、メニュー画面が現れます。

④電源を切ってから、カートリッジを抜く

センサーキットを楽しんだ後、カートリッジを抜くときは必ずパソコンの電源を切った上で行います。

メニュー画面

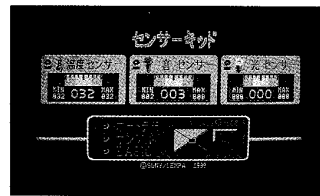
電源投入時に出る画面がメニュー画面です（写真1）。

画面の上部には、各センサーからの情報がリアルタイムで表示されています。

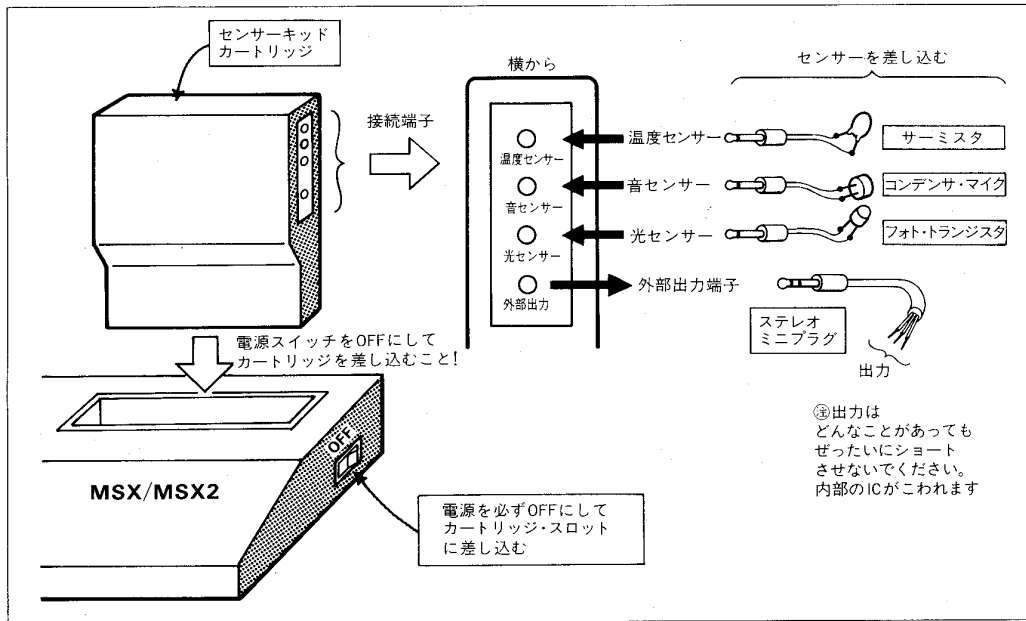
●メーターの内容

左のメーターから温度、音、光の順で並び、入ってくるセンサー値が大きいかほどメーターが右に振れます（第2図）。

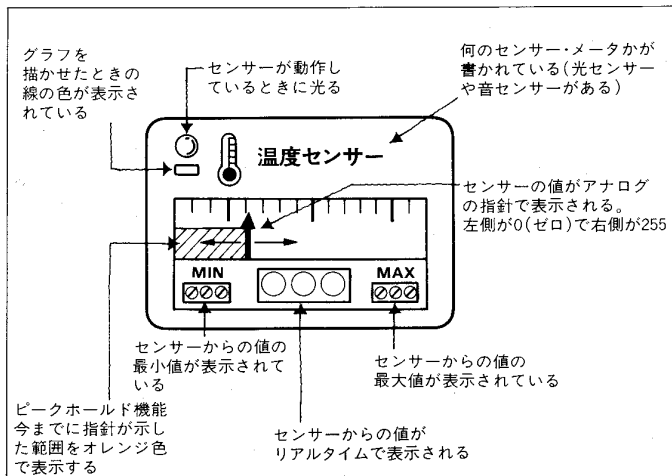
メーターには“ピークホールド”機能が付加されています（オレンジ色で表示される）。これは、入ってきた値の最大値（ピーク）を覚えておく機能です。これにより、どれぐらいの値が入ってくるかが事前にわかるので、センサ



〈写真1〉センサーキットのメニュー画面



《第1図》センサーキットの使いかた



《第2図》メーターの読みかた

ーを利用したプログラミングやサンプル・プログラムに便利に活用できます。

また、メーター内にはセンサー値の最小値(MIN)と最大値(MAX)が表示されています。

●メニューの選びかた

画面下には、メニューが表示されています。上から“オートデモ”、“サンプル・プログラム”、“BASIC”と順にならび、“BASIC”のところまで赤く点滅していると思います。

メニューの選びかたは、カーソルキーの上下で赤い点滅を動かします。

“RETURN: []”キーを押せばスタートします。

●メニューの内容は?

▶“オートデモ”は、Dr.Dがセンサーの働きを順に説明し、自分でセンサーを操作して、その働きを知ることができます(写真2)。

▶“サンプル・プログラム”は、センサーの情報を元に、グラフを描かせたり、設定した値でアラームを鳴らすこともできます。

▶“BASIC”は、BASICプログラムが楽しめるモードですが、センサーキット



《写真2》オートデモの画面

では、センサーの情報を自分のプログラムで生かすための専用コマンドが付加されています。

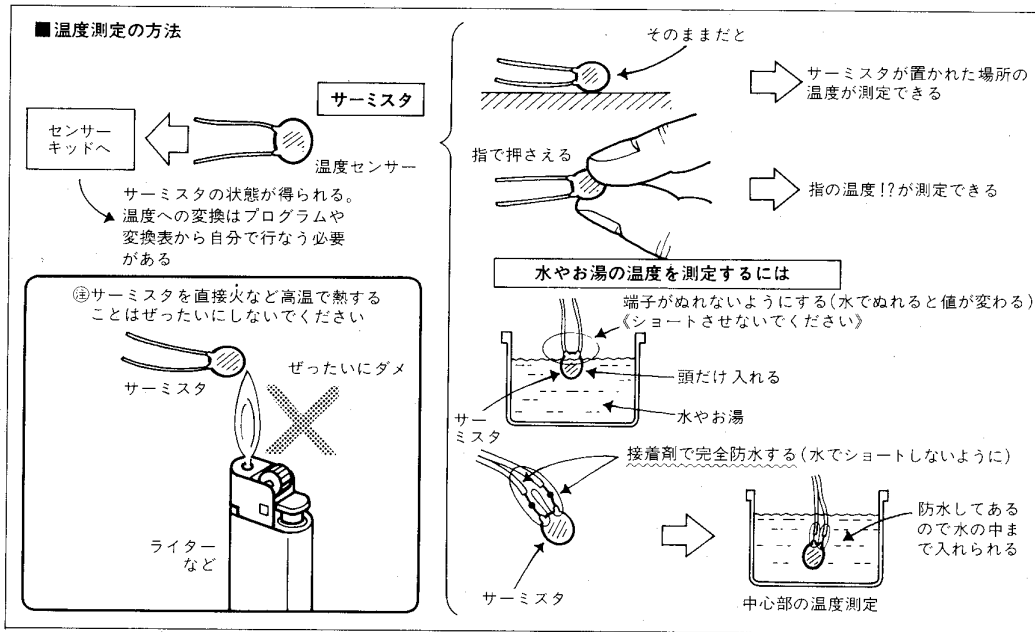
オートデモ・プログラム

電源を入れた状態にしておくと、“オートデモ”が自動的にスタートします。電源投入時に出るメニューでオートデモを選んで動作します。

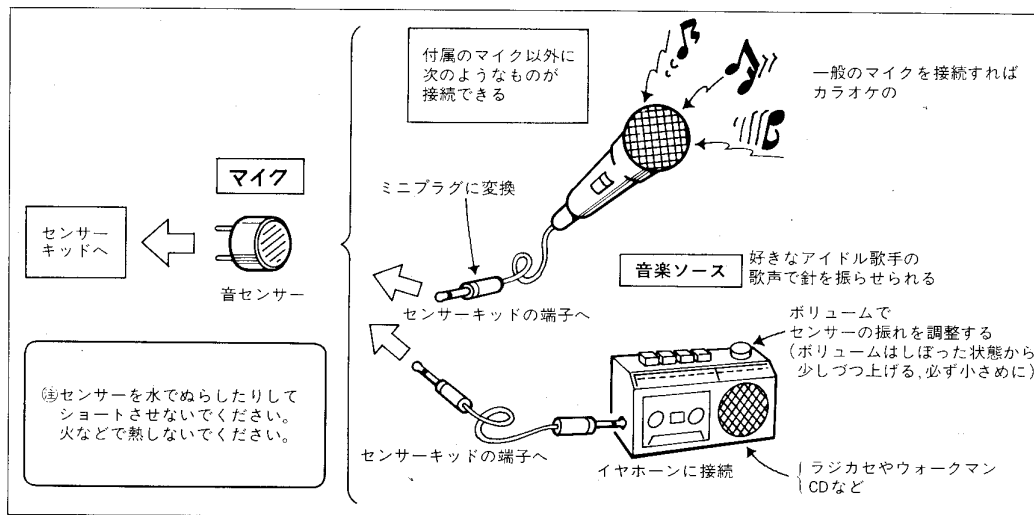
スタートするとパソコン博士Dr.Dが現れ、センサーの講義が始まります。“スペースキー”を押すと開始します。そのままにしておいたり、“RETURN”キーを押すと、メニューに戻れます。

●最初の講義は、温度センサーです。カートリッジの上から1番目に差し込んだ温度センサー(サーミスタ)を手で持ったりして、温度変化を与えてみてください(第3図)。

温度が上がるにつれて値が上がっていきます。つまり、温度が低いほど値が小さくなり(0に近づく)、高いほど



《第3図》温度センサーの使いかた



《第4図》音センサーの使いかた

値が大きくなります(255に近づく)。

●2時限目の講義は、音センサーです。カートリッジの上から2番目に差し込んだ音センサー(マイク)に向かって、音を入れてみましょう(第4図)。

音が大きいほど値が大き(255に近づく)、小さいほど値が小さくなります(0に近づく)。

●最後の講義は、光センサーです。センサーキッドの3番目に差し込んだ光センサー(フォト・トランジスタ)を光に当てたり、手でおおって隠した

りして、光の変化を与えてみましょう(第5図)。

光が当り、明るいほどセンサーの値が大きくなります(255に近づく)。逆に、暗いほど値が小さくなります(0に近づく)。

このように、センサーキッドでは、センサーの状態が値として得られることがわかります。

講義が終了するとメニュー画面に戻ります。

サンプル・プログラム

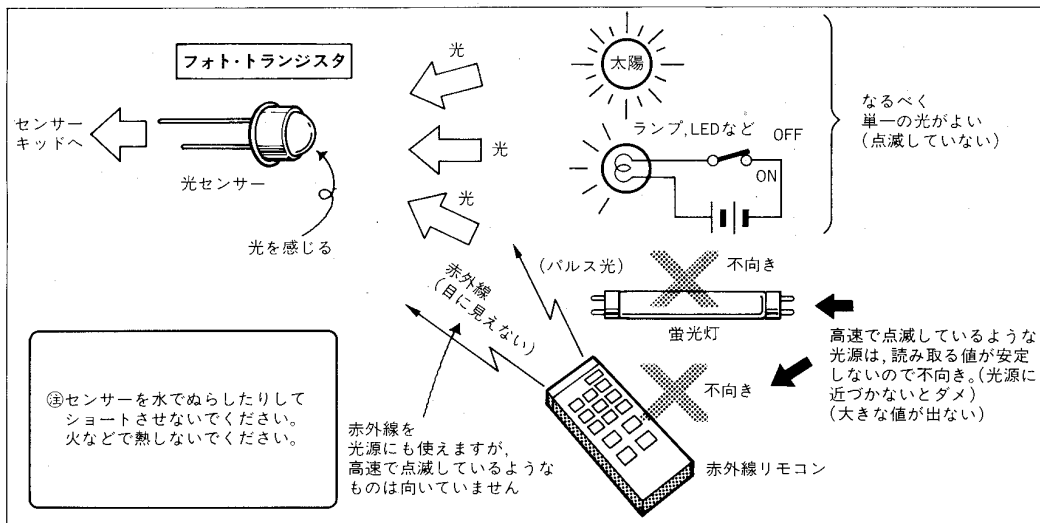
メニューの上から2番目は、サンプル・プログラムです。

●サンプル・プログラムのメニュー

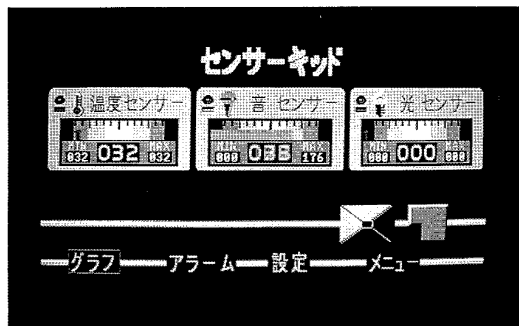
サンプル・プログラムがスタートすると、メニューが現れます(写真3)。

[グラフ]は、センサーの値でグラフを描くモードです。時間変化でセンサーの状態がチェックできます。

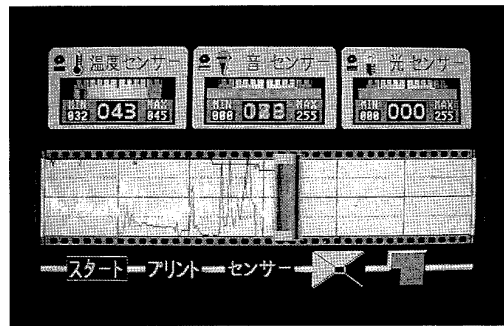
[アラーム]は、センサーの値が設定



《第5図》光センサーの使いかた



《写真3》サンプルのメニュー画面



《写真4》グラフを描いているようす

値を通過したとき、アラーム音を出すことができるものです。

〔設定〕は、サンプル・プログラム内の各種の設定をするためのモードで、グラフの拡大縮小設定、感度設定、センサーのON/OFF、最小/最大値の設定ができます。

〔メニュー〕は、サンプル・プログラムを終了し、電源投入時のメニューに戻ることができます。

これらは、カーソルキーの左右で選び、RETURN (□) キーで実行を開始します。

●〔グラフ〕を選んだら！
このモードは、センサーの情報を元にグラフが描けます(写真4)。

画面上部には、センサーのメーターが並び、中央にはグラフを描くためのプロッターが見えます。

▶〔スタート〕を選び“RETURN”キーを押すと、プロットが開始されます。このとき、描かれるグラフの色は、赤色が温度センサー、音センサーが

緑色、光センサーが青色です(各メーターの左上の帯の色がそのグラフの線の色を表します)。

描く速度は、“設定”の感度の値で変わり、同様に拡大を選んでおくとプロッタ画面のみになりグラフが大きく描かれます。

また、描く範囲は、メーター上にあるMIN/MAXの値をもとに自動的に計算され、フル・スケールで描くようになっています。

▶〔プリント〕を選ぶと、結果が印字されます。

▶〔センサー〕は、サンプル・プログラムのメニューに戻ります。

●〔アラーム〕を選んだら！
このモードは、センサーの値が、設定値を通過したときアラーム音を出すことができるモードです(写真5)。

このモードで一度設定しておくと、グラフを選び、描いている間でも、設定値を越えるとアラームが鳴ります。設定方法は、カーソルの左右キーで

設定したいセンサー部分まで赤い枠を動かします。

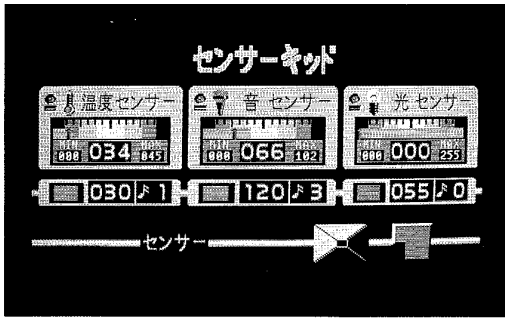
メーターの各センサー値の真下にある数字は、鳴らすセンサーの値です。ですから、メーターに表示されている値以上の値を設定してください。その値を越えたときに鳴ります。もし、小さければ鳴りません。

値の入力方法ですが、設定値の所で“RETURN (□)”キーを押し、カーソルの上下で数値を変えするか、数値の頭から入力してください(注：数値の頭に付くゼロは省略できません)。良ければRETURNキーを押します。

たとえば、35に設定するなら035□と入力するわけです。5なら005□ですね。

値の設定ができたら、アラーム音を設定します。カーソルの右矢印で右隣の音符マークに移します。

RETURNキーを押すと入力モードになります。0のときは鳴りませんが、1、2または3のときアラームが鳴りそれぞれ音が違います。



《写真5》アラームも設定できる

▶【センサー】を選ぶと、メニューに戻ります。(カーソルの上下を押すと、すぐにセンサー部分にカーソルが移せます)。

〔設定〕を選んだら！

グラフを描くときの各種の設定が行えるモードです(写真6)。

カーソルキーの左右矢印で、赤い枠(カーソル)を動かして、RETURN (□) キーを押すことで入力モードになります。

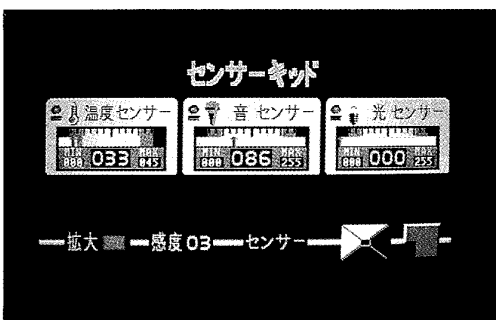
▶【拡大】モードにするとグラフモードの時に画面いっぱいに描かれます(写真7)。

拡大にカーソルを移し、RETURN キーを押すと、拡大(赤く光る)/普通の設定ができます。

▶【感度】は、グラフを描くスピード(時間)を変えるものです。

値は0から15まで選べ、値が大きくなるほど反応が鈍くなり、長時間にわたって測定し描かれます。

▶最大/最小、センサーON/OFFの設定グラフは、すでに入力されているセンサーの最大/最小値を元に描かれますが、最大値が変わったり、最小値が変わったときは、自分でその値を設定できます。



《写真6》設定時の画面

設定前は、自動的にMIN/MAXが変化しますが、ここで設定すると、その値で固定されます。

カーソルの左右矢印でセンサーの各メーターの中のMIN/MAXの部分に動かしします。

RETURNキーを押すと入力モードになります。MINが最小値、MAXが最大値です。入力後RETURNキーで設定終了です。

センサー情報をグラフに描きたくないときは、センサーをOFFにできます。

カーソルを左右の矢印キーで動かし、各センサーの左上の部分に合わせます。ここで、RETURNキーを押すことでON/OFFが指定できます(赤く光っているときに動作)。

▶【センサー】を選ぶと、サンプル・プログラムのメニューの画面に戻ります。

センサーってなんだろう？

わからないうちにセンサーを使ってきたわけですが、“センサー”っていったい何でしょうか？

人間は、見たり、聞いたり、味わったり、臭いを嗅ぐことができ、さわるでも質感や温度などを感じ取ることが可能です。でも、機械はそうはいきません。

人間なら暑くなったら、エアコンを入れるとか、暗くなったら電灯をつけるなどといったことは、すぐにでもできますが、機械だとだめですね。

暑い/寒いという温度を感じるこ

とができないし、明るいとか暗いといったことがわからなければ、電灯をONにすることもできません。

そこで、温度を感じたり、明るさを感じる“センサー”というものが考え出され、実用化されているのです。

エアコンに入っているものは、温度を感じる温度センサー、街灯に付いているものは光を感じる光センサーです。

ほかにも、身の回りにはセンサーがいっぱい使われています。こたつやアイロン、冷蔵庫には温度センサーが、カメラには光センサーが、電子レンジでは、出来上りを感じる？センサーが活躍しています。そのほか見えない所で人間に変わって活躍しているのです。

身の回りのセンサーを探して見ませんか？

センサーキットのセンサー

センサーキットにはそんなセンサーの一部が付いています。

温度を感じる“温度センサー”、音を感じる“音センサー”、光を感じる“光センサー”です。

●温度センサー

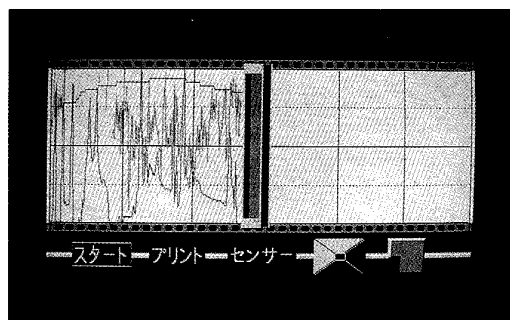
温度を感じるセンサーとしては、“サーミスタ”が使われています。

このサーミスタというのは、温度によって抵抗値(電流の流れにくさ)が変化するものです。センサーキットの回路ではその抵抗値の変化によって起こる電圧変化をとらえて、A/D(アナログ/デジタル)コンバータで値に変換しています。

●音センサー

音の強さを感じるセンサーとして、センサーキットでは、“コンデンサー・マイク”が使われています。ラジカセなどに付いているマイクと同様なものです。

このコンデンサー・マイクは、電圧



《写真7》拡大のグラフのモード

を加えないと動作しません。マイクをみると2芯で、電圧端子がありませんが、実は音声信号の出力端子に動作させる電圧を加えているのです。

また、センサーキットの内部では、アンプ回路が入っていて、微小な音声の電圧変化を増幅して、A/Dコンバータに入力しています。

●光センサー

光を感じるセンサーとして“ホト・トランジスタ”が付いています。

これは、トランジスタの仲間に入りますが、通常のトランジスタと異なり、トランジスタの動作が光の明るさで決まります。

トランジスタには、エミッタ、コレクタ、ベースという三つの端子がありますが、入力であるベースの電圧は、

フォト・トランジスタでは、光の強さによります。

つまり、光の強さによって、トランジスタの動作が変わるわけです。よって、弱い光でもトランジスタの増幅動作によって動作、光の変化を大きくとらえることができます。

センサー出力を自分のプログラムで生かす方法

センサーキット・BASICコマンド

センサーがどんなものかわかったところで、実際にセンサーの情報を利用してみたいことにしましょう。

センサーキットの内蔵コマンド

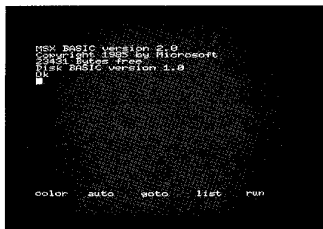
この“センサーキット”では、内蔵のデモ・プログラムやサンプル・プログラムの他に、センサーの情報を自分のプログラムで利用することができます。

センサーキットがスタートしたときのメニュー画面で“BASIC”を選びます(カーソルキーの上下で“BASIC”を選びRETURN(Enter)キーでBASICに入る)。

これでOK!センサーキットの内蔵コマンドがBASIC中に付加されています(写真8)。

こんなコマンドがあるんだ

コマンドは全部で四つ。温度センサー、音センサー、光センサーの状態がわかるコマンド。キットからの情報で接続機器が動かせるコマンドと、キットのセンサーをプログラムで有効に利用



《写真8》BASICを立上げたところ

できます(第6図、第7図)。

●温度センサーの情報を得る

暖かさ/冷たさがわかるコマンドです。

CALL ONDO (数値変数)

↑

温度と読む

温度センサー(サーミスタ)からの暖かさ情報が、()内の数値変数に入ります。

結果は0~255までの値が得られ、冷

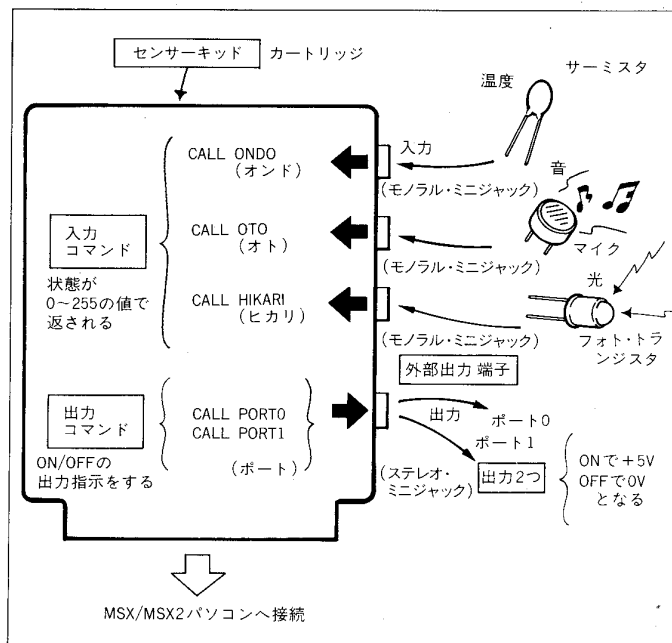
たいほど値が0に近付き、暖かくなるにしたがって255に近付きます。

注意として、温度センサーからの情報結果を“温度”として直読できません。温度が約-25℃から約90℃の範囲が数値として0から255の数値に変換されます。

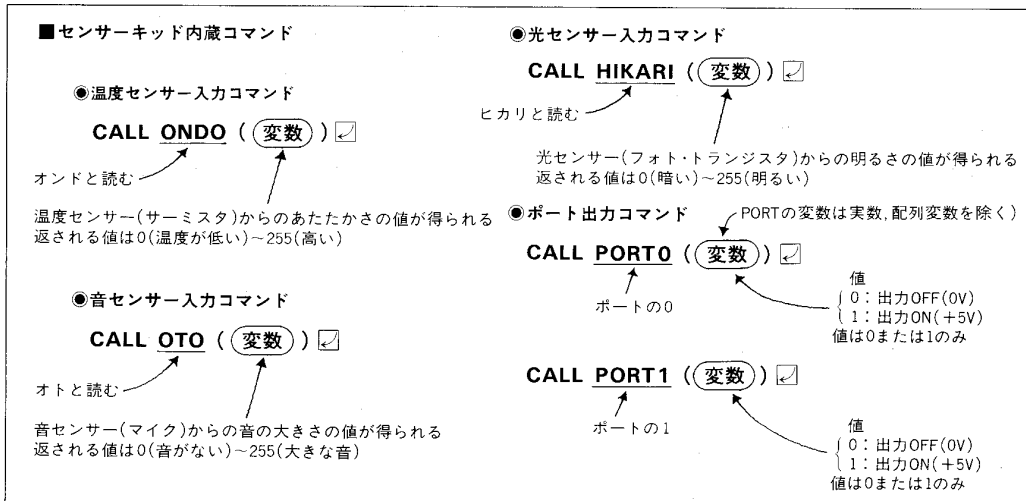
温度への変換は後で、紹介しています。

例：現在の暖かさを調べる

10 CALL ONDO (A)



《第6図》センサーの端子とコマンドの関係



《第7図》センサーキッドのコマンド

```
20 PRINT A
30 END
```

●音センサーの情報を得る
音の大きさを知ることができます。
CALL OTO (数値変数)
↑
音と読む
音センサー(マイク)からの音の大きさが得られます。
結果は、0~255までの値が得られ、音が小さいほど0に近付き、大きくなるに従って255に近付きます。
例：部屋の騒音を調べる

```
10 CALL OTO (A)
20 PRINT A
30 END
```

●光センサーの情報を得る
明るさの状態がわかるコマンドです。
CALL HIKARI (数値変数)
↑ ↑
コール 光と読む
光センサー(フォト・トランジスタ)からの明るさ情報が()内の"数値変数"に入ります。
結果は0~255までの値が得られます。暗いときは値が0に近付き、明るくなるに従って255に近付きます。
例：現在の部屋の明るさを調べる。

```
10 CALL HIKARI (A)
20 PRINT A
30 END
```

●外部出力のコントロール
外部出力端子に接続してある機器をコントロールできます。
出力は二つあり、1をPORT (ポート) 0, もう一つをPORT (ポート) 1

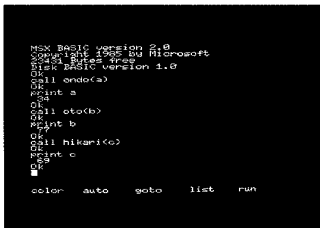
としています。
CALL PORT0 (数値変数か値)
↑ ↑
ポート0の場合 0または1
CALL PORT1 (数値変数か値)
↑ ↑
ポート1の場合 0または1
出力端子の状態は、()内の値が"0"のとき、L (ロウ) 0Vとなり、"1"のときH (ハイ) +5Vとなります。
注意として、()内の値は、0または1でなければなりません。数値変数の場合も同様で内容が0か1以外ではエラーがでます(詳しい使いかたは後で説明しています)。
例：PORT0を5V(ハイレベル)に、PORT1を0V(ロウレベル)にする

```
10 CALL PORT0 (1)
20 CALL PORT1 (0)
30 END
```

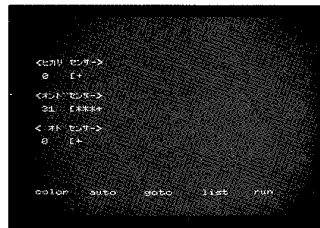
※数値変数は1文字です。

プログラムしてみよう!

さて、コマンドがわかったところで実際にプログラムしてみましょう。
【例題】センサーの状態を表示する
センサーのコマンドを実行すると、状態が変数に入りますから、これをPRINT文で表示すればOKですね(写真9)。
プログラムを実行させ、センサーにいろいろな状況を与えてみましょう。表示される数値が変わるはずですが、プログラムは、光センサー用ですが、CALL HIKARIの部分を変更すると温度センサーの状態、CALL OTOにすると音センサーの状態がわかります。
【例題<リスト1>】センサーの状態をグラフ表示させる。



《写真9》PRINT文で変数を表示



《写真10》簡単なグラフを描かす

《リスト1》

```
1 ' ex1
10 CALL HIKARI (A) ← 光センサーの状態 (CALL ONDO(A) ← 温度センサ
20 PRINT A ← CALL OTO(A) ← 音センサーの内容
30 GOTO 10 ( )内の変数に状態が入る(0~255の数値)
```

《リスト2》

```

1 ' ex2
10 CLS
20 LOCATE 0,3 :PRINT"<ヒカリ センサー>"
30 LOCATE 0,8 :PRINT"<オト センサー>"
40 LOCATE 0,13:PRINT"<オト センサー>"
50 '----- センサー ニュウリョク -----
60 CALL HIKARI(A)
70 CALL ONDO(B)
80 CALL OTO(C)
90 '----- ヒカリ センサー ノ グラフ -----
100 LOCATE 0,5 :AA=A/8
110 IF AA>AM THEN AM=AA
120 PRINT A;TAB(5);"[";STRING$(AA,"*");SPC(40-AA)
130 LOCATE AM+6,5:PRINT"+
140 '----- オト センサー ノ グラフ -----
150 LOCATE 0,10:BB=B/8
160 IF BB>BM THEN BM=BB
170 PRINT B;TAB(5);"[";STRING$(BB,"*");SPC(40-BB)
180 LOCATE BM+6,10:PRINT"+
190 '----- オト センサー ノ グラフ -----
200 LOCATE 0,15:CC=C/8
210 IF CC>CM THEN CM=CC
220 PRINT C;TAB(5);"[";STRING$(CC,"*");SPC(40-CC)
230 LOCATE CM+6,15:PRINT"+
240 GOTO 60
    
```

《リスト3》

```

1 ' oto1
10 SOUND 7,252:SOUND 8,15
20 '
30 CALL OTO(A)
40 SOUND 0,A
50 GOTO 30
    
```

センサーの結果は数値ですから、アナログ感覚ではわかりにくいですね。そこで、結果をバーグラフで表示してみましょう(リスト2)。

プログラムは、STRING\$命令を使って、センサーの結果の値だけ"*"を表示させています(写真10)。

CDにチャレンジ!

君達もよく知っている"CD"はなかなかいい音をしていますね。

そのCDは、音をデジタル化して盤に記録し、再びアナログに戻して再生できるものです。

このデジタル/アナログをセンサーキッドで行ってみましょう。

●入ってきた音の情報でサウンドを鳴らす。

音の入力は、音センサーから行います。CALL OTOで音の強弱によりその結果が変わります。つまり、音をデジタル化できるわけです。

CDなどでは、音の時間間隔をととも短く区切ってそのときの音の高さをデジタル化しています。つまり、センサーキッドでも時間間隔を短く区切って、

音をどってくれます。

そのときの音の高さを記録/再生すれば、CDの実験ができそうです。

まずは、入ってきた音の状態でパソコン本体からサウンドを出してみましょう。

音センサー入力にCDやラジカセのヘッドホン出力をつなぎ、実行してみましよう。

命令は、音源を直接コントロールする"SOUND"命令を使います(リスト3)。

このプログラムは、音の高さを音センサーの結果で変えています。

このプログラムでは、音程が音の強弱で変わるので、入力音が再現されません。

そこで、音を出しておき、ノイズの高さと音量を入力される音センサーの情報で変えてみます。

パソコンがリズムや音程も変えた例も載せて置きます(リスト4、5)。

《リスト4》

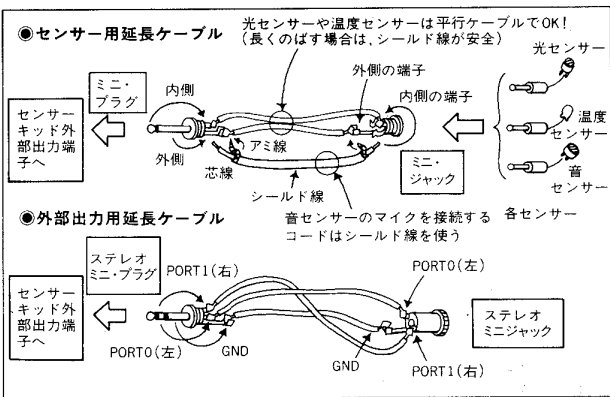
```

5 ' oto2
10 SOUND 7,246:SOUND 0,200
20 '
30 CALL OTO(A)
40 SOUND 6,A/8:SOUND 8,A/15
50 GOTO 30
    
```

■延長ケーブルを作ろう

センサーを利用したプログラムができて、センサーが本体から離せないのでは、お話になりません。そこで、センサーを離して使うためのハンダ付けに自信のある人で、パーツが入手できる人は延長ケーブルを作りましょう。

で結び、そのジャックにセンサーのプラグを差し込んで使うようにします。センサー側はモノラルのミニプラグ&ジャックですが、外部出力端子はステレオのミニプラグ&ジャックを使います。完成したらショートしていないことテストなどで確かめましょう。



《リスト5》

```

1 ' oto3
10 SOUND 7,246:SOUND 8,15
20
30 CALL OTO(A)
40 SOUND 0,A,SOUND 6,A/8
50 SOUND 8,A/10+5
60 GOTO 30
    
```

チャンネル設定
チャンネルAの音量
音センサーからのデータ
音センサーからのデータで変える
チャンネルAの周波数
チャンネルAの音量

通過人数カウンタ

学園祭などでは、何人の人が見にきてくれたのか気になります。“何人目”というのがわかれば、たとえば100人目の人に賞品をプレゼントするなんていうアトラクションが追加できますね。さて、プログラムですが“マザーセ

マザーセンサー

勉強しなさい！とうるさいお母さんが見えていないときは、つつい勉強しているふりをしてマンガなど読んでしまいますが、そんなとき突然お母さんが来てしまったら大変です。

そこで、光センサーを利用して、光センサーの前をお母さんが横切ったら、アラームが鳴る“マザーセンサー”を作ってみました(リスト6)。

プログラムは、光センサーの状態を常にチェックし、光が入らなくなったらアラームが鳴るようにしてあります。光が入っている間は、CALL HIKARIの値が0以上になっていますが、横切って光が来なくなったら、0に近づくはずで

センサーを取り付けた状態により、アラームのONを設定する感度を行番号10の変数STでセットできます。(光をささぎったときのアラーム時は、値が小さいとき感度が鈍い、大きいとき感度が高い。光が入ったときは逆)。

手を叩く間隔を利用する

音と音が入る間隔を数えて、その値を利用した何かを作ってみましょう。例えば、パン！パン！と手を叩く間隔の長さによって、おみくじの内容を変えたり、音でルーレットのスタート/ストップを行い、止めるまでの時間の長さで止めたあとの停止までの数を変えとか、いろいろと利用できそうです。

ポイントは、音が入ったかを常にチェックし(値が大きくなる)、入ったら今度は音が再び入るまで(値が小さいとき)数を数えます。

このとき、初めの音が入ったときにすぐ、次の音が入ったかをチェックすると、初めの音が長く続いている場合、次の音が入ったものと勘違いしてしまいます。そこで、初めの音が入ったら時間待ちして、音が切れるまで待ち、カウントを開始します(リスト7)。

《リスト6》

```

1 ' ex3
10 WIDTH 40:CLS:D=1:ST=0
20 CALL HIKARI(A)
30 PRINT"//// ママ - アラーム ////":PRINT
40 PRINT"ケイホウ ハツレイ";A
50 INPUT"アラームヲ ナラス フタイ ライレテカ サイ!(0-255)";ST
60 PRINT
70 PRINT"ヒカリカ キレタ アラーム...0"
80 PRINT" ハイッタ アラーム...1"
90 PRINT"トチラ デスカ?";K$:INPUT$(1)
100 IF K$=" " OR K$<>"1" THEN D=0
110 IF ST<0 OR ST>255 THEN BEEP:GOTO 50
120 LOCATE 10,10:PRINT"/ケンサ イノ フタイ/"
130 '----- メイン ルーチン -----
140 CALL HIKARI(A)
150 LOCATE 16,12:PRINT A
160 IF D=1 AND A<ST THEN 140
170 IF D=0 AND A>ST THEN 140
180 '----- ケイホウ ハツレイ -----
190 BEEP:CALL PORTO(1)
200 PRINT"//// ママ - ケイホウ ハツレイ! ////":PRINT
210 PRINT" * * * * *"
220 PRINT" * * * * *"
230 PRINT" * * * * *"
240 PRINT" * * * * *"
250 PRINT" * * * * *"
260 PRINT" * * * * *"
270 PRINT" * * * * *"
280 PRINT:PRINT
290 PRINT" * * * * *"
300 PRINT" * * * * *"
310 PRINT" * * * * *"
320 PRINT" * * * * *"
330 PRINT" * * * * *"
340 PRINT" * * * * *"
350 PRINT" * * * * *"
360 PRINT" * * * * *"
370 PRINT:PRINT:PRINT
380 '----- ケイホウ サウンド -----
390 PLAY"T200V15CE","T200V15EG"
400 IF INKEY$=" " THEN IF PLAY(1) THEN 400 ELSE 390
410 CALL PORTO(0)
420 CLS:END
430 '-----
440 ' ママ - センサ - PROG.
450 ' Copyright 1988/6
460 ' by S.TANJI
470 '-----
    
```

アラームを鳴らす光センサーの値(センサーの状態により変わる
ので、自分で変えてください)

設定値になっていないとき、再度入力

設定値になったらアラームを出す

アラームを止めたとき、外部出力も止める

《リスト7》

```

5 ' EX4
10 CLS
20 '----- オトカ ハイラムマ マツ -----
30 C=1:K=5
40 PRINT"RADY > ";
50 CALL OTO(A)
60 IF A<K THEN 50
70 PRINT"オトカ ハイラムマ!"
80 '----- シェカンマチ -----
90 FOR T=1 TO 100:NEXT
100 PRINT:PRINT"フタヒ オトカ ハイラム カウントヲ シュウリョウ シマス!"
110 '--- オトカ フタヒ ハイラムマ カウント ---
120 CALL OTO(A)
130 IF C>1000! THEN 150
140 IF A<K THEN C=C+1:GOTO 120
150 '----- ケッカ ヒョウシ -----
160 PRINT"カウント スウ:";C
170 FOR T=1 TO 300:NEXT
180 PRINT
190 GOTO 30
    
```

センサーの感度(小さいほど感度がよい)

音センサーからの入力

設定値(感度)以下なら再入力

ポイントの時間待ち

音センサーからの入力

センサーから入力が再びあるまでループし、カウントする

時間待ちを入れないと2回目の音が入ったとき、すぐに音センサーから(行50,60)の入力が新たにあったと思ってしまう

《リスト8》

```

1  'ENS
10 WIDTH 40:CLS:Y=8:N=0:S=10
20 PRINT TAB(6)"/" "フコニス"ウ カウンタ 1"
30 LOCATE 3, 6:PRINT STRING$(31,"-")
40 LOCATE 3,16:PRINT STRING$(31,"-")
50 LOCATE 9,3:PRINT"227"
60 LOCATE 31,15:PRINT"2"
70 GOSUB 1000
80
90 [CALL HIKARI(A)]
100 LOCATE 25,3:PRINT A:"
110 IF A>S THEN 90
120 LOCATE 15,18:PRINT"フコニス"
130 [CALL HIKARI(A)]
140 LOCATE 25,3:PRINT A:"
150 IF A<=S THEN 130
160 LOCATE 15,18:PRINT"
170 [REPEAT]GOSUB 1000
180 GOTO 90
1000 "----- 数字表示ルーチンへ
1010 K=1
1020 N$=RIGHT$( "000"+MID$(STR$(N),2),4)
1030 FOR X=7 TO 7+6:STEP 6
1040 D$=VAL(MID$(N$,K,1))
1050 GOSUB 1080:K=K+1
1060 NEXT
1070 RETURN
1080 "----- NUM.DSP
1090 ON D$+1 GOTO 1100,1190,1280,1370,1460,1550,1640,1730,1820,1910
1100 '
1110 LOCATE X,Y :PRINT" *** "
1120 LOCATE X,Y+1:PRINT" * "
1130 LOCATE X,Y+2:PRINT" * "
1140 LOCATE X,Y+3:PRINT" * "
1150 LOCATE X,Y+4:PRINT" * "
1160 LOCATE X,Y+5:PRINT" * "
1170 LOCATE X,Y+6:PRINT" *** "
1180 RETURN
1190 '
1200 LOCATE X,Y :PRINT" * "
1210 LOCATE X,Y+1:PRINT" * "
1220 LOCATE X,Y+2:PRINT" * "
1230 LOCATE X,Y+3:PRINT" * "
1240 LOCATE X,Y+4:PRINT" * "
1250 LOCATE X,Y+5:PRINT" * "
1260 LOCATE X,Y+6:PRINT" *** "
1270 RETURN
1280 '
1290 LOCATE X,Y :PRINT" *** "
1300 LOCATE X,Y+1:PRINT" * "
1310 LOCATE X,Y+2:PRINT" * "
1320 LOCATE X,Y+3:PRINT" * "
1330 LOCATE X,Y+4:PRINT" * "
1340 LOCATE X,Y+5:PRINT" * "
1350 LOCATE X,Y+6:PRINT"*****"
1360 RETURN
1370 '
1380 LOCATE X,Y :PRINT" *** "
1390 LOCATE X,Y+1:PRINT" * "
1400 LOCATE X,Y+2:PRINT" * "
1410 LOCATE X,Y+3:PRINT" * "
1420 LOCATE X,Y+4:PRINT" * "
1430 LOCATE X,Y+5:PRINT" * "
1440 LOCATE X,Y+6:PRINT" *** "
1450 RETURN
1460 '
1470 LOCATE X,Y :PRINT" * "
1480 LOCATE X,Y+1:PRINT" * "
1490 LOCATE X,Y+2:PRINT" * "
1500 LOCATE X,Y+3:PRINT" * "
1510 LOCATE X,Y+4:PRINT"*****"
1520 LOCATE X,Y+5:PRINT" * "
1530 LOCATE X,Y+6:PRINT" * "
1540 RETURN
1550 '
1560 LOCATE X,Y :PRINT"*****"
1570 LOCATE X,Y+1:PRINT" * "
1580 LOCATE X,Y+2:PRINT" * "
1590 LOCATE X,Y+3:PRINT"*****"
1600 LOCATE X,Y+4:PRINT" * "
1610 LOCATE X,Y+5:PRINT" * "
1620 LOCATE X,Y+6:PRINT" *** "
1630 RETURN
1640 '
1650 LOCATE X,Y :PRINT" *** "
1660 LOCATE X,Y+1:PRINT" * "
1670 LOCATE X,Y+2:PRINT" * "
1680 LOCATE X,Y+3:PRINT"*****"
1690 LOCATE X,Y+4:PRINT" * "
1700 LOCATE X,Y+5:PRINT" * "
1710 LOCATE X,Y+6:PRINT" *** "
1720 RETURN
1730 '
1740 LOCATE X,Y :PRINT"*****"
1750 LOCATE X,Y+1:PRINT" * "
1760 LOCATE X,Y+2:PRINT" * "
1770 LOCATE X,Y+3:PRINT" * "
1780 LOCATE X,Y+4:PRINT" * "
1790 LOCATE X,Y+5:PRINT" * "
1800 LOCATE X,Y+6:PRINT" * "
1810 RETURN
1820 '
1830 LOCATE X,Y :PRINT" *** "
1840 LOCATE X,Y+1:PRINT" * "
1850 LOCATE X,Y+2:PRINT" * "
1860 LOCATE X,Y+3:PRINT"*****"
1870 LOCATE X,Y+4:PRINT" * "
1880 LOCATE X,Y+5:PRINT" * "
1890 LOCATE X,Y+6:PRINT" *** "
1900 RETURN
1910 '
1920 LOCATE X,Y :PRINT" *** "
1930 LOCATE X,Y+1:PRINT" * "
1940 LOCATE X,Y+2:PRINT" * "
1950 LOCATE X,Y+3:PRINT"*****"
1960 LOCATE X,Y+4:PRINT" * "
1970 LOCATE X,Y+5:PRINT" * "
1980 LOCATE X,Y+6:PRINT" * "
1990 RETURN
2000 "-----
2010 " フコニス"ウ カウンタ
2020 " Copyright 1988/6
2030 " by S.TANJI
2040 "-----

```

ンサー”の応用になります。ポイントは、なんとと言っても通過中を判断させることです。

つまり、光を出しておき、通過したときセンサーに光が来なくなったのをチェックして数えるわけですが、ただそのまま数を数えたのでは、センサーに光が来なくなっている間だけ、数えてしまいます。

そこで、光が来なくなったとき（値が0に近づくとき）動作を開始し、光が再び入るまで（値が255に近づくまで）待ちます（通過中）。もし、光が入ってくれば通り過ぎたということですから、数を数えてもよいわけです。

プログラムでは、人数を大きな文字で表示させてみました。スプライトを利用すればもっと高速できれいな数字が出せるでしょう（リスト8）。

また、ある人数になったらファンファーレが鳴るなんていうのもおもしろいかも知れません。

現在の温度は？

温度センサーは、確かに現在の温度を測定しているわけですが、結果が0~255の数字でしか得られないので、何度℃の表示ができません。

そこで、値から温度の変換をしてみましょう（リスト9）。

●氷の温度やお湯の温度で測定！

実測データを載せておきますが、センサーキッドの部品やセンサーのばらつきにより、温度と値の関係が一定ではありません。

そこで、君達自身で“値”対“温度”の関係測定してください。

まず、温度センサーであるサーミスタの端子とコード部分に接着剤（ゴム系）を塗り、完全防水させます。そして、温度計の先の部分にサーミスタを取り付け、温度と値の関係を読み取ります。

CALL ONDOの値が落ち着かないので10回~100回測定して、その平均を求めるとよいでしょう。

まず、氷水の中にセンサーを入れ、そのときの温度と値を読み取ります。そのままにしておけば、常温までの関係が読み取れるでしょう。

そのあと、温めたお湯の温度を変えて、そのときの値を読み取りましょう。

結果をまとめ、グラフを描けば、値から温度を求められるでしょう（第8図）。

《リスト9》

```

1 'EX13
10 CLS:K=100
20
30 T=0 ← 合計をゼロに ← チェックする回数
40 FOR Z=1 TO K
50 CALL ONDO(A) ← 温度センサーからの入力
60 T=T+A ← 温度センサー値の合計
70 NEXT
80 O=INT(T/K) ← チェック回数で合計を割り、平均を求める
90 LOCATE 4,10:PRINT K;"カイ ソクテイノ ヘミンチ =" ;O;" "
100 GOTO 30
    
```

現在の温度を表示してみよう

値と温度の関係がわかったら、その値から温度を表示させてみましょう。

例として、計算で求めるのではなく、値対温度のデータをDATA文に入れておき、センサーからの値でそのデータを表示してみました《リスト10》。

プログラムには、約50°まで値がゼロから1ずつの温度をDATA文に入れておきます。

これを配列変数に書き込み、配列変数を読みだす値にセンサーからの値をそのまま利用しています。つまり、配列変数(値)とすれば、そのときの温度が得られるわけです。

値が1ずつの温度は、温度と値からの関係で計算によって求めます。

大きな数字で温度を表示してみる

通過人数カウンタの大きな数字の表

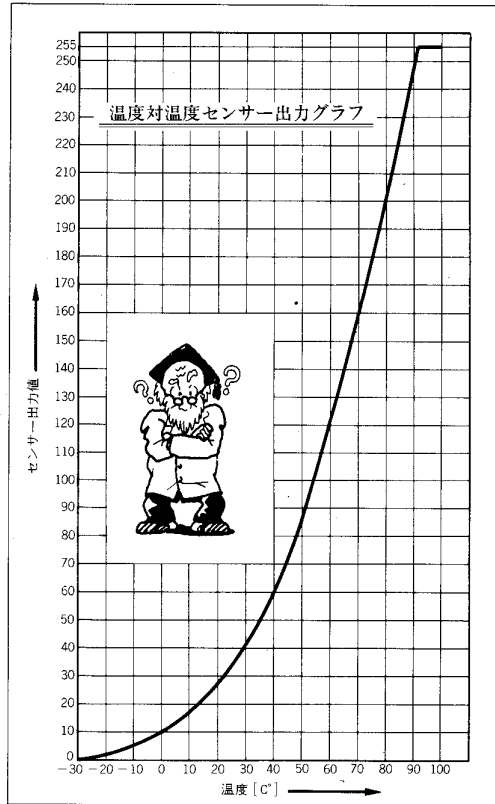
示を利用して、測定した温度を大きな数字で表示してみましょう。(写真11)

表示を工夫すると同じ温度測定でもおもしろくなりますね。《リスト11》

湯かげんセンサー

温度がわかるようになったので、今度はその値を利用してみましょう。

お湯にサーミスタを入れ、そのときの温度を測定します。そして、設定した温度にな



《第8図》センサーと温度の比較グラフ例

《リスト10》

```

1 'EX7
10 DIM E(90)
20 FOR T=0 TO 89:READ E(T):PRINT E(T):NEXT ← 温度データを読み配列変数に入れる
30 CLS
40 LOCATE 7,5:PRINT"//// ケンガイノ オント" ////"
50
60 FOR T=1 TO 100
70 CALL ONDO(A) ← 温度センサーからの値を100回読み
80 ZZ=ZZ+A ← 平均を求める
90 NEXT
100 ZZ=ZZ/100
110 IF ZZ>89 THEN 60 ← センサーの値が89以上の温度はデータ文に入れていないので戻る
120
130 LOCATE 13,10:PRINT E(ZZ);" "
140 LOCATE 19,10:PRINT "[C°]" ← これだけでセンサーの値(変数ZZ)から温度が得られる
150 GOTO 60
300 '----- オント データ -----
310 DATA -25,-22.5,-20,-15,-12.5,-10,-7.5,-5,-2.5,0
320 DATA 1,2.5,4,5,6,7.5,9,10,11,12
330 DATA 13,14,15,16,17,18,19,20,20.5,21
340 DATA 22,23,24,24.5,25,26,27,27.5,28,29
350 DATA 29.5,30,30.5,31,32,32.5,33,33.5,34,34.5
360 DATA 35,36,36.5,37,37.5,38,38.5,39,39.3,39.7
370 DATA 40,40.5,41,41.3,41.7,42,42.5,43,43.5,44
380 DATA 44.3,44.7,45,45.5,46,46.3,46.7,47,47.3,47.7
390 DATA 48,48.5,49,49.3,49.5,50,50.3,50.7,51,51.25
    
```

センサーの値が0から1ずつ変わったときの温度 (値が89のときまで入っている)

センサーの値が89のときの温度

(リスト11)

```

1 'EXR
10 DIM E(90)
20 FOR I=1 TO 90:READ X(I):PRINT E(I):NEXT I
30 CLS:Y=7
40 LOCATE 7,3:PRINT"/ / / / / * * * * / / / / /"
50 LOCATE 30,16:PRINT("C")
60
70 FOR I=1 TO 100
80 CALL OND(A)
90 Y=Y+1
100 NEXT I
110 GOTO 100
120
130 H=22:GOSUB 1000
140 GOTO 10
150
160 DATA -25,-22.5,-20,-15,-12.5,-10,-7.5,-5,-2.5,0
170 DATA 1,2.5,4.5,6.5,7.5,9,10,11,12
180 DATA 13,14,15,16,17,18,19,20,20.5,21
190 DATA 22,23,24,24.5,25,25.5,27,27.5,28,29
200 DATA 29.5,30,30.5,31,32,32.5,33,33.5,34,34.5
210 DATA 35,35.5,36,36.5,37,38,38.5,39,39.5,40
220 DATA 40,40.5,41,41.5,41.7,42,42.5,43,43.5,44
230 DATA 44.2,44.7,45,45.5,46,46.5,47,47.5,48,48.7
240 DATA 48.8,49,49.3,49.5,50,50.5,50.7,51,51.25
250 H=25
260 GOSUB 1000
270 END
1000
----- DISPLAY
1010 K=1
1020 M=STR$(M)
1030 FOR X=2 TO 2+K+1 STEP 6
1040 D=MID$(M,X,1)
1050 IF D="." THEN D="0"
1060 IF D="0" THEN D="1"
1070 D=VAL(D)
1080 IF D=0 AND D<="0" THEN D=-1
1090 GOSUB 1100
1100 K=K+1:NEXT X
1110 RETURN
1120
----- NUM.DSP
1130 ON D+1 GOTO 1220,1210,1400,1490,1580,1670,1760,1850,1940,2030,2120,2210
1140 LOCATE X,Y:PRINT " "
1150 LOCATE X,Y+1:PRINT " "
1160 LOCATE X,Y+2:PRINT " "
1170 LOCATE X,Y+3:PRINT " "
1180 LOCATE X,Y+4:PRINT " "
1190 LOCATE X,Y+5:PRINT " "
1200 LOCATE X,Y+6:PRINT " "
1210 RETURN
1220
1230 LOCATE X,Y:PRINT "###"
1240 LOCATE X,Y+1:PRINT " "
1250 LOCATE X,Y+2:PRINT " "
1260 LOCATE X,Y+3:PRINT " "
1270 LOCATE X,Y+4:PRINT " "
1280 LOCATE X,Y+5:PRINT " "
1290 LOCATE X,Y+6:PRINT "###"
1300 RETURN
1310
1320 LOCATE X,Y:PRINT " "
1330 LOCATE X,Y+1:PRINT " "
1340 LOCATE X,Y+2:PRINT " "
1350 LOCATE X,Y+3:PRINT " "
1360 LOCATE X,Y+4:PRINT " "
1370 LOCATE X,Y+5:PRINT " "
1380 LOCATE X,Y+6:PRINT " "
1390 RETURN
1400
1410 LOCATE X,Y:PRINT "###"
1420 LOCATE X,Y+1:PRINT " "
1430 LOCATE X,Y+2:PRINT " "
1440 LOCATE X,Y+3:PRINT " "
1450 LOCATE X,Y+4:PRINT " "
1460 LOCATE X,Y+5:PRINT " "
1470 LOCATE X,Y+6:PRINT "###"
1480 RETURN
1490
1500 LOCATE X,Y:PRINT "###"
1510 LOCATE X,Y+1:PRINT " "
1520 LOCATE X,Y+2:PRINT " "
1530 LOCATE X,Y+3:PRINT " "
1540 LOCATE X,Y+4:PRINT " "
1550 LOCATE X,Y+5:PRINT " "
1560 LOCATE X,Y+6:PRINT "###"
1570 RETURN
1580
1590 LOCATE X,Y:PRINT " "
1600 LOCATE X,Y+1:PRINT " "
1610 LOCATE X,Y+2:PRINT " "
1620 LOCATE X,Y+3:PRINT " "
1630 LOCATE X,Y+4:PRINT "###"
1640 LOCATE X,Y+5:PRINT " "
1650 LOCATE X,Y+6:PRINT " "
1660 RETURN
1670
1680 LOCATE X,Y:PRINT "###"
1690 LOCATE X,Y+1:PRINT " "
1700 LOCATE X,Y+2:PRINT " "
1710 LOCATE X,Y+3:PRINT "###"
1720 LOCATE X,Y+4:PRINT "###"
1730 LOCATE X,Y+5:PRINT " "
1740 LOCATE X,Y+6:PRINT "###"
1750 RETURN
1760
1770 LOCATE X,Y:PRINT "###"
1780 LOCATE X,Y+1:PRINT " "
1790 LOCATE X,Y+2:PRINT " "
1800 LOCATE X,Y+3:PRINT "###"
1810 LOCATE X,Y+4:PRINT " "
1820 LOCATE X,Y+5:PRINT " "
1830 LOCATE X,Y+6:PRINT "###"
1840 RETURN
1850
1860 LOCATE X,Y:PRINT "###"
1870 LOCATE X,Y+1:PRINT " "
1880 LOCATE X,Y+2:PRINT " "
1890 LOCATE X,Y+3:PRINT " "
1900 LOCATE X,Y+4:PRINT " "
1910 LOCATE X,Y+5:PRINT " "
1920 LOCATE X,Y+6:PRINT " "
1930 RETURN
1940
1950 LOCATE X,Y:PRINT "###"
1960 LOCATE X,Y+1:PRINT " "
1970 LOCATE X,Y+2:PRINT " "
1980 LOCATE X,Y+3:PRINT "###"
1990 LOCATE X,Y+4:PRINT " "
2000 LOCATE X,Y+5:PRINT " "
2010 LOCATE X,Y+6:PRINT "###"
2020 RETURN
2030
2040 LOCATE X,Y:PRINT "###"
2050 LOCATE X,Y+2:PRINT " "
2060 LOCATE X,Y+3:PRINT " "
2070 LOCATE X,Y+4:PRINT " "
2080 LOCATE X,Y+5:PRINT " "
2090 LOCATE X,Y+6:PRINT " "
2100 LOCATE X,Y+6:PRINT " "
2110 RETURN
2120
2130 LOCATE X,Y:PRINT " "
2140 LOCATE X,Y+1:PRINT " "
2150 LOCATE X,Y+2:PRINT " "
2160 LOCATE X,Y+3:PRINT " "
2170 LOCATE X,Y+4:PRINT " "
2180 LOCATE X,Y+5:PRINT " "
2190 LOCATE X,Y+6:PRINT " "
2200 RETURN
2210
2220 LOCATE X,Y:PRINT " "
2230 LOCATE X,Y+1:PRINT " "
2240 LOCATE X,Y+2:PRINT " "
2250 LOCATE X,Y+3:PRINT "###"
2260 LOCATE X,Y+4:PRINT " "
2270 LOCATE X,Y+5:PRINT " "
2280 LOCATE X,Y+6:PRINT " "
2290 RETURN
2300
2310
2320
2330
2340

```



《写真11》大きな文字で現在の温度を表示する

ったらアラームが鳴るプログラムを作りましょう(リスト12)。

プログラムは、温度表示プログラムが原型となっています。ただ、異なる点は、設定温度を入力させて、得られた温度と設定温度を比較して、設定温度以上ならアラームが鳴るようになっています。感温に注意！を読むこと。

センサーで外部機器をコントロール

センサーキッドは、ただセンサーの情報が得られるだけではありません。外部機器をコントロールするための“外部出力”も備えています。

つまり、単独で外部機器のコントロールもできますが、センサーの情報を元に外部機器を動かすことができるのです。

たとえば、光センサーで現在の明るさを測定して、もし暗くなってきたら、外部出力により電灯をONにすると、温度センサーで暑くなってきたことがわかったら、扇風機を回すというように、センサーとの連携プレーが楽しめるのです。

センサーキッドの楽しみは、本来ここにあるのです。

外部出力につないでみよう

外部出力端子は、他のセンサーの端子と異なり、ステレオ・ミニジャックになっています。(ウォークマンのヘッドホン端子でおなじみですね)。

この出力はステレオの接続で言うと左側(端子の先)がPORT0(ポート)で右側(内側)がPORT1(ポート)です。そしてGND(グラウンド:アース)は共通です。

そして、コマンド“CALL PORT0/PORT1”命令で端子の電圧を変えることができます。

つまり、電圧の変化で動作するようなものを接続しておけばよいのです。出力電圧は、OFFのとき“0V”(L

《リスト12》

```

1 'EX9
10 DIM E(90)
20 FOR T=0 TO 89:READ E(T):PRINT E(T):NEXT
30 CLS
40 LOCATE 6,17
50 INPUT"オントニ セット シマスカ? [C*]";S
60 LOCATE 4,5:PRINT"//// ケンサ イノ      オント  ////"
70 "----- オント ソクテイ -----
80 FOR T=1 TO 50
90 CALL ONDO(A) ← 温度測定
100 ZZ=ZZ+A
110 NEXT
120 LOCATE 8,13:PRINT SPC(30)
130 ZZ=ZZ/50
140 IF ZZ>89 THEN 80
150 "----- オント ヒョウケ -----
160 LOCATE 13,10:PRINT E(ZZ);" "
170 LOCATE 19,10:PRINT "[C*]"
180 IF E(ZZ)<S THEN 80 ← 温度が設定値以下のときは戻る
190 LOCATE 8,13:PRINT"--- イヨカケンサヨ!! ---"
200 PLAY"L3205V15G" ← 設定値以上になったらアラーム
210 GOTO 80 (あまりにも熱いときは、注意表示を出したほうがいい)
300 "----- オント データ -----
310 DATA -25,-22.5,-20,-15,-12.5,-10,-7.5,-5,-2.5,0
320 DATA 1,2.5,4,5,6,7.5,9,10,11,12
330 DATA 13,14,15,16,17,18,19,20,20.5,21
340 DATA 22,23,24,24.5,25,26,27,27.5,28,29
350 DATA 29.5,30,30.5,31,32,32.5,33,33.5,34,34.5
360 DATA 35,36,36.5,37,37.5,38,38.5,39,39.3,39.7
370 DATA 40,40.5,41,41.3,41.7,42,42.5,43,43.5,44
380 DATA 44.3,44.7,45,45.5,46,46.3,46.7,47,47.3,47.7
390 DATA 48,48.5,49,49.3,49.5,50,50.3,50.7,51,51.25
400 "=====
410 '
420 ' Copyright 1988/6
430 ' by S.TANJI
440 "=====

```

レベル) でONにしたとき "+5V"(Hレベル) になります。
 コマンドは、前にも説明しましたが、CALL PORT 0 (0) で OFF (0V)、
 CALL PORT 0(1) で ON (+5V) となります。

【注意】 出力端子は絶対にショートさせないでください。PORT0とPORT1同士や各PORTとGND間をショートさせると、保護回路が入っていないので内部のICが壊れてしまいます。どんなことがあっても絶対にショートさせないようにしてください。

また、出力電流は最大10mA (ミリアンペア) までです。よって、電流を必要とする豆電球やモーターなどは直接つなげません (小電流の "赤色" の

LED (発光ダイオード) 1個を抵抗を介して接続できる程度)。

流れ過ぎた場合も同様にICの破壊につながります。注意しましょう!

- LEDを光らせる
出力がどうなっているかを見るため

に、LED (発光ダイオード) を光らせてみましょう (写真13)。

使用するLEDは小電流でも明るく光る赤色のものを使います (緑色などは電流を喰ってしまう)。

LEDの端子電圧を2Vとし、5Vが加わったときに3mA流すとすると、抵抗1KΩ (キロオーム) を接続することになります (第10図)。

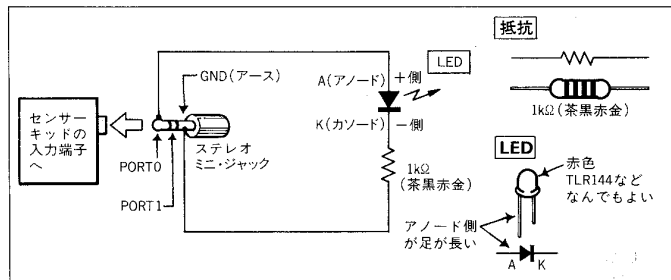
LEDには極性がありますので注意して接続します。+側 (アノード: 普通足の長いほう) をPORT 0につなぎ、-側 (カソード) を抵抗を介してGNDにつなぎます。

```

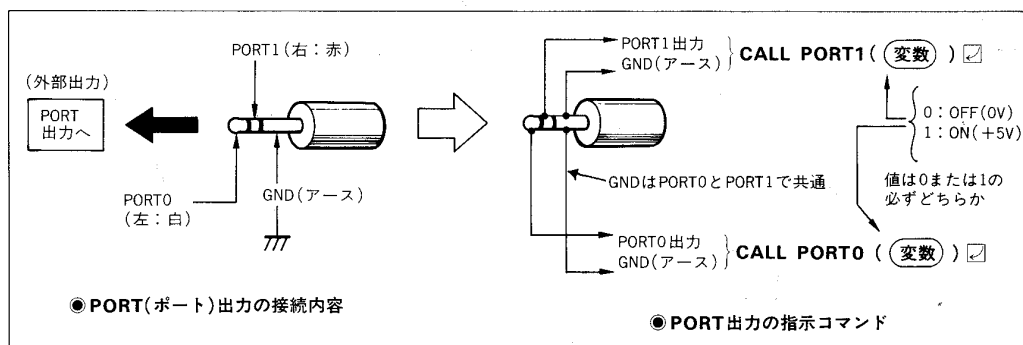
CALL PORT 0 (1)
でLEDが光り、
CALL PORT 0 (0)
で消えるはず。

```

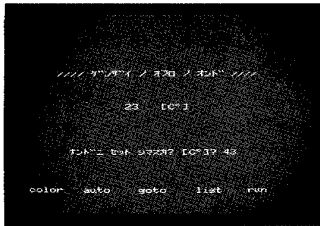
★端子の内容を覚えている!
 実行してわかることですが、プログラムを実行後も端子の電圧は、実行させたときのままで保持されています。これを "ラッチ" といいます。
 センサーキッドでは、BASICに入ったとき、端子の状態はいずれも "OFF (0V)" ですが、プログラムでONにすると、プログラムが終了しても、BASICを終了したり、電源を切らない限り、



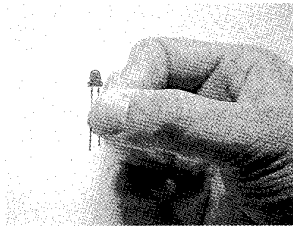
《第10図》LED点灯回路



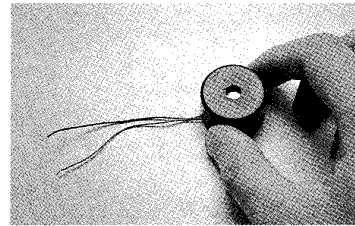
《第9図》外部出力端子の接続と指示コマンド



《写真12》温度を表示



《写真13》赤いLEDのほうが消費電流が少ない



《写真14》圧電ブザー

その状態を覚えています。
プログラムでのコントロールの際に覚えておきましょう。

●ブザーを鳴らす

同様に今度は、ブザーを鳴らしてみましよう(写真14)。

+5Vで鳴るブザー(圧電ブザーに駆動回路が入っているものがよい)をPORT 1につないでみましょう(第11図)。

すると、

```
CALL PORT1 (1)
    でブザーが鳴り、
CALL PORT1 (0)
    で鳴りやむはずです。
```

また、これをLEDと連動することもできます。プログラムを考えてみてください。

●LEDを点滅させる

さて次は、接続したLEDをチカ/チカ/と点滅させてみましょう。

ただ単に点滅させたのではおもしろくありませんから、キーボードの“スペースキー”が押されたら点滅させ、“RETURN(Enter)”キーが押されたら、ブザー音とともに点滅を終了させてみましょう。

点滅は、CALL PORT0(1)でONにし、CALL PORT0(0)でOFFさせると、早過ぎて点滅になりませんから時間待ちを入れて点滅するようにします。どうですか？自分のプログラムしだ

いで外部の端子を操れることがわかっていただけます。《リスト13 リスト14》

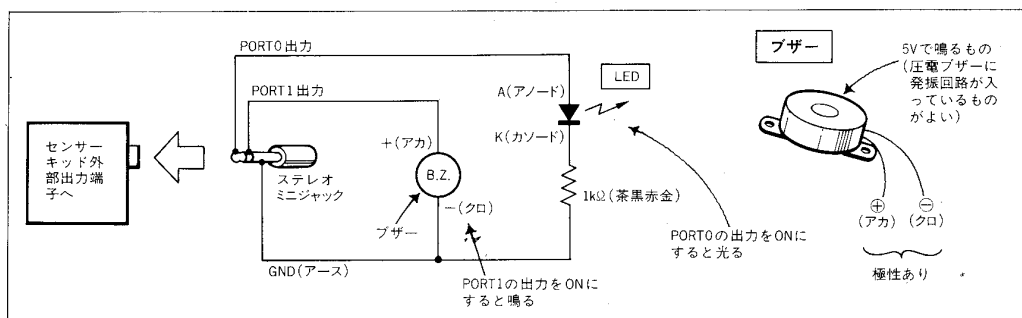
オリジナル・プログラムで利用して

《リスト13》

```
5 'ex12-1 この値を大きくするとゆっくり点滅し、小さくすると早くなる
10 ZZ=150
20 CALL PORT0(1) ← 外部出力0を ON
30 FOR T=1 TO ZZ:NEXT ← 時間待ちのループ
40 CALL PORT0(0) ← OFF
50 FOR T=1 TO ZZ:NEXT
60 GOTO 20
```

《リスト14》

```
5 'EX12-2
10 ZZ=6:T=0:F=-1
20 CALL PORT0(0):CALL PORT1(0) ← 出力をすべてOFFに
30 '----- メイン ルーチン -----
40 K$=INKEY$ ← リアルタイムキー入力(変数K$に内容が入る)
50 IF K$=" " THEN F=1 ← スペースキーが押されたとき
60 IF K$=CHR$(13) THEN F=0 ← Enterキーが押されたとき
70 IF F=1 THEN GOSUB 100
80 IF F=0 THEN GOSUB 170 } 押された内容の処理を行なう
90 GOTO 30
100 '----- LED テンメツ サフールーチン -----
110 IF T<0 THEN P=0
120 IF T>=0 THEN P=1
130 IF T>ZZ THEN T=-ZZ } 点滅させる内容をコントロール
140 CALL PORT0(P) ← 変数Pの値で
150 T=T+1 ← てんめつ 外部出力0をON/OFFする
160 RETURN ← カウント
170 '----- ブザー-ON & テンメツ シュウリョウ ----
180 CALL PORT1(1):CALL PORT0(1) ← 外部出力0と1をON(LEDが光り
190 FOR Z=1 TO 200:NEXT ← ブザーが鳴る)
200 CALL PORT1(0):CALL PORT0(0) ← LEDが消え、ブザーも止る
210 F=-1 ← キーが押されていない状態にする
220 RETURN
230 '=====
240 ' PORTチェック LED&ブザー PROG.
250 ' Copyright 1988/6
260 ' by Saichi-Tanji
270 '=====
```



《第11図》LED & ブザー出力回路

参考資料

センサーキットをもっと詳しく知りたい方のために、回路図などの資料を載せておきます(第12図)。

応用

センサーキットの各センサーの入力は、接続するセンサーが決められています。他のセンサーやオリジナルのセンサーなどを接続することができます。

温度センサーである“サーミスタ”は、温度により抵抗値が変化するもの

です。逆を言えば抵抗のあるものを接続することで、抵抗値によって値が変化させられるわけです。

たとえばバリオーム(可変抵抗)やニクロム線の抵抗となるものをつなげられるということです。ニクロム線で身長計のセンサーを作ったり、お風呂の水位をリアルタイムで観測できるかもしれません。CDSを接続して、明るさの測定もできるかもしれません。

また、光センサーである“フォト・トランジスタ”は明るさに応じて、コレクタ/エミッタ間の電流が変化し、結果的に電圧変化として利用しています。また、電圧の変化のあるものやサーミスタ同様に抵抗となるものを接続する

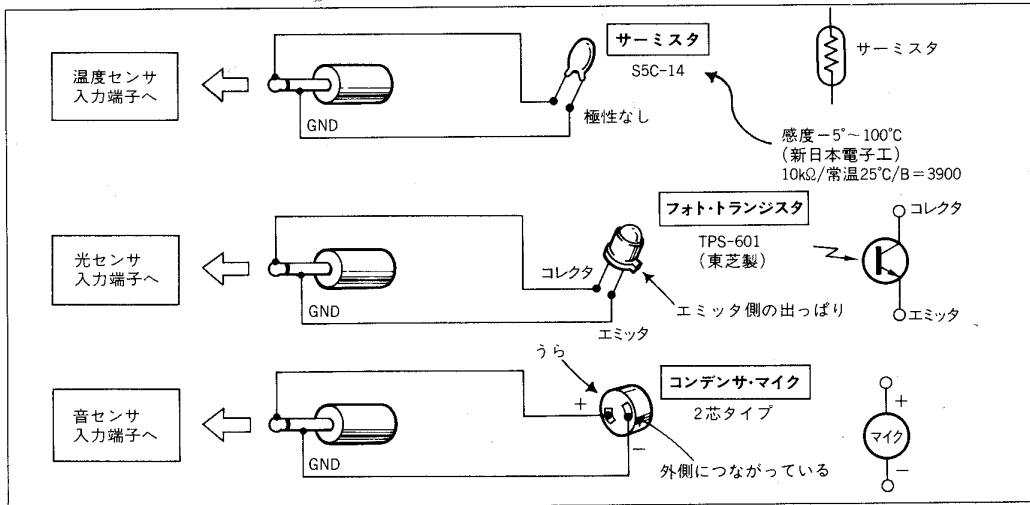
ことができます。

マイク入力には、アンプ回路が組み込まれています。マイク以外には、CDやラジカセなどの音源が接続できます。

補修・修理

センサーキットを利用して、もしセンサーが壊れてしまったら、同様な部品を購入して修理もできます。

このとき、音センサーである“マイク”や光センサーの“フォト・トランジスタ”には極性がありますので、注意してハンダ付けします(第13図)。



《第13図》使用センサーとその回路

感電に注意しよう!

センサーキットをお風呂場など水に濡れた場所で使うと、感電する危険性があります。

パソコン本体をお風呂場などに持ちこんだり、センサーをむきだしのまま水につこんだりするようなことは、絶対にやめてください!!

もちろん、口にふくんだりすることも危ないので、してはいけません。

電気の危険性についてよく知らない人は、通常の使い方だけで実験してください。また、センサーのコードをのぼしたりする場合など、特殊なことにチャレンジするときは、適切な指導をしていただける方に相談しましょう。

なお、本機の使用法を誤って発生した事故や故障については、その責を負いかねますので御了承下さい。

電波新聞社

※本機に関するお問い合わせは電波新聞社出版部03-445-6111にお願いします。
※近く本機の応用例を載せた本が出版される予定です。お楽しみに。