

ハードウェア技術者のための 組み込み型ソフトウェア設計入門

—— 勘どころは保守性への配慮，ドキュメント作成，デザイン・レビュー

太田博之

ここでは、ハードウェア設計の知識のある技術者を対象に、組み込み型ソフトウェア設計の概要を紹介する。組み込み型ソフトウェアの開発では、組み込み型マイコン(マイクロコントローラ)とリアルタイムOSの機能を理解することが必要になる。ハードウェアの場合と違って、ソフトウェアでは複数の処理が同時並行に進むことはない。性能上の問題がある場合には、タスクの優先順位を指定するなどの対処を行う。なお、ソフトウェアのテストやコーディングに関するいくつかの考え方は、HDL 設計にも適用できる。(編集部)

コンピュータと言った場合、まず頭に思い浮かぶのはパーソナル・コンピュータ(パソコン)でしょう。パソコンの中には、CPUと呼ばれるコンピュータの頭脳と、プログラムが格納されるハード・ディスクやCD-ROMのドライブがあります。また、外部とのインターフェースとして、マウス、キーボード、CRT、モデムなどがあります。

パソコン以外のもう一つの世界として、組み込み型マイコン(マイクロコントローラ)の世界があります。組み込み型マイコンとは、デジタル・カメラやデジタル・テレビなどの

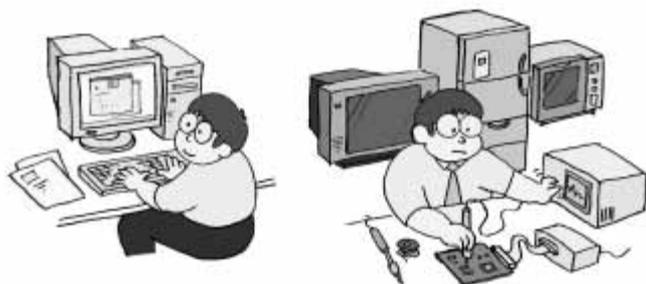
制御対象機器に組み込まれ、その制御対象機器を制御するコンピュータのことです。組み込み型マイコンは、プログラムをROMと呼ばれるメモリに格納しており、外部とのインターフェースには液晶パネルやタッチ・パネルなどが使用されます。組み込み型マイコンの場合、制御対象機器に組み込まれているため、一般の人の目に触れる機会はほとんどありませんが、テレビ、洗濯機、冷蔵庫など、ほぼすべての家電製品で使われています(図1)。

ここではハードウェア設計者を対象に、組み込み型マイコンの上で実行する組み込み型ソフトウェアの設計(以下、ソフトウェア設計とする)について紹介します。

●組み込み型ソフトウェアの設計フロー

ソフトウェアの設計フローは、図2のような流れになっています。設計の対象を明確にするための要求仕様(要求分析、仕様策定)から、それをもとに行われる基本設計、詳細設計、コーディング、テスト/デバッグと作業は流れていきます。また、開発したプログラムの動作を、試作基板などを使って早期に確認するためのプロトタイプングも行われます。

現在は、ハードウェア設計もHDLなどの言語を使用して行うため、コーディングやデバッグなどの手法は似たものもあります。ソフトウェア設計において行われている手法のなか



〔図1〕 パソコンと組み込み型マイコン

組み込み型マイコンは、プログラムをROMと呼ばれるメモリに格納しており、外部とのインターフェースには液晶パネルやタッチ・パネルなどが使用される。ソフトウェアの開発にはICE やロジック・アナライザが利用される。



〔図2〕 設計フロー

組み込み型ソフトウェアの設計フローを示す。

```

●呼び出し方法
unsigned char bitmov(unsigned short movin[], unsigned short movout[],
                    unsigned short instbit)

●引き数
IN      : movin      入力データの先頭配列アドレス
         : movout     出力データの先頭配列アドレス
         : instbit    入力移動開始ビット数

●戻り値
OUT     : errrcode   エラー・コード
         : NO ERROR   : 0
         : instbit    =0: 1

●機能
movinに示される先頭番地のinstbitのデータから、1ビット分のデータを
movoutに示される先頭番地に格納する。
    
```

[図3] モジュール定義の例

モジュールの呼び出し方法、引き数、モジュールの機能を定義する。表形式でもとめてもよいし、ソース・コードの関数記述の前の行にコメントとして挿入してもよい。

```

●コメント開始けた数を統一する。
  (入れ子の制限ができる。見た目がきれいで疲れない)

●関数の始まりに付けるヘッダの形式を統一する。
  (モジュール外部仕様に対応させ、引き数と戻り値、ふるまいなどを明記)

●定数名、変数名、関数名は意味のあるものにする。

●コメントにはアルゴリズムを書く。
  (コードと同じことを書かない。
   なぜそうしたかを書く。なにをするかはコードを見ればわかる)

●コメントに使用する言葉を統一する。用語集を作る。
  (init, success, fall, hold, go, ngなど)

●演算子前後のスペース、入れ子のインデント数を統一する。
    
```

[図4] コーディング規約の例

プロジェクトまたは部署ごとにこのような内規を決めて設計すると、設計物の資産価値が向上する。

には、ハードウェア設計に有効なものが多くあります。それらも紹介しながら、話を進めていきます。

(1) 要求仕様

要求仕様の作成はソフトウェアでなにを実現するかを明確にする作業です。組み込み型ソフトウェアの開発では、ソフトウェアに対する要求仕様は以下の式で表されます。

$$Sreq = Preq - Hexe + Hctrl$$

Sreq : ソフトウェアに対する要求仕様

Preq : 製品に対する要求仕様

Hexe : ハードウェアで実現する機能

Hctrl : ハードウェアの制御要求仕様

一般的な家電製品の場合、要求仕様の一部はハードウェアで実現されます。したがって、製品仕様(Preq)からハードウェアで実現(Hexe)する部分を除いたものがソフトウェアに対する要求仕様になります。ただし、ハードウェアを制御する機能(Hctrl)もソフトウェアがもたなくてはなりません。

これらの関係を明確にし、ソフトウェアで行うことを書き出したものが要求仕様です。

(2) 基本設計

要求仕様を満足する基本構造を設計します。通常は、ジェネラル・フローと呼ばれるフローチャートの記述から始まります。ちょうどハードウェア設計において、全体の構成を把握するために全体のブロック図を作成する作業と似ています。

基本設計では、ジェネラル・フローをもとにプログラムの基本構造を検討し、作りやすい大きさにモジュール分割します。作業としても、ハードウェア設計において、シミュレーションしやすいブロックや論理合成しやすいブロックに分ける作業とよく似ています。モジュール分割が終了したら、各モジュールの外部仕様を定義します。

この時点では、以下の項目を定義します。

- モジュールの呼び出し方法(関数名、戻り値の型)
- 引き数(数とその種類)
- モジュールの機能

とくに最後のモジュールの機能については、そのふるまい(ビヘイビア)を規定します。機能の実現方法を詳細に定義する必要はありません。図3に定義例を示します。

(3) 詳細設計

詳細設計では基本設計で定義したモジュールの内部を設計します。コーディング直前の仕様を作成します。ここでは基本設計の工程で考えなかったモジュール機能の実現方法について、詳細に検討(設計)します。

(4) コーディング

詳細設計で決まった仕様に沿って、コーディングを行います。プログラム言語としてはテストや保守が容易な高級言語の使用が主流で、とくにC言語がよく使われます。最近では、コンパイラの性能がよくなっているため、機械語(アセンブリ言語)で記述しなければならない部分は一部のみですみます。C言語がよく使われる理由には、次のようなことがあげられます。

- 移植性が高い(大半の組み込み型マイコンにはCコンパイラが用意されている)
- 実アドレスを扱えるポインタ変数がある
- ビット操作機能が強力
- 構造化プログラミングのためのフロー制御構文がある

多くの場合、ソフトウェアの保守性を高めるために、プロジェクトや部署ごとにコーディング規約を作成し、規約に沿ってコーディングを進めていきます。ソフトウェア設計の場合、10人以上のチームで設計する機会が少なくありません。そうした場合、プロジェクト・リーダーからのチェック以外に、設計メンバー間でのすりあわせやチェックも多く行われます。そのときにベースとなるのがソース・コードです。ソース・コ