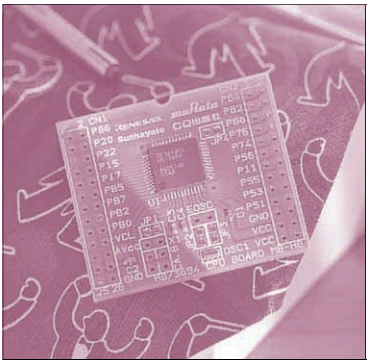

第2章　マイコン開発の第一歩…開発環境のインストールとマイコンとの通信

プログラミング言語の基礎とモニタ・プログラムの書き込み

大中 邦彦
Kunihiko Ohnaka

「コンピュータ．ソフトがなければただの箱」などといわれるように，付録基板に実装されたH8マイコンもプログラムがなければ動作しません．ここがアナログICなどとはちょっと違うところで，はんだ付けして「はい終了」というわけにはいきません．

本章では，マイコンを動かすプログラミング言語の基礎知識を説明した後，マイコンの開発環境を整えます．そして，早速H8/3694Fを実装した付録マイコン基板にプログラムを書き込んで動作させてみます．

マイコン開発のはじめの一歩です．じっくりと読んでください．

マイコンを動かすプログラム言語の基礎知識

● マイコン界の言語「マシン語」

人に何か仕事を依頼する場合は，日本語などの人間の言葉で伝えます．でも，マイコンには日本語は伝わりません．マイコンが理解できる言葉を使って話しかけなければなりません．このマイコンの世界の言語を「マシン語」または「機械語」といいます．

マシン語をお見せしましょう．**リスト1**のように数値だけの羅列で，これは皆さんが普段よく使用しているメモ帳などのテキスト・エディタで書いたものです．これをマイコンに転送して書き込む必要があります．

マシン語の文法は，CPUコアの種類ごとに異なるのが普通です．方言のような細かな違いから，日本語と英語のように単語や文の作り方からまるっきり異なるものまでさまざまです．この文法はCPUのメーカが決めているので，メーカが同じだったり，生い立ちが同じコア用のマシン語は似ています．

付録マイコン基板に実装されているH8/3694Fには，内部のCPUコアであるH8/300H CPUが理解できる専用のマシン語で話しかけなければなりません．

リスト1　マシン語の例

```
                              :
00000b0 0000 0000 7a07 0000 0000 f8ff 6aa8 00fe
00000c0 e000 6aa8 00fe e001 6aa8 00fe e002 6aa8
00000d0 00fe e003 6aa8 00fe e004 6aa8 00fe e005
00000e0 6aa8 00fe e007 6aa8 00fe e008 6aa8 00fe
00000f0 e009 6aa8 00fe e00a f8ff 6aa8 00fe e021
                              :
```

マシン語の一つの「語」

● マシン語を英単語に置き換えた「アセンブリ言語」

マシン語は数値の羅列で，何がなんだかわかりません．でも安心してください．通訳を使って，人間が使う言語で話しかければよいのです．この言語をアセンブリ言語，通訳をアセンブラといいます．

リスト2にアセンブリ言語の例を示します．これもテキスト・エディタで書いたものです．アセンブリ言語は，マシン語の数値が表す意味を，英単語を省略した形で置き替えたものです．アセンブラは，この英単語を翻訳してマイコンに話しかけてくれるソフトウェアです．

● アセンブリ言語よりわかりやすい「C言語」

アセンブリ言語はマシン語と比べれば，人間に馴染みのある言語ですが，それでもまだとっつきにくい感じがします．例えば，数値どうしを加算したい場合，本来なら“A+B”のように書ければ簡単ですが，アセンブリ言語では“add　A,B”のように記述します．また，乗算したい場合に，使用するマイコンが理解で

リスト2　アセンブリ言語の例

```
    .h8300h
    .section    .text
    .global     _start

_start:
    ;; スタックポインタの初期化
    mov.l     #_stack, sp

    ;; IOモードの設定(P0DDR~PBDDR)
    mov.b     #255, r0l
    mov.b     r0l, @0xfee000:24
    mov.b     r0l, @0xfee001:24
    mov.b     r0l, @0xfee002:24
    mov.b     r0l, @0xfee003:24
    mov.b     r0l, @0xfee004:24
    mov.b     r0l, @0xfee005:24
    ;; P8DDR (-CS0~-CS3の出力の有効)
    mov.b     r0l, @0xfee007:24
    mov.b     r0l, @0xfee008:24
              :
```

セミコロン二つで始まる行は，コメント(注釈)．プログラムの実行には影響しない

この行がマシン語に変換されると図2のf8ffという数値になる

きるマシン語に「乗算」という単語がない場合は，乗算を加算に分解して記述しなければなりません．これでは手間ですよね．

そこで，アセンブリ言語よりもさらに人間に理解しやすい言語が考えられました．これを「高級言語」と呼びます．特集では「C言語」と呼ばれる高級言語を使います．**リスト3**にC言語の例を示します．これもテキスト・エディタで書いたものです．まだ，意味まではわからないかもしれませんが，省略された英単語が使われていたアセンブリ言語よりは，いくぶん読みやすいのではないかと思います．

● C言語をマシン語に翻訳してくれる「コンパイラ」

C言語で書かれたプログラムをマイコンに理解させるためには，C言語からマシン語への翻訳作業が必要です．といっても人間がやるわけではなく，Cコンパイラというパソコン上で動くソフトウェアにやらせます．

図1にC言語で書かれたプログラムがマシン語になるまでの過程を示しました．

まず，C言語のプログラムはCコンパイラによってアセンブリ言語に変換されます．そしてアセンブラによって，いったんオブジェクト・ファイルというファイルに変換されます．オブジェクト・ファイルの中身はマシン語にかなり近い形をしています．

「かなり近い形」と言ったのは，この段階ではまだマイコンで実行可能なマシン語になっていないからです．通常，メモリ上に存在するプログラム，つまりマシン語のプログラムは，メモリ上の違うアドレスへ移動されると動かなくなってしまいます．これはマシン語の命令の中に，プログラムのアドレス情報が直接書かれているのが原因です．そこで，そういった情報をひとまず空欄にしておき，書き換えられるような形式にしたものがオブジェクト・ファイルです．

● よく使うルーチンをまとめた便利なファイル「ライブラリ」

オブジェクト・ファイルはアドレスなどが空欄なので，好きなところに配置できます．言い換えれば，オブジェクト・ファイルは再利用性に優れてます．よく使うプログラムをあらかじめオブジェクト・ファイルの形で用意しておけば，必要なときに，好きなプログラムに簡単にくっつけることができて便利です．しかも，オブジェクト・ファイルなので毎回コンパイル/アセンブルする必要はありません．

このような，よく使うオブジェクト・ファイルをたくさん集めて一つのファイルに固めたものをライブラリと呼びます．

● 複数のオブジェクト・ファイルをまとめる「リンカ」

オブジェクト・ファイルの空欄部分を埋めて，さら

リスト3　C言語の例

```
void main(void) {
 static int flag = 0;
 volatile unsigned char *led_register
                          = (unsigned char *)0xffdb;

 while(1) {
   if( flag == 0 ) {
     flag = 1;
     *led_register = (unsigned char)0x02;
   } else {
     flag = 0;
     *led_register = (unsigned char)0x04;
   }
 }
}
```

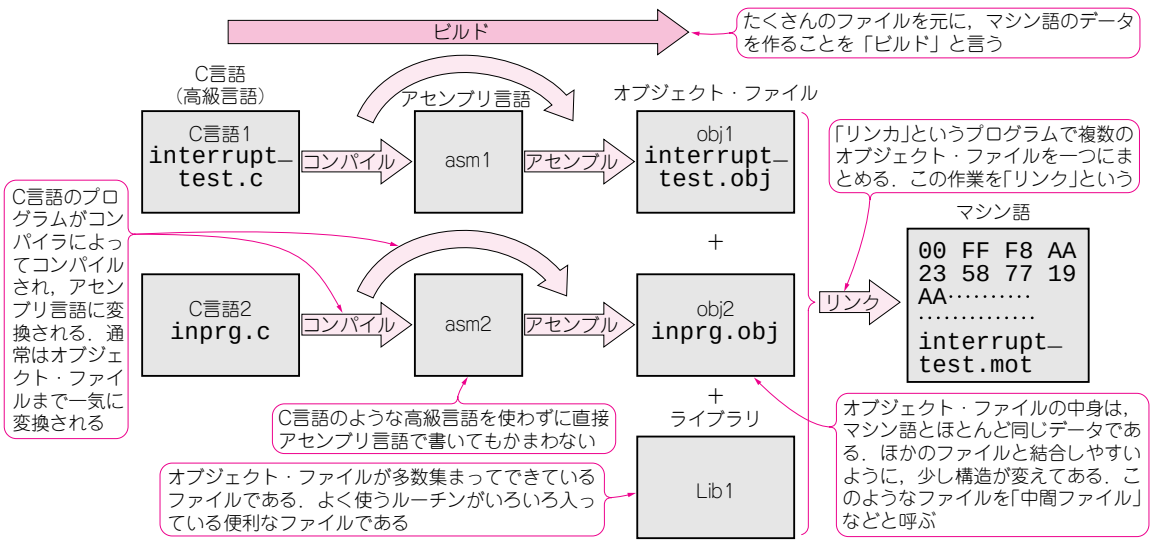

図1　C言語で書いたプログラムがマシン語に変換されるまで