

無駄のないデバイス・ドライバを設計する

地道な分析と自由な発想を両立しよう

森 孝夫

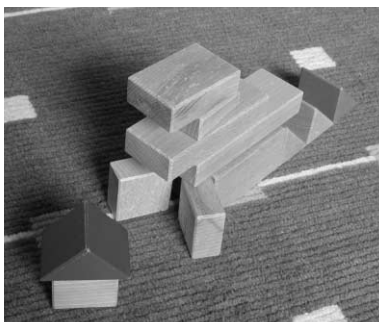
ここでは、デバイス・ドライバの設計について解説する。デバイス・ドライバは、ハードウェア（デバイス）と密接に関連しているので、これを作成するにはソフトウェアからデバイスを操作するための知識や、割り込みに関する知識のほか、移植しやすいようにモジュール化する技術も必要となる。（編集部）

デバイス・ドライバとは、特定のハードウェア（デバイス）を制御するために用意されるソフトウェア・モジュールです。デバイス・ドライバが用意されていると、アプリケーション・ソフトウェアはデバイスの機能を簡単に利用することができます。

組み込みシステム開発では、製品に新機能を追加するときなどに新しいデバイスを搭載することがあり、その際にデバイス・ドライバの設計が必要となります。本稿では、移植性に優れ、なおかつデバイスの性能を引き出すことができるデバイス・ドライバを設計するノウハウを解説します。

写真1 積み木

積み木を見ると積んでみたくなる。積んでいるうちに、新しい形を作り出してしまふ。積み木やブロックは、そのような不思議な性質を持っているように思われる。



1 機能美のあるシステムを目指して

デバイス・ドライバに限らず、組み込みソフトウェアは「無駄がない」ということを強く求められます。なぜなら組み込みシステムは、ぎりぎりの性能、ぎりぎりまで小さくしたハードウェアで構成されることが多いからです。

組み込みシステム開発では、ユーザ（顧客）に「良いものを安く」提供するために、製品に必要な機能を満たせるハードウェアの中で、できるだけ安価なものを使おうとします。その結果、多くの場合、性能面ではぎりぎりのハードウェアが使われることになるのです。

組み込みシステムに対する要求は、機能のほかにも、リアルタイム性や低消費電力、（取り付けスペースなどの関係で）回路を小さくしてほしいという要求など、次々と出てきます。これらを達成するためには、いずれも無駄をなくしたシステム設計が必要です。無駄をなくした設計こそが、結果として、ユーザに価値を提供することになります。

この、無駄のないシステム作りを支えているのが、組み込みソフトウェアです。組み込みソフトウェアがシステムの性能を極限まで引き出すからこそ、ぎりぎりの性能や大きさのシステムに価値ある機能が搭載できるのです。そのスマートさはまさに「機能美」と呼べるものではないかと思います（コラム「組み込みシステムには機能美がある」を参照）。

● 組み込みソフトウェアの組み立て方

皆さんは、小さなころに「レゴ」で遊んだことはありませんか

KeyWord

デバイス・ドライバ、機能美、設計、大喜利、あいうえお作文、Sound Blaster、隠れた要求、責務

COLUMN

組み込みシステムには機能美がある

筆者は、電化製品や自動車、ロケットの模型などに触れたとき、感覚的に「機能美」を感じることがあります。「知性の美しさ」といった方が、筆者の感覚を正しく表しているかもしれません。でき上がった物の形や動きを見るだけで、「機能を実現するためのスマートな知性」を感じ、そこに美しさを感じることがあるのです。音楽、美術、建築物、はたまたスポーツのプレー、時には動物の本能的な行動を見て、同じような感覚を抱くことがあります(写真A)注A。

そのような「知性」を感じる製品について設計思想を聞いてみると、機能を実現するためのきめ細かい工夫や、無駄をそぎ落とすための工夫、信頼性を保つための工夫が随所に施されています。

製品を見ただけでは、製品の設計思想を知る術はありません。なのに、見ただけで感覚的に、しかも一瞬にして知性を感じるというのは不思議なものです。この感覚が生まれる理由を筆者は合理的に説明できないのですが、エンジニアが続けていると、こうした機能美を感じる感性みたいなものが備わってくるような気がします。

注A：実は逆もある。ちょっと触って「これはバグが出る気がする」と感じることがあるのだ。経験的に言って、エンジニアが持つ「なんか怪しい…」という感覚はかなり当たる。

仕事で大切なのは、「良い仕事をした」と思えるような、そして周りにも「良い仕事をした」と思ってもらえるような仕事をする事です。われわれ組み込みシステムのエンジニアは、ユーザに安くて良い物を提供しようと追求することで、自然と「機能美」を作っているように思います。それは、エンジニアにとっての大きな楽しみなのではないでしょうか。



写真A 「機能美」を感じる製品の例
筆者の愛車。最近入手したばかり。

んか？人は誰しも、レゴ・ブロックを見るとなぜか、とりあえず「カポツ」と組み合わせます。そして、いろいろ組み合わせで試行錯誤を繰り返し、そのうち新しい形を作り出します。これは、レゴ・ブロックや積み木遊びが持つ不思議な性質であるように思われます(写真1)。

人間には、レゴ・ブロックや積み木のような「材料」が転がっているのを見ると、それを組み合わせたり、分類したりして、自身がじっくりくるような「新しい設計」を考える性質があるような気がします。

組み込みソフトウェアの設計も、まずは「材料」探しから始めてみましょう。組み込みソフトウェアの材料とは、「組み込みソフトウェアが内部で行うこと」です。「内部で行うこと」をまとめてプログラミングすると、ソフトウェアのモジュールになります。そして、モジュールの作り方と組み合わせ方が「設計」というわけです(p.117のコラム「設計とは大喜利である」を参照)。

「内部で行うこと」を決めるためには、組み込みシステムで実現したいと思っている機能のほか、提供されたハードウェアやOS、プログラミングに使用するライブラリなどの仕様をよく知る必要があります。例えば、「前進する」という機能をマイコンとモータで実現しようと思ったら、「モータに電圧をかける」ことが必要です。そのためには、

マイコンのI/Oポートとモータの接続を知る必要があります。I/Oポートに電圧をかけるための命令も知る必要があります(図1)。そして、知った内容をうまく組み合わせ、モジュール構造を作るのが「設計」です。

組み込みソフトウェアの設計とは、要求された機能や性能を、与えられたシステムの制約下で実現する方法を考える行為です。その第一歩は、要求仕様書、ハードウェアの仕様書、OSの仕様書などをよく読み、要求と制約の両方を把握することです。「機能美」は、こうした地道でまっとうな努力の積み重ねから生まれるのです。

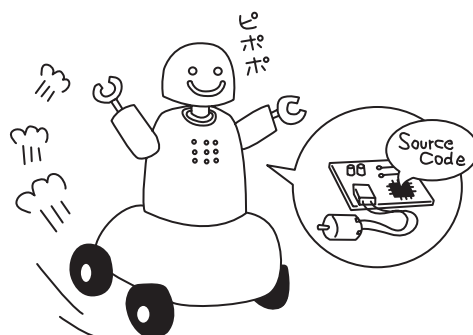


図1 要求と制約を把握して機能や構造を考える

ロボットを前進させるためには、マイコンとモータの接続や、I/Oポートへの命令の仕方を知る必要がある。これらの「知った内容」をうまく組み合わせモジュール構造を作るのが「設計」である。