組み込みビギナ必読！

強いプログラムを作るテクニックを学ぶ

第11回　実装条件を設計に盛り込む

酒井 郁子
舘 伸幸

前回（第10回，2011年10月号，pp.146-151）作成した構造図からコーディングできそうですが，それだけでは実際の動くプログラムが作成できません．実装制約や動的条件を考慮して，もう一段階掘り下げた設計図が必要になります．
（編集部）

（野比さん）

…カタカタ…

（パソコンに向い，ひたすら無言で打ち込み中）

（物造先輩）

何しているの？ うぉっ！！ もうコーディングしている．DFD（Data Flow Diagram）から変換した構造図のままコーディング始めたのか．なんと，早計な！

（野比さん）

え〜，まだコーディングしては駄目なのですか？ だって構造図作ったから，各モジュールを関数にして，呼び呼ばれ構造で実装すればいいのですよね．構造図に展開する作業はすぐ終わりましたし，先輩はひとしきり講釈述べたら1人で休憩に行ったまま帰ってこないし…

（物造先輩）

DFDから作った構造図は設計のたたき台であって，プログラム化するにはまだまだ考えるべきことがあるのだよ．実装する前に，設計で取り込む条件はたくさんあってだね…

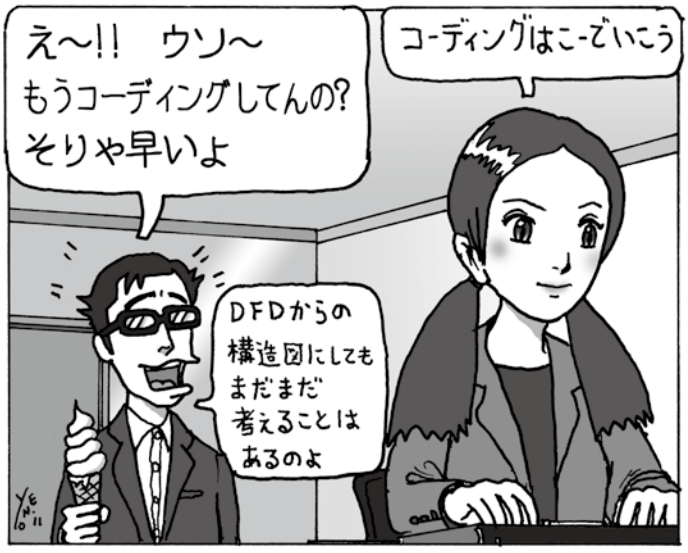

図1　開発風景

（野比さん）

はいはい．長い話のようですので，本編へいってみましょう！

前回の「分析モデルからプログラム構造を設計する」では，DFDモデルによる分析結果から，機械的にモジュール構造図を作成しました．そのDFDを展開したモジュール構造図が**図2**で，この構造図をもとに野比さんがコーディングしたものが**リスト1**です．

DFD分析によってできたものは，論理面で機能分割を考えた静的な機能構造です．そして，このDFDの変換により作成できるモジュール構造は，コール＆リターン・アーキテクチャ（呼び呼ばれ関係の構造）として，プログラムに実装できます．しかし，「動く」プログラムにするためには，単に論理的な静的機能構造からコール＆リターンへ変換するだけでは十分ではありません．論理に加えて，プログラムが実際に動作する上での，さまざまな制約を考慮する必要があります．例えば，

①外部資源（入力信号や出力メモリ，etc.）の物理的制約を考えて制御手順を工夫する
②マイコンの処理能力の制約に対して，並行に機能を動かすなどで処理効率を上げる
③時間制約のある周期動作を優先的に動かす

といったものが代表的です．このような制約群を，一般に実装条件といいます．

DFDで機能分割しただけの構造には，この「実装条件」が盛り込まれていません．したがって，プログラムを実装する前に，もう1ステップ設計を深めておく必要があります．では，代表的な実装条件についての考え方について見