

基本設計を考えよう

坂本 直史

基本設計工程では、要求仕様をどのように実現するかを決定する。システムとして実現する機能を決定してからハードウェアとソフトウェアに分割し、その後、ソフトウェアの構造を決定する。なお、ハードウェア設計については本稿の対象外とする。
(編集部)

1. 要求分析の結果から「どう作るか」がテーマ

基本設計工程では、前工程の要求分析結果を受けて開発対象のシステムをどう作るかを決めます。組み込みシステムの場合、システム自身がハードウェアを含み、また(パソコン上で動作するソフトウェアとは異なり)プログラムの動作するハードウェアがその製品専用であることがほとんどなので、この「どう作るか」にはハードウェア設計も含まれます。今回扱っているのは実験的な簡単なシステムな

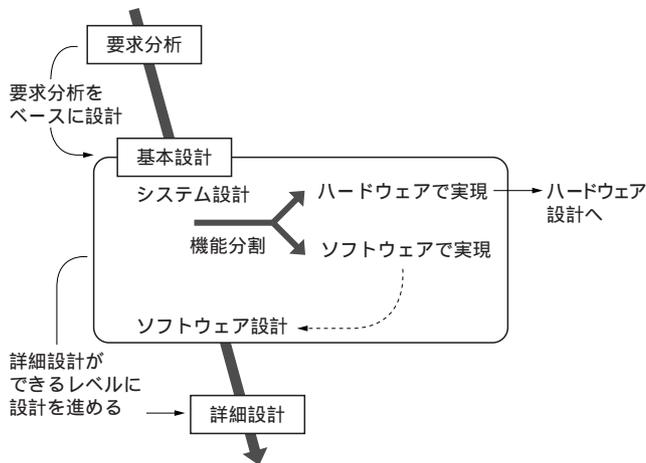


図1 基本設計の概要

まず、要求仕様の各項目をシステムとしてどのように実現できるのかを設計し、実現すべき要求をソフトウェアとハードウェアに分割する。ここまでがシステム設計となる。その後、ハードウェアで実現すると決めた仕様はハードウェア設計に、ソフトウェアで実現すると決めた仕様はソフトウェア設計に委ねられる。

ので、マイコンを搭載したボードを決定するだけですが、実際のシステム開発ではハードウェア設計に非常に重い負荷がかかります。

システム設計からソフトウェア設計へ

ここでは、基本設計の前半を「システム設計」、後半を「ソフトウェア設計」と大ざっぱに分けて説明します。

システム設計は、要求分析の結果として得られた要求仕様の各項目について、ハードウェアかソフトウェアかといったことにはこだわらず、「システムとして」どのようにすれば実現できるのかを設計します。次に、実現するべき要求をハードウェアとソフトウェアに分割します。この分割は、機能要求や性能要求などを考慮して行います。つまり、どの要求をハードウェアで実現するのかソフトウェアで実現するのかを決め、それぞれの果たすべき役割を明確にするのがシステム設計です(図1)。

ハードウェアで実現すると決めた役割や仕様は、ここからハードウェア設計に委ねられます(こちらは本特集の対象外とする)。そして、ソフトウェアで実現すると決めた役割や仕様に基づいて「ソフトウェア設計」を行います。

基本設計工程におけるソフトウェア設計では、モジュール(ある機能を実現する処理単位を指す。プログラムでいう関数ではない)ごとに見たときの構造を設計します。

システム設計 機能や性能の実現方法を検討する
最近では、マイコン(またはCPUコアを搭載したLSI)をシステムの核としてハードウェアを構成することがほとんどなので、ハードウェアとソフトウェアの役割を決める「システム設計」とは、マイコンの選択やLSIの仕様決定が

中心になります。その際に、ソフトウェアで実現したい機能に必要なマイコンの外部端子を割り当てられるか(つまり周辺ハードウェアと必要な接続を実現できるか)や、必要な割り込みと必要な動作を実現するための割り込みのレベルが実現できるか、マイコンの周辺機能をどのようなモードで使用するかなどを合わせて検討します。この検討は、ハードウェアを担当するメンバと一緒にすることになります。

今回の例で検討した結果を表1に示します。

実際にはこれ以外に、デバッグ用の端子を数本使用できる余裕が欲しいものです。また内部処理のために、タイマが必要になります。今回の例では、CANの定期送受信のためと、中速走行が一定時間以上続いた場合のオートロック制御機能を実装するためにタイマを使います。

適度に余裕を持った設計を心がけよう

このように、どのような機能が要求されているか(また

表1 必要な端子の検討例

実現したい機能	実現方法	マイコンに必要な端子や機能
表示部	7セグメントLED × 2	出力ポート × 10本 (7セグメントLEDの制御用。うち2本は7セグメントLEDの左右の選択に使う)
速度(アナログ値)入力	ポリリウム・スイッチ	8ビットA-Dコンバータ (速度としてアナログ値を入力する)
CAN通信	CANモジュール	CANモジュール, CANデータの送受信端子 × 2本 (CAN通信対応マイコンを選択する)
車速度とドアの開閉状態、ドアのロック状態の検知	割り込み	割り込み入力 × 3本 (ポーリングではなく、まずは反応の速い割り込みを想定した。受信は少なくとも割り込みにする必要がある)

後から要求されるか)によって、必要なハードウェア資源が変わってきます。しかし、ハードウェアの仕様が決まっていたんボードを設計してしまうと、あとから「こんな機能が必要になったのでマイコンを変えてください」とか

表2 端子定義表

番号	端子名称	端子の入出力機能	マイコン内プルアップ/プルダウン	機能名称	機能	使用する入出力方向	初期設定値	外部プルアップ/プルダウン	端子有効設定が必要な機能選択関連レジスタ
2	CTX	O		CTX	CANデータ送信	O			C0CTL レジスタ: PortEn ビット
3	CRX	I		CRX	CANデータ受信	I			C0CTL レジスタ: PortEn ビット, PD9 レジスタ: PD9_2 ビット
15	INT2#	I	プルアップ	INT2	INT2# 割り込み入力	I			PD8 レジスタ: PD8_4 ビット (PUR2 レジスタ: PU21 ビット)
16	INT1#	I	プルアップ	INT1	INT1# 割り込み入力	I			PD8 レジスタ: PD8_3 ビット (PUR2 レジスタ: PU20 ビット)
17	INT0#	I	プルアップ	INT0	INT0# 割り込み入力	I			PD8 レジスタ: PD8_2 ビット (PUR2 レジスタ: PU20 ビット)
32	P3_7	I/O		DBG7	デバッグ用 LED 出力 7	O			PD8 レジスタ: PD3_7 ビット
33	P3_6	I/O		DBG6	デバッグ用 LED 出力 6	O			PD8 レジスタ: PD3_6 ビット
34	P3_5	I/O		DBG5	デバッグ用 LED 出力 5	O			PD8 レジスタ: PD3_5 ビット
35	P3_4	I/O		DBG4	デバッグ用 LED 出力 4	O			PD8 レジスタ: PD3_4 ビット
36	P3_3	I/O		DBG3	デバッグ用 LED 出力 3	O			PD8 レジスタ: PD3_3 ビット
37	P3_2	I/O		DBG2	デバッグ用 LED 出力 2	O			PD8 レジスタ: PD3_2 ビット
38	P3_1	I/O		DBG1	デバッグ用 LED 出力 1	O			PD8 レジスタ: PD3_1 ビット
39	P3_0	I/O		DBG0	デバッグ用 LED 出力 0	O			PD8 レジスタ: PD3_0 ビット
44	P2_7	I/O		ST7	状態表示用 7セグメントLED 出力(dp)	O			PD2 レジスタ: PD2_7 ビット
45	P2_6	I/O		ST6	状態表示用 7セグメントLED 出力(g)	O			PD2 レジスタ: PD2_6 ビット
46	P2_5	I/O		ST5	状態表示用 7セグメントLED 出力(f)	O			PD2 レジスタ: PD2_5 ビット
47	P2_4	I/O		ST4	状態表示用 7セグメントLED 出力(e)	O			PD2 レジスタ: PD2_4 ビット
48	P2_3	I/O		ST3	状態表示用 7セグメントLED 出力(d)	O			PD2 レジスタ: PD2_3 ビット
49	P2_2	I/O		ST2	状態表示用 7セグメントLED 出力(c)	O			PD2 レジスタ: PD2_2 ビット
50	P2_1	I/O		ST1	状態表示用 7セグメントLED 出力(b)	O			PD2 レジスタ: PD2_1 ビット
51	P2_0	I/O		ST0	状態表示用 7セグメントLED 出力(a)	O			PD2 レジスタ: PD2_0 ビット
66	P0_1	I/O		STL	状態表示用 7セグメントLED 左右切り替え出力(右)	O			PD1 レジスタ: PD1_1 ビット
67	P0_0	I/O		STR	状態表示用 7セグメントLED 左右切り替え出力(左)	O			PD1 レジスタ: PD1_0 ビット
76	AN0	I		SPD	速度情報入力	I			PD10 レジスタ: PD10_0 ビット