

第6章



ウェブ・ブラウザでLED出力や
スイッチ入力を扱う

H8/Tiny BASIC による ネットワーク I/O 制御

三岩 幸夫
Yukio Mitsuiwa

ローカル I/O の実例

ここでは、AKI-H8/3069F フラッシュマイコン LAN ボードに図1のようにポート A に LED を三つ、スイッチを一つ外付けた回路を前提にします。ポート A の割り当ては表1のとおりです。

■ プログラミング

まず、三つの LED を一つずつシフトしながら繰り返し点灯するプログラムを制作してみます。

各 LED はそれぞれポート A の 04h (PA2), 10h (PA4), 40h (PA6) に割り当ててあります。ですから、三つの LED を一つずつシフトしながら点灯するには、ポート A に出力するデータを 04h → 10h → 20h → 04h

→ 10h → 20h → … のように無限に繰り返します。

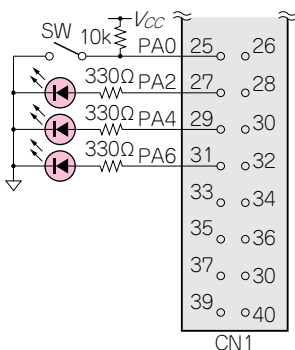
H8/Tiny BASIC で、このような繰り返し処理をするには以下の二つのやりかたがあります。

- **for** 文による繰り返しと **goto** 文による無限ループ
この処理をフローチャートで表すと図2のようになります。これに基づいてコーディングした BASIC プログラムがリスト1です。
- **if** 文によるデータ初期化と **goto** 文による無限ループ
この処理をフローチャートで表すと図3のようになります。これに基づいてコーディングした BASIC プログラムがリスト2です。

■ ローカル I/O 対応のモジュール (ram3068.mot) のインストール

ram3068.mot のインストールは、Windows なら

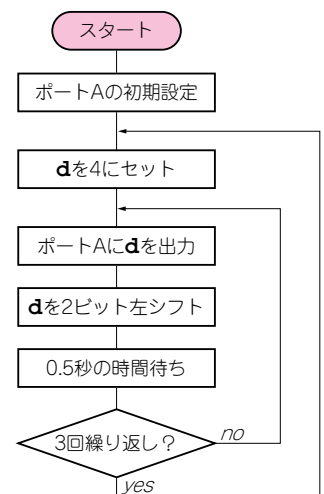
〈図1〉 ポート A に LED を三つ、
スイッチを一つ外付けた回路



〈表1〉 図1の回路でのポート A の割り当て

ビット	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
割り当て	(未使用)	LED6	(未使用)	LED4	(未使用)	LED2	(未使用)	SW0
CN1 のピン番号	32	31	30	29	28	27	26	25

〈図2〉 for 文による繰り返しと goto 文による無限ループのフローチャート



Keywords

HD64F3069, AKI-H8/3069F フラッシュマイコン LAN ボード, H8/OS, ram3068.mot, rom3068.mot, net3068.mot, 自動スタート, ローカル I/O, モトローラ S 形式, CGI, HTML, FORM タグ, IP アドレス, ネット・マスク, ポート番号, netconfig.

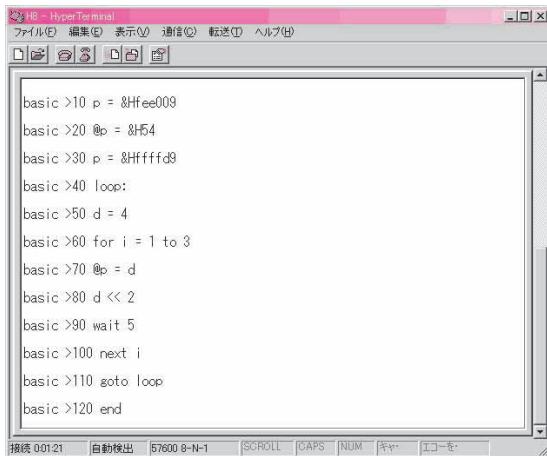
〈リスト1〉 for文による繰り返しと goto文による無限ループ (LED1.bas)

```

p = &Hfee009 ← ポートAのデータ方向レジスタ(PADDR)
@p = &H54 ← ビット2, ビット4, ビット6は出力で, それ以外は入力
p = &Hffffd9 ← ポートAのデータ・レジスタ(PADR)
loop: ← ラベルloop
d = 4 ← 変数dを4にセット
for i = 1 to 3 ← iを1から3まで繰り返す
  @p = d ← PADRにdを出力
  d << 2 ← 変数dの内容を2ビット左へシフト
  wait 5 ← 500ms待ち
next i ← 変数iの内容を一つ増やす
goto loop ← ラベルloopへ行く

```

〈図4〉 BASICプログラムの入力



ばCygwinのコマンド・プロンプトか、PC-UNIXならktermなどのターミナル・ソフトウェアを使います。最新版のh8writeを使って、ram3068.motを以下の手順で内蔵ROMに書き込みます。

まず、パソコンとH8/3069Fマイコン・ボードをシリアル・インターフェース・ケーブルでつなぎます。

次にH8/3069Fマイコン・ボードのDIPスイッチの1番と4番をON, 2番と3番をOFFにしてから、マイコンの電源をONし、ブート・モードで始動します。そして、以下のようにパソコン上で操作します。

```
h8write -3069 -f20 ram3068.mot
```

■ BASICのコマンド操作によるプログラム制作と保存

● ターミナル・ソフトウェアの起動

パソコンとH8/3069Fネットワーク・マイコン・ボードをケーブルでつないだまま、パソコンでは適当なターミナル・ソフトウェアを起動して、マイコンの電源を入れます。ターミナル・ソフトウェアは、Windowsでは「ハイパーターミナル」、PC-UNIXならcuやkermitなどが代表的です。通信設定は、57600 bps, データ長8ビット, パリティなし, 1スト

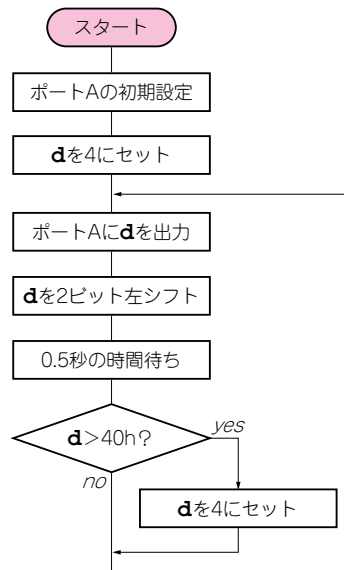
〈リスト2〉 if文によるデータ初期化と goto文による無限ループ (LED2.bas)

```

p = &Hfee009
@p = &Hff
p = &Hffffd9
d = 4
loop:
  @p = d
  d << 2
  wait 5
  if d > &H40
  then d = 4
  goto loop
end

```

〈図3〉 if文によるデータ初期化と goto文による無限ループのフローチャート



ップ・ビット, ハードウェア・フローです。

● プログラムの入力

パソコン側でターミナル・ソフトウェアを起動すると図4のようにBASICの入力プロンプトbasic >が現れます。その画面上で図のように行番号を付けて、BASICプログラムを入力していきます。

● 入力した内容を確認して実行

BASICの入力プロンプト上でプログラムをすべて入力し終わったら、図5のようにlistコマンドで入力したBASICプログラムの内容を確認し、runコマンドでBASICプログラムを実行します。

このBASICプログラムは無限ループになっており、そのままでは終わらないので、CTRL + Cキーを押して強制終了させます。

● プログラムの保存

BASICプログラムが無事に動作したら、saveコマンドでファイルに保存して、念のためにfilesコマンドでファイル一覧を表示して、きちんと保存されたかどうかを確認します。

以降、マイコンの電源が切れても、外付けのSEEPROM内に制作したBASICプログラムが保存されているので、いつでも読み出して実行できます。

■ BASICコマンド操作によるプログラム読み出しと実行

外付けのSEEPROMに保存したBASICプログラムを読み出して実行するには、図6のように、まずloadコマンドで読み出します。念のため、listコマンドでプログラムの内容を確認してから、runコマ